Development and verification of a numerical simulation for predicting the scattering of an incoherent light band from 30 µm to 210 µm from a random rough surface as well as a fabrication technique for said surfaces

Anton Nikolaev Atanasov

June 22, 2014

#### GRADUATION REPORT

Submitted to Hanze University of Applied Science Groningen

in partial fulfillment of the requirements for the degree of

Fulltime Honours Bachelor Advanced Sensor Applications

#### Abstract

This bachelor research consists of creating two two-dimensional light band scattering simulations using the method of moments technique applied to the scalar wave approximation and the small perturbation method taken from Tsang et al. (2001). The simulations are verified for wavelengths in the far infrared ranging from 30 µm to 65 µm and are then used to simulate band scattering from 30 µm to 210 µm which represents the SAFARI detection range (ESA, 2014). It is shown that with the current manufacturing capabilities present within SRON, proper scattering surfaces cannot be achieved. The test samples for the light scattering experiments are aluminum type 6061 plates and are sandblasted using various pressures, nozzle distances, exposure times, and grain sizes. An artificial neural network (ANN) is created with the purpose of imitating the sandblasting process. Taguchi's orthogonal arrays scheme is used to create a training set and the network was verified against 5 samples with different parameters. A surface profile analysis tool is written in MATLAB which can detrend, extrapolate, and perform several hypothesis tests on the measured profile data. The analysis of these statistics has shown that due to the variable irregularities of the entire surface profile, extensive care must be taken when applying filters to separate the drift component from the rough component of the measured profile. Additionally it is investigated whether independent component analysis (ICA) can be applied in the case when a flat test sample is processed with two different types of grains sequentially.

#### DECLARATION

I hereby certify that this report constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of another.

I declare that the report describes original work that has not previous been presented for the award of any other degree of any institution.

Signed,

Anton Nikolaev Atanasov

#### ACKNOWLEDGMENTS

I would like to thank Willem Jan Vreeling, Stephen Yates, Darren Hayton, and Andrey Baryshev for their numerous suggestions, advice, tips, hints, and overall common sense input with designing, assembling, and running my experiments.

I would like to thank Ronal Hesper and Andrey Khudchenko for the entertaining and fruitful dinner conversations.

I would like to thank Bryan Williams and Julian Wilson for their efforts in teaching proper engineering mathematics.

I would like to thank my SRON supervisors Wouter Laauwen and Pieter Dieleman for turning a blind eye to the fact that for the past half a year I've never managed to arrive earlier than 10 a.m. to work.

I would like to thank my housemates Krisiana Rozite and Nonhlanda Dube for their kind constructive criticisms. Finally I'd like to thank my friends for always supporting me in bars.

## Contents

1	Rationale         1.1       Previous Work	<b>8</b> 15			
2	Surface Analysis         2.1       Situation and Theoretical Analysis         2.2       Conceptual Model         2.3       Research Design         2.4       Research Results	<ol> <li>18</li> <li>19</li> <li>22</li> <li>25</li> </ol>			
3	Manufacturing Control Procedure         3.1       Situational and Theoretical Analysis         3.2       Conceptual Model         3.3       Research Results	<b>34</b> 34 36 38			
4	Light Band Scattering Simulation         4.1       Situational and Theoretical Analysis         4.2       Conceptual Model         4.2.1       Method of Moments         4.2.2       Small Perturbation Method         4.3       Research Design         4.4       Research Results         4.5       Extrapolation         4.6       Comparison of the SPM and MoM Simulations	<b>42</b> 45 45 52 53 55 63 64			
5	Complex Scattering Surfaces5.1Situational and Theoretical Analysis5.2Conceptual Model5.3Research Design5.4Research Results	68 68 70 72 72			
6	Conclusion 79				
7	Discussions	81			
Aj	Appendices 82				
Α	A Data Analysis Code 8				
в	ICA Simulation Code	90			

#### CONTENTS

С	ICA Analysis Code	96
D	MoM Band Scatter Simulation Code	102
$\mathbf{E}$	SPM Band Scatter Simulation Code	110
$\mathbf{F}$	Neural Network Code	114

## List of Figures

$\begin{array}{c} 1.1 \\ 1.2 \end{array}$	3D rendering of SPICA, taken from Klandermans (2013)	8
1.3	(2013)	10
	has several inner isolation containers in order to maintain the temperature gradient of approximately $300 \mathrm{K}$ external and $1.7 \mathrm{K}$ internal. The image was taken from	
1.4	Klandermans (2013)	11
	coupled optically tight using thermal breaks (5) and the yellow components on the sides represent heat absorbers. The image was taken from Klandermans (2013).	12
1.5	Mechanical iris used to control the optical intensity. The mechanical iris is actu- ated using electromagnets and has an inbuilt control loop which uses a Hall sensor	
	for feedback. This allows for precise apertures to be set and kept constant. The image was taken from Klandermans (2013).	13
1.6	Difference between specular and diffuse scattering. It can clearly be seen that the rougher the surface, the more diffuse the scattering. The image of c) repre- sents what is known as approximating a Lambertian scattering - equal intensities in all directions regardless of the incident angle. Image was taken from Japan	
	Association of Remote Sensing (1996).	14
1.7	Example profile measurement. The surface has not been detrended. The y-axis is in nm while the x-axis is in µm for the purpose of distinction.	16
1.8	FTS6000 (Online, 2014) measurement of two samples processed with F-8 and F-12	1 🗖
	grain sizes (Abrasives, 2013). The x- and y-axis are logarithmic.	17
2.1	Flowchart of the surface analysis tool	21
2.2	Example of proper fabrication and detrending	26
2.3	Example of defective fabrication	27
2.4	Comparison between partial and full ACFs	28
2.5	AR model evolved for 5 million data points, following the same resolution as the	
	measured profile	29
2.6	The induced most likely original signals	30
2.7	AR model evolved for 5 million data points, following the same resolution as the	0.0
	measured profile	32

#### LIST OF FIGURES

3.1	An example of a single neuron and a complete neural network. Details have been omitted for the sake of generality.	35
3.2	Example neural network used as a plant controller. $u$ represents the control inputs, $y_p$ represents the analyzed fabricated samples, and $y_m$ represents the network's	00
3.3	output. Image recreated from Demuth and Beale (2002)	37
3.4	The network's performance suggests that the data set has been properly divided which suggests that there should be little overfitting	39 40
$4.1 \\ 4.2$	Light scattering measurement setup alone	53
4.3	rotating plate to the right	54
4.4	is used to recapture evaporating helium	55 56
4.5	Overview of the entire experimental setup. Emphasis on the lack of connection between some of the components	56
$4.6 \\ 4.7$	Normalized discretized Planck distribution	57
4.8	m, $\lambda = 65 \mu\text{m}$	57 58
4.9	Scattered band of 2D spherical coordinates (represented in Cartesian space) with $\sigma = 21.8 \times 10^{-5}$ m, $\tau = 7.83 \times 10^{-5}$ m, $b = 2.26 \times 10^{-7}$ m, $\lambda = 30 - 65 \mu\text{m}$	58
4.10	Conditionality of operator matrix. It can be seen that as the simulation proceeds, the conditionality converges	50
4.11 4.12	Beam pattern measured from an unprocessed sample	59
4.13	the simulation. Angle error of $4^{\circ}$	60
4.14	2.031×10 <sup>-7</sup> m. Measured data has been normalized. Square data points represent the simulation. Angle error of 4°	60
4 15	$2.05 \times 10^{-7}$ m. Measured data has been normalized. Square data points represent the simulation. Angle error of 4°	61
	$2.09 \times 10^{-7}$ m. Measured data has been normalized. Square data points represent the simulation. Angle error of 3.5°	61
4.16	Type 1, experiment 5: $\theta_{inc} = 45^{\circ}$ , $\tau = 1.08 \times 10^{-4}$ m, $\sigma = 1.93 \times 10^{-5}$ m, $b = 2.058 \times 10^{-7}$ m. Measured data has been normalized. Square data points represent	01
4.17	the simulation. Angle error of $4^{\circ}$	62
4.18	the simulation. Angle error of $4^{\circ}$	62
0	m. The wavelength range is from $30 \mu\text{m}$ to $65 \mu\text{m}$	64

4.19	Comparison between the two power distributions	65			
4.20	) Comparison between the two scattered fields from 180 µm to 210 µm 6				
4.21	SPM simulation for $\theta_{inc} = 45^{\circ}$ , $\tau = 1.08 \times 10^{-4}$ m, $\sigma = 1.93 \times 10^{-5}$ m, $b = 2.06 \times 10^{-7}$				
	m. The wavelength range is from $180\mu\text{m}$ to $210\mu\text{m}$ for a Planck temperature of				
	850 K	67			
5.1	Two arbitrarily generated signals	69			
5.2	Two independent combinations of the arbitrary signals	70			
5.3	The induced most likely original signals	71			
5.4	Surface profiles of sample "A" and "B" after being detrended by a 5 <sup>th</sup> order				
	polynomial	73			
5.5	Comparison of the ACFs of sample "A" and "B". Emphasis on the sinusoidal				
	component of "A"	74			
5.6	Comparison between the extrapolated distributions of samples "A" and "B". Sam-				
	ple "A" has a truncated Gaussian	75			
5.7	Comparison between the normality plots of samples "A" and "B"	76			
5.8	Other half of profile "B" treated as a linear sum of two unknown signals	77			
5.9	The two new estimated signals are the original signals. No separation has occurred	78			

# Chapter 1 Rationale

SPICA (Space Infra-Red Telescope for Cosmology and Astrophysics telescope) is part of the JAXA (Japanese space agency) (JAXA, 2003) future science program and is planned for launch at the end of 2026. High sensitivity photometric observations in the MIR/FIR are made possible thanks to the large 3 m telescope which is actively cooled to 5 K to effectively eliminate the non-astronomical photon noise. Figure 1.1 shows a 3D rendering of the entire observatory. The thermal environment required by the telescope and the instruments will be maintained by a combination of passive cooling (via dedicated solar and thermal shields combined with radiators) and active cooling, using a number of mechanical coolers to provide base temperatures of 4.5 K and 1.7 K (Ferlet et al., 2009).



Figure 1.1: 3D rendering of SPICA, taken from Klandermans (2013)

One of the main goals of SPICA will be to provide a multidisciplinary approach to determining the conditions for planetary system formation. This includes the first detection of the most relevant species and mineral components in the gas and dust of protoplanetary disks at the time of planet formation. SPICA will have the unique ability to observe water ice in all environments and thus fully explore the impact of water ice on planetary formation and evolution as well as the emergence of habitable planets (ESA, 2014).

It will also provide direct imaging and low-resolution mid-infrared spectroscopy of young giant exoplanets (Goicoechea et al., 2009), which will allow the study of the physics and composition of their atmospheres in a wavelength range particularly rich in spectral signatures (e.g.  $H_20$ ,  $CH_4$ ,  $O_3$ , silicate,  $NH_3$ , and  $CO_2$ ) and to compare this to the planets in our Solar System for the first time. The combination of these observations will provide key clues to the question of whether our Solar System is unique in our universe.

The SAFARI instrument is an imaging Fourier Transform Spectrometer. It operates simultaneously in three wavelength bands to cover the 34 µm to 210 µm range over the full field of view. Within one hour in a single field SAFARI will typically observe spectra for 5 - 7 individual sources, thus allowing large area surveys yielding data for many thousands of objects. To reach the extreme sensitivity needed to fully profit from the unique low background condition provided by the SPICA satellite, SAFARI uses transition edge sensors (TES), which is a cryogenic detector that works by transitioning in and out of a superconducting state whenever it's temperature dependent resistance is changed, operated at 50 mK in the three detector arrays. SAFARI is split into two major components - the optics and the detectors in the cold 4.5 K focal plane unit, and the control and readout electronics in the SPICA service module.

SAFARI's large instantaneous field of view combined with the sensitive TES detectors will allow astronomers to very efficiently map large areas of the sky in the far infrared - in a square degreee survey of a 1000 hours many thousands of faint sources will be detected. A large fraction of these sources will be fully spectroscopically characterised by the instrument. Efficiently obtaining such a large number of complete spectra will be help further our understanding of how planets like those in our own solar system come into being, what is the true nature of out own Milky Way, and how do galaxies form and evolve?

The big advantage of the SPICA mission is the mechanically cooled mirror, providing sky background limited observations and allowing the usage of orders of magnitude more sensitive detectors. The characterization of such a sensitive imaging spectrometer requires the development of a dedicated facility: the Optical Ground Support Equipment (OGSE). The purpose of the OGSE is the verification of the optical performance of the instrument via aspects like radiometry and image quality. In Figure 1.2 the OGSE with all its subunits is shown. The Focal plane Unit (FPU) is mounted on the back side of the optical bench. The beam goes via the reimager to the OGSE space. The reimager, with properties similar to the spacecraft telescope, provides an accessible reimaged focal and pupil plane, which can be scanned. The beam can either be deflected via a flip mirror into a cryogenic calibration source or continues towards an XYZ scanner system with a pinhole mask wheel back illuminated by an integrating sphere. The extreme low background environment required by the ultra-sensitive detectors (few fW/pixel) demands the use of cryogenic mechanisms capable of operating at a temperature of 4 K. To meet these criteria a dedicated system has been designed.

In order to achieve these goals, extensive work has been done not only on designing and building the detector, but on its performance verification and calibration. This is the entire purpose of the AIV program and the motivation to design the necessary test equipment. Towards this goal, work has been done on the overall design and fabrication of the optical test bench as shown in



Figure 1.2: The SAFARI instrument is located underneath the optical test bench. The integrating sphere is seen at bottom middle. The image was taken from Klandermans (2013).

Figure 1.2 (Ferlet et al., 2009). The bench contains the integrating sphere located at the bottom center, a flip mirror designed to operate at 4 K, located top center, a light pipe which is connected to the integrating sphere, allowing external light sources to be connected to it, a signal source with several filters and an optical chopper is to the right of the flip mirror and to the left of the mirror are located a pupil scanner and an optical reimager.

Figure 1.3 shows how the entire test bench will be arranged within the cryostat. The SAFARI instrument is located underneath the bench and the cryostat itself has several inner compartments for increased thermal isolation, as the inner temperature must be maintained at approximately 4 K. Once the test bench has been completed it will be placed within the cryostat along with the SAFARI instrument. This will simulate the operating conditions of the instrument. This will allow for the complete characterization and performance evaluation of the SAFARI instrument. The absolute radiometric calibration process will involve the instrument to be illuminated uniformly by a light source with known characteristics. The idea on how to achieve this is by using a black body cavity with a small opening port. The hot source has been designed to behave like a Planck radiator with an emissivity coefficient close to 1. This coefficient indicates the radiation of energy from a body according to the Stephan-Boltzmann law, compared with the radiation of energy from a black body, which has a coefficient of 1. Achieving such an emissivity is physically impossible, but under certain conditions, such as the low temperatures and vacuum within the OGSE, it is possible to approximate such an emissivity coefficient. Unfortunately the intensity of the source is too high for the sensitive SAFARI instrument, hence the iris, shutter, and integrating sphere are used to dilute and equalize the light intensity. In Figure 1.4, the hot source is represented as component number (3).



Figure 1.3: Cut away view of the cryostat in which the test bench will be placed. The cryostat has several inner isolation containers in order to maintain the temperature gradient of approximately 300 K external and 1.7 K internal. The image was taken from Klandermans (2013).

The light leaving the hot source, of which the power and spectral distribution are given by Planck's Law, is only dependent on the temperature of the grey body (90 K). The spatial distribution of the Planck radiation can be represented as a Gaussian function, where the majority of the light intensity is concentrated around the geometric center of the optical path and decreases as it spreads towards the edges of said path. The light beam can be truncated with the use of a mechanical iris and a shutter as shown in Figure 1.5. Despite the apparent simplicity of the mechanical iris, it is worth mentioning that this device has been designed to operate at cryogenic temperatures with minimum heat dissipation.

The iris is also represented in Figure 1.5 as component number 4, whereas the shutter is represented as component number 2. The iris will be used to regulate the intensity of the grey body emission, whereas the shutter will be used to successively block and unblock the light passage, improving the signal to noise ratio during calibration. The hot source is connected to the integrating sphere with the help of thermal breaks, indicated as component number 5 in Figure 1.4. The hot source is connected to a 4 K cooler which has a higher cooling capacity, whereas the integrating sphere is connected to a 1.7 K cooler in order to create a very low background noise level. The thermal breaks prevent the intense radiation to escape through the coupling and scatter within the test chamber, resulting in increased noise levels during the calibration procedure. The light leaving the gray body cavity is temporally incoherent which means that diffraction effects are very weakly pronounced and occur very weakly on the edges of the physical components and thus can effectively be ignored.



Figure 1.4: Side cut of the calibration unit. Consists of a grey body radiator (3), the integrating sphere (1), mechanical iris(4) and shutter (2). The two components are coupled optically tight using thermal breaks (5) and the yellow components on the sides represent heat absorbers. The image was taken from Klandermans (2013).

To achieve the desired attenuation geometrical dilution can be used, which has been designed to be adjustable. Since the SAFARI instrument is a multipixel device, the entire field of view needs to be illuminated with a homogeneous intensity (Klandermans, 2013). The intensity of the light beam will be attenuated in a controlled manner using the iris and the baffles. The band equalization will be performed by the integrating sphere, shown in Figure 1.4 as component number 1, an instrument which distributes light in all directions equally, thus creating the necessary optical output needed for the accurate calibration of the instrument. The main challenge is to create an integrating sphere which can scatter the light efficiently and produce a uniform, spatially incoherent distribution, maintain the temporal incoherence distribution, that is any point in the output port should have the same spatial and spectral distribution of radiation, and lose as little energy during this process as possible. The internal surface of the sphere must be roughened in such a way as to reduce specular scattering and to maximize diffuse scattering. The difference being that specular scattering obeys the laws of reflection and this can create fixed light paths that do not distribute themselves in all directions. Diffuse scattering on the other hand does not obey the standard laws of reflection but approximates a Lambertian scatter, which means that the beam redistributes itself in all directions uniformly, if perpendicular to the scattering surface. At different angles, the diffuse scattering is more prominent in certain directions than others. It is well known, by those who know it well, that the Lambertian scatterer is a theoretical model, and as such can never be achieved, only approximated. Figure 1.6 shows clearly the difference



Figure 1.5: Mechanical iris used to control the optical intensity. The mechanical iris is actuated using electromagnets and has an inbuilt control loop which uses a Hall sensor for feedback. This allows for precise apertures to be set and kept constant. The image was taken from Klandermans (2013).

between specular and diffuse scattering.

Currently there are no known integrating spheres which have been designed to work within the SAFARI band. The chosen technique for roughening the inside of the sphere is sandblasting. This is a meticulous process with multiple critical parameters and poorly understood theory. In addition to this the aggressive environment which is created within the sandblasting chamber means that automation is very costly, thus a human operator is necessary. This makes the manufacturing of such devices both expensive and time consuming meaning that the trial and error approach is not preferred. This thesis aims to provide the basic tools needed to properly design an efficient integrating sphere. Thus the focus will not be on the integrating sphere itself, but rather on understanding the fundamental physical and manufacturing processes and attempting to model them. In this work the integrating sphere has been substituted with flat plates of the



Figure 1.6: Difference between specular and diffuse scattering. It can clearly be seen that the rougher the surface, the more diffuse the scattering. The image of c) represents what is known as approximating a Lambertian scattering - equal intensities in all directions regardless of the incident angle. Image was taken from Japan Association of Remote Sensing (1996).

same type of aluminum.

This leads to the main research questions:

Can the scattering surface fabrication process be analyzed and modeled using an artificial neural network? To answer this question, first a review of the existing theory is given, from which an approach will be undertaken. Then a surface analysis tool will be developed that will obtain useful statistics which will be used in predicting the light scattering patterns and also to train the neural network which will aid the future fabrication process. In addition to this a different approach to removing low frequency components from the surface profiles will be investigated.

Can a numerical simulation which can predict the scattering of an incoherent light band from 30 µm to 65 µm from a rough surface, modeled as a correlated Gaussian random process, be created and implemented? To answer this research question first a review of the existing theory is given, from which an approach will be undertaken. Then two different simulations will be investigated and compared - one using the rigorous solution of the wave equation in two-dimensional space and one which is designed to be a special case approximation.

In addition to this, an additional investigation was made whether independent component analysis can be used to analyze surfaces which have been processed twice under different conditions. The investigation was limited in nature and was aimed at obtaining some preliminary conclusions when applying the theory to the case of fabricating scattering surfaces using sandblasting.

In order to answer these research questions, first, the understanding of the underlying processes must be attained. The first to be investigated is the process of surface roughening for the creation of scattering surfaces. While the topic of impact erosion has received a lot of attention in the past few decades (Levkin et al., 1999; Abd-Elhady et al., 2006; Klinkov, 2005; Dang et al., 2013; Rao and Buckley, 1984; Dubey and John, 2013; Tian et al., 2007; Doja and Singh, 2012; Ripken, 1969; Han et al., 2008), little attention has been given to trying to predict how surface roughness changes depending on the impact conditions. Research has been done in the area of obtaining surface profile descriptive statistics (Gascon and Salazar, 2011; Zhenrong et al., 2010), however little progress has been made at modeling the deformation of surface profiles due to erosive processes such as sandblasting (Khorasanizadeh, 2010; Slatineanu et al., 2011; Arokiadass et al., 2011; Kamely et al., 2011; Tavares, 2005). The majority of research falls into two categories statistical analysis employing Design of Experiments and/or ANOVA (Slatineanu et al., 2011; Kamely et al., 2011; Arokiadass et al., 2011), and deterministic analysis based on mechanical physics (Tavares, 2005; Dubey and John, 2013; Evans et al., 2000). Creating rough surfaces with known parameters is important for several industries such as optics (Zhou et al., 2011), adhesives (Khorasanizadeh, 2010), and machining (Kleinedlerova et al., 2013) and so a more robust set of predictive tools must be investigated.

The second major process which will be investigated is the light scattering. Unlike the case with surface roughness analysis and prediction, an incredible amount of research has been done on light scattering theory starting from the early 20<sup>th</sup> century up until the present day. A plethora of theories based on various electromagnetic approximations have been developed and numerically validated (Tsang et al., 2001; Torrance and Sparrow, 1967; Maradudin, 2007; Harvey and Shack, 1978; Mischenko et al., 1999; Mie, 1908; Schuerman, 1980). Due to the mathematical diversity of the different theories, an equally large field of numerical techniques has been developed alongside (Jandhyala et al., 1998; Du and Liu, 2009; Burghignolin et al., 2002; Sanchez-Avila and Sanchez-Reillo, 2002; Sun, 2006; Nakajima et al., 2009; Nasser, 2013; Ciarlet and Zou, 1999; Garg, 2008; Ottusch et al., 1998; Hamilton et al., 1999). In this research the method of moments (MoM) technique applied to the scalar wave approximation theory and the small perturbation method (SPM), following the work of (Tsang et al., 2001), were tested against experimental data.

#### 1.1 Previous Work

Currently very little work has been done in terms of analytically investigating the phenomenon of light scattering at SRON. An attempt at creating a reliable light scattering simulation has been made, however it was aimed at modeling the entire process within an integrating sphere using ray tracing techniques (Klandermans, 2013) using the commercial software ZEMAX (Normanshire, 2012). In addition to this the results obtained from this simulation were not conclusive enough and failed to predict experimental results (Klandermans, 2013). Attempts have been made at manufacturing an integrating sphere using sandblasting, however due to the poor understanding of the parameters influencing the process the scattering properties of the sphere were subpar to expectations. Very little work has been performed in terms of understating the process of sandblasting, with only a few Dektak (Nanotech, 2009) measurements of aluminum plates which have been processed under varying parameters (Ferrari and Panman, 2013). The effects of varying

#### CHAPTER 1. RATIONALE

the air pressure, exposure time, distance, and grain size have been investigated independently, but their combined effects were not. The measured profiles are only 2 mm long and thus do not provide sufficient statistical information for any further analysis currently. In addition to this no work has been done on creating a manufacturing procedure for the rough surfaces.



Figure 1.7: Example profile measurement. The surface has not been detrended. The y-axis is in nm while the x-axis is in µm for the purpose of distinction.

Figure 1.7 shows one such profile measurement. As it can be seen, the profile is slowly varying despite the fact that 18000 measuring points have been recorded, which means that it is not possible to obtain accurate descriptive statistics. Despite the high resolution, there simply isn't enough detail to obtain a proper understanding of the surface topography.

According to numerical investigations into light scattering and early experimental work performed at SRON, it has been concluded that higher roughness and specific correlation length will result in better scattering of longer wavelengths in the micrometer range, as illustrated in figure 1.8. And indeed it can be seen that the specular normalized intensity from 30 µm to 100 µm is less than 0.1. This can be interpreted in several ways - either the rough surface is scattering excellently, or the absorption losses are very high, or both effects are prominent to a certain extent. It is also worth mentioning that the intensity spikes between 10 µm and 30 µm are due to photon noise. Absorption from rough surfaces has been addressed by Bergstrom (2008), however his research focused on lasers. Losses have also been studied in cryogenically cooled environments (Finger and Kerr, 2008) where the anomalous skin effect has been taken into consideration. Unfortunately, such research is out of the scope of this work. Other techniques such as spark erosion have been considered, but no experiments have been performed because the process is difficult to scale.

The structure of this thesis is organized in the following manner. Chapter 2 will focus entirely on the surface analysis. First the situational and theoretical analysis will be presented, showing what progress has been made in the field. Following this, the conceptual model will be put forward, in which the development of a surface analysis tool will be described in detail. Following this, in the research design section of chapter 2, the choice of experiments and experimental procedures will be discussed and defended. Finally, the research results will be analyzed and



Figure 1.8: FTS6000 (Online, 2014) measurement of two samples processed with F-8 and F-12 grain sizes (Abrasives, 2013). The x- and y-axis are logarithmic.

discussed. Chapter 3 will focus on the development of the artificial neural network that will facilitate the fabrication of scattering surfaces. The contents of this chapter are identical to those of chapter 2 - an introduction to the topic will be given, followed by a conceptual model, and finally the design of experiments and the corresponding results will be discussed. Chapter 4 will focus on the development and evaluation of a light band scattering tool based on the MoM and SPM techniques from (Tsang et al., 2001). The chapter's layout is identical to that of the previous chapters. Chapter 5 is dedicated to performing a preliminary investigation of whether more complicated surface profiles, such as one that has been processed twice under the same conditions and with different sized grains, can be analyzed using ICA (Naik and Kumar, 2011). Once again, the chapter's layout is identical to the previous chapters. In chapter 6 a discussion will be given on the difficulties of creating a 3D simulation. Finally we will finish with recommendations a conclusion and for future work in chapters 7 and 8, respectively.

### Chapter 2

### Surface Analysis

In this chapter the development of a surface analysis tool written in Matlab is presented. The purpose for creating such a tool is to have a means of obtaining meaningful statistical information directly from a surface profile measurement, which can be used in conjunction with the light band scattering simulation or simply as a means to compare different fabrication techniques and their various control parameters.

#### 2.1 Situation and Theoretical Analysis

Sandblasting, as the name clearly suggests, is the process of blasting, or bombarding, a given material with a continuous stream of small, hard particles. The first machines used sand, hence the name, however with the advent of technology various other materials have become available, the most common today being SiC. From an industrial point of view, sandblasting is useful for cleaning up welds and preliminary polishing of malleable materials such as metals. Lately sandblasting has been applied to the optics industry (Zhou et al., 2011) as a fast and cheap way of creating surfaces with certain optical properties, improving the performance of LED screens. In addition to this the process has been used in the study of adhesive strengths of steel pipe coatings (Khorasanizadeh, 2010). Research like this is generally used to improve the design of adhesive coatings allowing a wide diversification based on the different particle flow conditions encountered in various pipes. In geology the process of sandblasting has also been recognized as an important process in dust production and climate modeling (C. et al., 1998). And although the process is fairly primitive, very little work has been done in terms of modeling it properly. A further added complexity of using this process is the difficulty of automating it. The aggressive environment created within the sandblasting chamber makes the employment of automation a somewhat daunting task due to the environmental complexity. This means that either a static fixture must be used, such as the one used in (Patel, 2011; Rao and Buckley, 1984), or a human operator must control the process. The involvement of a human operator leads to the introduction of noise into the experiments, as maintaining the same distance and orientation while moving one's hand at a steady rate over the entirety of the sample without trembling is highly unlikely. Even if such mastering of one's hands is possible there is also the problem of the variance of grain sizes. Currently the Federation of European Producers of Abrasives (FEPA)(Abrasives, 2013) classifies grain sizes according to a mean diameter. This suggests that the flux of particles leaving the nozzle is fluctuating and as such can be considered as another source of noise. Another source of noise which must be considered is the velocity variations caused by the carrying medium. In certain circumstances the carrying medium is water, however during this research an air operated sandblasting machine was used. When buffer tanks are added to dampen the pressure oscillations the variance does decrease, however it does not disappear completely. For certain engineering practices this is of little concern, however for the sake of completeness these sources of noise deserve to be mentioned.

To this day a lot of research has been conducted in attempting to model and understand the process of erosion of metals, especially in pipeline systems. Despite the numerous investigations, very little attention has been given to fundamentally understanding how changing the roughness of a metal changes it's behavior within a system. The changes in roughness are most frequently simply described statistically and little discussion is given as to the mechanisms that form them (Foldyna et al., 2013; Vigolo et al., 2013; Doja and Singh, 2012; Miyoshi et al., 2004). This clearly indicates that the process of surface deformation and roughening is a very complex one. It is, however, puzzling that very little attempts have been made at applying any sort of mathematical analysis. The most common approach has been to apply standard statistical analysis tools directly to the problem, with the results often being considered as special cases related to the specific problem at hand and not a general treatment. The problem can be reduced to two sub-problems which have both been partially addressed, yet no bridging between them has been made. This is mostly due to the nature of the two problems - the first one being of a purely mechanical nature, describing the collision mechanics, whereas the second one being of a mostly statistical nature, describing the distribution of impact material. In terms of modeling the collision mechanics, the first attempt was made by Issac Newton when he observed that regardless of the speed a projectile is going at, if it collides with a surface that has the same density, then it will only travel approximately one body length before it stops (Young and Laboratories, 1967). The theory has then been expanded upon resulting in the Hertzian theory of non-adhesive elastic contact, the Johnson-Kendall-Roberts model of elastic contact, the Maugis-Dugdale model of elastic contact, the Bradley model of rigid contact, and Derjaguin-Muller-Toporov model of elastic contact being some of the most prominent ones (Johnson, 1987). The problem of analyzing the distribution of impact has been addressed by the work of Sidorchuk et al. (2004). The lack of bridging between these two fields is the lack of a mechanism which would predict the behavior of the surface material when being struck more than once. In addition to the analytical models, FEM simulations have been used to study shapes which diverge from the assumptions used to derive the above mentioned models (Negrea and Predoi, 2012).

#### 2.2 Conceptual Model

Analysis of Dektak profile measurements is made difficult due to the necessity to separate the drift component from the "rough" one. The drift component is the slow varying drift found in most direct measurements and could be due to a systematic error within the measuring device, such as tilts and other offsets within the measuring system, or due to the sample imperfections. The physical interpretation of removing the drift component means that the surface will not be able to scatter very long wavelengths, but will reflect them. In the case of the drift component being linear, the detrending process simply aligns the surface profile with a given axis on interest. There are several standards which can be followed (Tavares, 2005), however there is one flaw in adopting them. They represent a set of filters used to eliminate the low frequency components present within the sample, however they are based on the assumption that the low frequency component remains the same for the entire sample. However as one might imagine, when cutting

any material, tensions form along the surface, with their distribution rarely being uniform. Such tensions will always arise even if the cut was perfectly parallel to the surface. This was especially true for the aluminum samples used in these experiments. Their thickness was approximately 3 mm, which meant that even before the sandblasting experiments were carried on them, there would be a low frequency drift component. In addition to this, due to the clamping system used within the sandblasting chamber, there would be additional torsion from the air pressure as well. The final result is a very complicated profile for which there is no guarantee that it can be predicted by the filtering standard. In order to confirm this reasoning several statistical tests were incorporated into the analysis tool. It is worth noting that the only results from the surface analysis tool which are being used by the scattering simulation are the standard deviation, the correlation length, and the Laplace diversity. The rest of the analysis is used to confirm that the earlier mentioned parameters are accurate, and to give an overall insight into the surface profile. The results of these tests will be addressed further in this chapter. The flowchart of the surface analysis tool is shown in figure 2.1.

Thus a different approach was used to detrend the surface measurements. Originally the use of an extended Kalman filter was considered and implemented, however the problem of adjusting the filter accordingly remained. Interestingly enough, the area of surface analysis shares very little theory with the field of time series analysis, yet it is noticeable that there are similarities. In their work, Koopman et al. (1999) have developed a free to use software written in C which is capable of performing filtering and smoothing. An example application of the software package can be found at the end of Chapter 6 in Durbin and Koopman (2012), which has several fascinating chapters on filtering and smoothing.

The detrending technique which was adopted for the surface analysis tool was to simply fit a  $2^{nd}$  order polynomial onto the data. The difference between the original measurement and the fitted polynomial represents the "rough" profile, with the drift component being suppressed at a certain number of lags. This approach can clearly be improved following the work of Durbin and Koopman (2012) on smoothing and filtering. The main goal was to utilize a tool which is self adjusting and not user dependent. Thus, when the processed sample is measured several times at different locations, the same filter can be applied directly. And it has been shown that measured profiles taken at different locations within the sample can have very different drift frequencies. This choice reduces the risk of losing surface information due to improper use of the filters set by the standards. The use of a  $2^{nd}$  order polynomial as much as possible without creating a straight line. This approach causes the stationarity tests to reject stationarity at the first couple of lags, however the polynomial manages to detrend the surface successfully within 100 lags. This is one definite flaw in the current design, however the errors from this approach have been analyzed and addressed.

Following the detrending, the rough profile's partial autocorrelation is evaluated for 100 lags. Together with the normalized autocorrelation function developed by Bergstrom (2008) it serves as an optical indicator to determine whether the detrending is good or bad, as an autoregressive (AR) model is created from the detrended data and extrapolated for 5 million data points. Should the AR model be unstable, several parameters will indicate this. The partial autocorrelation function also serves as an indicator for the appropriate number of lags that are necessary for the creation of the AR model.

The next stage of the analysis is the creation of an AR model with 10 lags applying the Burg

## SURFACE ANALYSIS



Figure 2.1: Flowchart of the surface analysis tool

method (Kay, 1999). The Yule-Walker equations were also investigated, however the predicted surfaces had a noticeable high frequency component, which was not present in the original data. For this reason the Burg method was chosen. The number of lags present in the AR model were also arbitrarily chosen based on the observation that all surfaces could be accurately modeled with 8 parameters at most. In the case when the data series can be modeled with less than 10 coefficients, their values are simply set to very low values.

A simple time series analysis is also included in the form of differencing on the data and it was observed that there is a truncated Laplacian noise component. The rationale behind this statement is that differencing can loosely be considered as the limit case of a 1<sup>st</sup> order continuous-time RC high-pass filter as  $\alpha$  approaches 0, which translates to the filter having a very high cut off

frequency. The information obtained by this analysis showed that the surface generation models used by Bergstrom (2008) and Tsang et al. (2001) could be improved upon slightly.

Following the creation of the AR model, histograms of the detrended data series, the AR extrapolated series, and the differenced series are computed. This information provides a quick visual inspection as to what distribution the data might have. When the data series is small analyzing a histogram is useless, however should large data series be analyzed, this tool becomes useful.

The normalized autocorrelation function (ACF) (Bergstrom, 2008) is computed for both the detrended series and AR model. The ACF is an indicator of error of the detrending process. Should the detrending be poor, the ACF function will have a noticeable low frequency sinusoidal component. Additionally the ACF can be used to investigate the presence of noise. The most famous example is the analysis of the Brusselator, a theoretical model of an autocatalytic chemical reaction (Gaspard, 2002), with an ACF when the model is subjected to noise. As the noise increases, the ACF converges towards 0 faster, which is the equivalent of observing a noisy frequency spectrum.

In addition to all the visual statistics, the Kwiatkowski-Phillips-Schmidt-Shin (KPSS), augmented Dickey-Fuller (ADF), Kolmogorov-Smirnov (KS), and Jarque-Bera (JB) tests have been included (Freedman et al., 2007). They provide a more trustworthy analysis of the detrended series than simply relying on visual inspections. The KPSS tests for stationarity by performing a regression to find the ordinary least squares fit between the data and the null model. The results of this test are also a measure of how successful the detrending technique has been given several measurements of the same sample. The KPSS test employed in Matlab uses tabulated data to evaluate the critical values and the p-values. The KPSS test is evaluated from 10 to 100 lags with a step of 10 lags. This provides a more detailed assessment of the detrending performance. The KPSS test plays an important role in demonstrating the varying drift frequencies present within a single sample.

The ADF test determines whether the given data series have a unit root or not. This test is used in conjunction with the KPSS test for robustness. The ADF test is set to 100 lags in order to avoid the correlation introduced by the size of the grain. In addition to the test rejection decision the p-value and test statistic is also evaluated.

The KS test is used to test whether the data series follows a standard normal distribution. It works by comparing the empirical, or test, empirical cumulative distribution function with a reference one, which can by of any kind as long as it can be computed. The implementation of this test has given strange results, as it produced results which were conflicting other tests. Additionally to the KS test, the JB test is also performed which also agrees with the KS test.

#### 2.3 Research Design

The sandblasting experiments were performed inside a Skat Blast 310 machine with a large tungsten nozzle. A sample holder was constructed which would hold 3 samples at a time. Two types of samples were processed. A  $10 \times 10 \times 0.3$  cm type and a  $3 \times 6 \times 0.3$  cm type, which will be referred to as type 1 and type 2, respectively. The type 1 samples were used for the band scattering measurements, whereas the type 2 samples were used for the modeling of the

sandblasting process. The SiC grain types were F-8, F-12, and F-16 with mean diameters of  $2460 \,\mu\text{m}$ ,  $1765 \,\mu\text{m}$ , and  $1230 \,\mu\text{m}$ , respectively.

The orthogonal array experimental design proposed by Taguchi can be used to provide insight into the influence of various parameters on a system's performance in a reduced set of experiments. Once the parameters affecting a process that can be controlled have been determined, the levels at which these parameters should be varied must be determined. In an optimum situation, determining the resolution of a variable to test requires a proper understanding of the system's capability and performance. In the case of fabricating scattering surfaces using sandblasting the only thing that was known were the maximum and minimum values each control parameter could take. In addition to this, each parameter can differ in terms of what is a maximum and a minimum, thus one is presented with the choice of either keeping the same parameter resolution or to restrict all parameters to a fixed number of experiments. Often it is easier to go for the latter choice, as was done in the current case. Also, the cost of conducting experiments must be considered when determining the number of levels of a parameter to include in the experimental design. Knowing the number of parameters and the number of levels, the proper orthogonal array can be selected (Fraley et al., 2007).

The control parameters of the sandblasting machine were air pressure, nozzle distance from the target, exposure time, and grain size. It was chosen to have a resolution of 4 levels per variable, except for the grain sizes which were restricted to only 3. This would result in a total of 192 experiments which would take too much time and resources to process properly. Instead, Taguchi's orthogonal arrays were applied to reduce the number of experiments to 16, a more modest number. Such a size reduction is not without consequences, of course. This number is the bare minimum necessary to map evenly the entire system at the desired levels. The reduction of experiments also translates to a reduction of available information, a consequence which has also been addressed. The experimental arrangement is of a L'16 array and is shown in table 2.1.

Experiment	Pressure (bar)	Distance (cm)	Time (s)	Size (F-number)
1	3	4	60	F-16
2	3	5	90	F-12
3	3	6	120	F-8
4	3	7	150	F-16
5	4	4	90	F-8
6	4	5	60	F-8
7	4	6	150	F-16
8	4	7	120	F-12
9	5	4	120	F-16
10	5	5	150	F-8
11	5	6	60	F-12
12	5	7	90	F-16
13	6	4	150	F-12
14	6	5	120	F-16
15	6	6	90	F-12
16	6	7	60	F-8

Table 2.1: Taguchi's orthogonal array scheme for the type 1 sandblasting experiments

There were 3 samples per experiment in order to investigate the difficulties of distributing the particle jet evenly. One major problem that was encountered during the process of experimentation was that the F-8 particles were too large for the nozzle. Unfortunately no larger nozzle could be obtained that would fit the Skat Blast 310's pressure hose. For this constant mechanical agitation, in the form of kicks and hits, had to be applied during the processing of experiments 3, 5, 6, 10, and 16. This is an experimental flaw which could not be circumnavigated and so these experiments were repeated several times until an optically even roughness was achieved, which was similar in appearance to that of the rest of the samples. Since there were only 3 types of grains, the missing elements of the orthogonal array were randomly filled with one of the existing grain types (Fraley et al., 2007).

The type 1 samples were used both for the band scattering measurements and for the evaluation of the ANN. They were processed using random parameters which were still within the established boundaries of the L'16 array, except for experiments 4 and 5 which would have much lower processing times in order to test whether the ANN could accurately extrapolate their parameters. Their control parameters are as follows

Experiment	Pressure (bar)	Distance (cm)	Time $(s)$	Size (F-number)
1	4.5	5	75	F-16
2	3.8	7	110	F-16
3	5.2	6	125	F-12
4	5	5	20	F-12
5	4	4.5	40	F-12

Table 2.2: Control parameters of the type 2 sandblasting experiments

The F-8 grain size was not used on the type 1 samples due to the difficulty of maintaining a constant jet of particles over a larger surface area. The statistical analysis of such samples would be erroneous and such surfaces could not be modeled within the scattering simulations, and as such they were rejected. In addition to this two mixed grain experiments were performed in which type 2 samples were sandblasted with two different grain sizes consecutively. The control parameters were kept constant for each, decreasing the difficulty for the ICA analysis tool.

The experimental procedure was to keep the nozzle perpendicular to the surfaces while quickly moving it over them. As it was known that there would be air pressure variations, the fast distribution of the particle jet would ensure that the errors would be evenly distributed over the entire surface and not clustered in specific areas. After 3 experiments the grains would be replaced by new ones in order to mitigate the effects of reducing the grain's mean diameter and the sandblasting machine was carefully cleaned whenever a different grain size would be used in order to avoid contamination.

The sources of noise in this set of experiments are many. The biggest one being the human operator which performed the experiments. The control parameters were monitored by eye and as such it is expected that the error is not insignificant. This is also the main argument for selecting ANNs for mapping the control. Their associative and robust memory allows for noisy inputs to be predicted accurately, but they still require low noise training sets. Unfortunately this is currently the only way surface roughening can be performed at SRON.

The profiles were measured at the Faculty of Mathematics and Natural Sciences, University

of Groningen. They were measured at 4 different locations using the same instrument settings and stylus. The Dektak software did include detrending options, however the inner workings of these options were unknown and as such were ignored.

The choice of the 4 measuring locations were chosen to be as equally spaced as possible. Two measurements were taken along the length of the sample and the other two were taken along the width of the sample. In such a way it would be possible to measure the surface roughness ergodicity. The directions were measured in pairs for statistical significance, should there be a large difference between any two measurements. The Dektak profilometer unit is located in a clean room and no photos were taken of it.

#### 2.4 Research Results

We begin the surface analysis by presenting the surface results for experiment 1 from the type 2 samples. The rest of the results will be summarized in tables for compactness. As it can be seen from figure 2.2, the quadratic polynomial manages to fit the apparent drift component easily. The resultant detrended surface is then determined by subtracting the raw profile with the evaluated polynomial. It can be seen that there is no apparent sinusoidal behavior, so by first glance the detrending looks successful. The normalized ACF function, the KPSS, and ADF tests will confirm this as well. Some surface profiles have more complicated drift components than this one, however the quadratic polynomial is still more than capable of removing it.



(a) Raw profile with the detrending polynomial evaluated using the least squares measure, shown in green



(b) Detrended profile, optical inspection shows no apparent sinusoidal components. ACF is a better measure of periodicity

Figure 2.2: Example of proper fabrication and detrending

The difference between the two profiles is uncanny, yet they represent two measurements at different locations within the same sample. This is a clear demonstration of how easily a mistake can be done when creating deterministic filters. If the filter was adjusted to remove the drift component of figure 2.2 it would fail to remove the one in 2.4, however if the filter was trained on the second profile, it would overfilter the first profile.

This is also a clear indication of faulty processing. In this case, experiment 10 was done using



(b) The surface profile is uneven, a clear indication of faulty processing



grain size F-8. Next we consider the partial and normalized ACF of the profile in figure 2.2

The most noticeable difference between the partial and the normalized ACFs is that the partial ACF decays much faster than the normalized one. This suggests that an AR model can be created, which will not diverge. The normalized ACF is decaying which means that the noise structure is predominant, does not seem to have long periodic oscillations which is a visual confirmation of a successful detrending. The somewhat periodic components with different amplitudes are a direct result of the high measured resolution. These measurements contain 36000 data



(b) Normalized ACF, evaluated over the entire measured length

Figure 2.4: Comparison between partial and full ACFs

points over a surface profile length of  $3 \,\mathrm{cm}$ . For an F-8 grain type, with an average diameter of  $2460 \,\mu\mathrm{m}$ , and given the geometry of the grain, it becomes less of a surprise that such large correlations exist.

Next we explore the creation of an AR model based on the detrended data. The model can be used to extrapolate surface statistics, confirming the hypothesis that the surface profiles follow a Gaussian distribution.



Figure 2.5: AR model evolved for 5 million data points, following the same resolution as the measured profile

The first thing to notice in figure 2.5 is the stability. The AR model does not diverge even after 5 million data points, which was to be expected. Subsequent analysis of this extrapolation can predict the complete distribution shape of the data, which in all cases was found to be Gaussian.

Next we consider three histograms - of the detrended profile, the AR model, and the differenced detrended profile. As discussed earlier, differencing can be broadly considered as a discretized first order passive high pass filter, who's control parameter  $\alpha$  approaches 0. Thus, differencing gives us the highest present frequencies. For compactness the plot of the difference is omitted, as we are only interested in the difference's distribution.

There have been several investigations into light scattering from micro cracks (Germer, 2001), however from a numerical and programming standpoint it is easier to work with perfectly smooth Gaussian surface profiles. The choice can be justified if one considers the necessary surface resolution involved in the simulations arguing that if such effects were included the memory requirements and computation time would become too big, however this is clearly unrealistic. Despite this, the simulations remain accurate to measured data (Tsang et al., 2001; Bergstrom, 2008). In both of these cases, the type of sensing is called active sensing, which has more relaxed accuracy criteria. In the field of passive sensing the accuracy restrictions are as high as up to 1%. In these cases it is definitely worthwhile to investigate in the development of more accurate surface models. Currently the most common surfaces being used are Gaussian, ocean spectrum, and fractal (Tsang et al., 2001), however the Gaussian case is physically impossible, as there is no such thing as a smooth surface in a manufacturing process.



(a) Detrended profile histogram which shows an indeterminate distribution







(c) Detrended profile difference histogram showing a truncated Laplacian

Figure 2.6: The induced most likely original signals

The detrended profile appears to have a somewhat Gaussian resemblance, however once the AR model extrapolation is evaluated the Gaussian distribution becomes much more apparent, of course if the AR model is evolved for two orders of magnitude less points, the distribution would resemble that of the detrended profile. Thus it appears reasonable to assume that sandblasting produces a Gaussian random rough surface. Of course the limit case would be to process the sample with much harder grains at much higher pressures. Then the distribution will might become a truncated Gaussian. Naturally, not all measurements had such apparent Gaussian distributions. However all the AR models did converge to a Gaussian distribution with different standard deviations, as expected.

The bottom histogram is that of the differenced detrended profile. It closely resembles a truncated Laplacian distribution. Several conclusions can be drawn from this observation. The most obvious one is that this distribution has a much larger population than the Gaussian one, as the probability distribution function's (PDF) shape is better defined. The second conclusion is that this is most likely the distribution of micro cracks along the surface, due to the fact that the PDF is truncated on the positive side. The physical interpretation is that a micro crack, or micro crater, is more likely to have a deeper center than the height of the edges.

Next we investigate the final visual analysis. The results from this test are intriguing because the KS and JB tests both reject their null hypotheses, yet the current visual test shows otherwise. As it can be seen in the upper half of figure 2.7, the empirical, which in this case is the measured, cumulative distributive function (CDF) and the CDF of a standard normal distribution overlap very well. This could be explained by the fact that the detrended profile is highly correlated. The KS and JB tests fail to reject their null hypotheses when a every 100<sup>th</sup> data point is skipped, however it was considered tampering too much with the data at hand.

Finally we consider the results of the KPSS, ADF, KS, and JB tests for the current surface profile. The KPSS test is evaluated for every 10 lags starting from 10 and stopping at 100. It provides a measure of how well the surface profile has been detrended. The null hypothesis is that the process is trend stationary, which means that there is a definite trend, even around the 0, that is being followed. The alternative is that the process is difference stationary, which means that there are no trends whatsoever present, meaning that the series in unstable. For the measurement that has been analyzed above the KPSS test's decision is that between 10 and 50 lags the process can be considered to have a unit-root, that is the null hypothesis is rejected. Above these lags the test fails to reject the null hypothesis, which means that the surface profile is most probably trend-stationary. This is a direct measure of how successful the detrending has been. When evaluating the surface statistics taken at a location from a sample, in one of the cases the KPSS test reports that at all lags the process can be considered to have a unit-root, while a different location on the same sample would be considered trend-stationary within 20 lags . This confirms the previous statements about the difficulties present when designing filters that should detrend a surface. The confidence interval was set to 95% for all lags.

As an example, the KPSS test hypothesis decisions for type 2 sample 2 are shown in table 2.3 where it can clearly be seen that the entire  $y_1$  measurement has been considered a unit-root process. The contrast between the other measurements, especially between  $y_1$  and  $y_2$  is indicative of the complex nature of the drift component. However the ADF tests does reject the null hypothesis of a unit-root for the case of  $y_1$ . This is an example of why it is important to never rely on a single measure.



Figure 2.7: AR model evolved for 5 million data points, following the same resolution as the measured profile

Lags	$\mathbf{x1}$	$\mathbf{x}2$	y1	y2
10	1	1	1	1
20	1	1	1	1
30	1	1	1	1
40	0	0	1	1
50	0	0	1	0
60	0	0	1	0
70	0	0	1	0
80	0	0	1	0
90	0	0	1	0
100	0	0	1	0

Table 2.3: KPSS test decisions for sample 2 measured at 4 different locations. 1 represents a unit-root process, whereas 0 represents a trend-stationary process.

The ADF test also had a confidence interval of 95% and was evaluated for 100 lags only. For every single profile measurement the test rejected the null hypothesis of a unit root. The p-value

and the critical value were found to be 0.00100 and -3.41230, respectively.

The KS test completely rejects the null hypothesis of the profile data having a standard normal distribution, even though the profile data was z-scored, with a p-value of 0. The Jarque-Bera test also rejects the null hypothesis of observing something close to a normal distribution, with a p-value of 0.001. However, when the sample size is reduced by skipping every 100<sup>th</sup> sample point, both tests fail to reject the null hypothesis of having data from a normal distribution (standard in the case of the KS test). However in these cases failure to reject the null is not a proof that the null is even approximately true at the population level. This is because with small samples these tests have low power. As the sample size grows both tests begin to reject the data, and they should, unless it has a perfect and exact normal distribution. And so, even if the histogram of the AR model or the normality plots of the detrended data or the AR model show nearly perfect similarity to a normal distribution, the fact that there are so many data points causes the two tests to completely reject the null hypothesis.

In order to investigate this claim a simple investigation was performed - a correlated Gaussian surface profile was generated using the same resolution for 10 million data points. This profile was also analyzed as a regular surface profile and despite the nearly perfect visual inspections, the KS and JB tests once again rejected the null hypothesis. For the sake of completeness, a set of 10 million normally distributed random numbers analyzed as well, and this time the KS and JB tests did not reject the null hypothesis.

The same analysis has been performed for every measurement which results in a total of 64 analysis being done. The analysis tool saves all the test information into a designated folder, thus making the analysis of all the information easier.
# Chapter 3 Manufacturing Control Procedure

This chapter deals with the development of a manufacturing control procedure using an artificial neural network (ANN). It has been shown that ANNs are very potent tools for predicting surface roughness parameters in machining (Suresh et al., 2002; Ozel and Karpat, 2004; Benardos and Vosniakos, 2003), however currently they have not been applied to the process of sandblasting. The rationale is to try and introduce some form of control when scattering surfaces are being fabricated. To help achieve this an ANN is employed which has been trained using Taguchi's orthogonal arrays scheme. The benefit of this training choice was that it significantly reduced the number of necessary experiments, which is also it's main drawback. The ANN will serve in aiding the human operator who will perform the fabrication by suggesting which parameters will yield good results. The research design is related entirely to that of chapter 2, and in order to avoid repetition, has been omitted from this chapter.

## 3.1 Situational and Theoretical Analysis

Neural networks work on the principle of establishing connections between core elements, analogous to the neuron, which operate in unison, all dependent on each other. Such networks, despite their apparent simplicity, can be trained to perform a given task or tasks, depending on the design. Commonly neural networks are trained so that a particular input leads to a specific target output. The most common type of learning is called the supervised learning in which the connection weights are adjusted so that an input signal produces a specific output, with minimal error. Another popular training technique is called batch training, which proceeds by presenting the network with an entire set of input parameters and output targets. The sets are then randomly broken several times into two subsets, one serving for training and one serving for testing. The batch process is iterated several times until the error between the output targets and the network's predictions is minimal relative to the previous iterations. Incremental training is popular in areas where data logging is unfavorable. Such networks improve over time as more and more data is passed through them. This type of training is referred as either "adaptive" or "on line". Several types of training functions have been developed which all serve as a means to asses the error between the network's output and the target output. These algorithms then adjust the corresponding weights within the network in an attempt to minimize said error (Wilde, 2010).

The elegance and simplicity of the neural network can be demonstrated immediately. The first part of figure 3.1 represents a single neuron with an input vector  $\mathbf{x}$  with three elements  $(x_1, x_2, x_3)$ , a weight matrix  $\mathbf{w}$  with three elements  $(w_1, w_2, w_3)$ , a summing operator, and a



Figure 3.1: An example of a single neuron and a complete neural network. Details have been omitted for the sake of generality

transfer function f. Different transfer functions are used depending on the design of the network. Thus, the operation of the neuron can be represented in the following manner, with the vector multiplication representation being more compact. Should the output be a vector and not a scalar, then it is preferable to use vector notations.

$$y = f\left(\sum_{i=1}^{N} x_i w_i + b_{output}\right)$$
(3.1)

Where N is the size of the input vector  $\mathbf{x}$ . It should also be noted that sometimes the transfer function  $\mathbf{f}$  and the bias are omitted in diagrams in favor of aesthetics. Such is the case with the

complete network in figure 3.1.

Thus, for a complete neural network with five input neurons, three neurons in the hidden layer and a single output neuron, the complete equation can easily be written down in the following manner:

$$y = f\left(\mathbf{g}\left(\mathbf{x} \cdot \mathbf{W}_{hidden} + \mathbf{b}_{hidden}\right) \cdot \mathbf{W}_{output} + b_{output}\right)$$
(3.2)

With the proper choice of transfer functions and network topology, the neural network can accomplish incredible tasks. However the true power of these systems lies in the fact that they posses associative memory. This allows them to still evoke a proper response, even when the input signal is noisy or incomplete. The mathematical analysis of neural networks is still a growing field in linear algebra. The author of (Wilde, 2010) gives a wonderful summary of the possibilities and limits of various neural networks.

The origin of the attempts to model an artificially neural network (ANN) can be traced back to 1943 when a paper authored by neurophysiologist Warren McCulloch and mathematician Walter Pitts tried to model the working mechanism of a neuron. Later on the concept was further developed by Donald Hebb who reasoned that neural pathways are strengthened each time they are used. The further development of computers throughout the mid 20<sup>th</sup> century allowed prominent scientists such as Nils Aall Barricelli, Bernard Widrow, Marcian Hoff, Teuvo Kohonen, and James Anderson to further develop the concept. Despite this interest was lost temporarily into the field up until the early 1980s, when John Hopfield demonstrated that ANNs with bidirectional connections are more powerful than the original architectures. This has allowed neural networks to be applied to various problems ranging from voice recognition to control of robotic actuators. When properly applied and trained ANNs can be used as estimators to multidimensional functions for which analytic derivation is either unfeasible, impractical, or simply impossible (Mehrotra et al., 1996).

Their strength, however, lies in the ability to recognize and classify patterns. As they are modeled around mimicking the human mind, it becomes clear that ANNs are unsuitable for performing numerical calculations, as neither are we. Despite this, with proper design and training ANNs have found an application in predicting surface roughness in end milling and wire electrical discharge machining (Esme et al., 2009; Colak et al., 2007; Ozel and Karpat, 2004). The results from these investigations have shown that ANNs are more flexible and more capable than the current empirical and semi-empirical models used in these areas. They are fast and easily implemented into environments which are generally considered difficult to study in detail (Rashid and Lani, 2010; Ali and Dhar, 2010; Aguiar et al., 2008; Suresh et al., 2002). For an in depth review of the current applications of neural networks to the field of roughness prediction the reader is directed towards Benardos and Vosniakos (2003).

## 3.2 Conceptual Model

The design of the ANN was aimed towards creating a universal function approximator which is a combination of a tan-sigmoid transfer function in the hidden layer and a linear transfer function

in the output layer. Such a network would then be trained to represent the forward dynamics of the plant. The opposite is impossible to achieve, as a NN should not have more outputs than inputs. The prediction error between the plant output, which in this case is the sandblasting machine, and the NN output is used as a training signal.



Figure 3.2: Example neural network used as a plant controller. u represents the control inputs,  $y_p$  represents the analyzed fabricated samples, and  $y_m$  represents the network's output. Image recreated from Demuth and Beale (2002)

The neural network used previous inputs and outputs to predict future ones. Such a network can be trained offline in batch mode with data collected from the analysis of the fabricated scattering surfaces. As for the choice of a training algorithm, the most straight forward procedure when working with the Matlab package is to simply use trial and error until satisfactory performance is achieved. Of course, in more complex environments such an approach can be avoided, however given the fact that the plant's transfer function is unknown it is not possible to make an proper investigation.

The "early stopping" (Demuth and Beale, 2002) method for improving generalization is then used. It reduces the chance of the network overfitting the data. The available data set is divided into three subsets. The first subset is the training set, which is used for computing the gradient and updating the network's weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error will decrease during the initial epochs alongside the training set error. When the network begins to overfit the data, the error on the validation set should begin to rise. When this happens the training stops and the weights and biases at the minimum of the validation error are returned. The test set error is not used during the training. If the error in the test set reaches a minimum at a significantly different iteration number than the validation set error, this may indicate poor division of the data set.

### 3.3 Research Results

The network was trained using the type 2 data and was tested on the type 1 data. The results are shown in table 3.1. The network can predict the standard deviation to a reasonable accuracy, however the correlation lengths are not well predicted. However, given that the network was trained with only 16 data points, it's performance is acceptable because no negative values are present and all the results are of the same magnitude.

The neural network consists of 4 input neurons, 7 hidden layer neurons, and 2 output neurons. The hidden neurons use the tan-sigmoid transfer function, whereas the output layer neurons use a linear transfer function. Their number was determined by trial and error. Using above 7 neurons resulted in much more frequent cases of overtraining, whereas a smaller number of neurons resulted in poor training with predictions frequently becoming negative or far out of range. The training algorithm for this case was chosen to be a training function that updates weight and bias values according to conjugate gradient backpropagation with Polak-Ribire updates.

Experiment		1	2	3	4	5
Test	$\tau_{test}$	9.2641	9.9587	10.7641	7.6112	7.8335
	$\sigma_{test}$	2.5809	2.6048	1.9280	2.0207	2.1765
Prediction	$ au_{sim}$	7.3302	7.5180	5.8673	6.4748	6.4377
	$\sigma_{sim}$	2.3272	1.7831	2.1750	4.0406	3.6911

Table 3.1: All values are multiplied by 100000 for clearer representation

The network's weights and biases are shown in table 3.2.

Biases	Hidden	Output
1	2.45932	-0.30151
2	-1.49661	0.75354
3	-0.89924	
4	0.11674	
5	0.67269	
6	1.40575	
7	2.24648	

Table 3.2: Neuron biases numbered in a top to bottom manner

And the hidden weights are show in table 3.3.

And the input layer weights are shown in table 3.4.

The regression analysis of the network shows that it is possible to fit a linear regression between the inputs and the outputs of the network and that the regression is good, however the lack of training data skews the possible accuracy.

In addition to this it can be seen in figure 3.4 that the network was trained in a very small number of epochs, however the validation error and the test error reach a minimum with a difference of 1 epoch, which is an indication that the division of the data set is good. Once again, this is due

Hidden Weights	Output 1	Output 2
1	0.62986	0.15210
2	0.22319	0.57093
3	-0.05585	0.66485
4	1.01412	-0.42893
5	0.010812	-0.96382
6	-0.33328	-0.22910
7	-0.30337	-0.49489

Table 3.3: Hidden Weights numbered in a top to bottom manner

Input Weights	Input 1	Input 2	Input 3	Input 4
1	-1.20634	0.92534	-0.28261	-1.38521
2	1.41377	-1.1470	0.77326	1.14442
3	1.53212	0.68046	-0.84186	-1.15178
4	1.12194	0.22545	0.04902	1.95192
5	0.75917	0.25323	1.55404	-1.43515
6	0.53262	1.01630	0.441112	2.01404
7	1.16680	-1.91867	0.15895	0.427061

Table 3.4: Input Weights numbered in a top to bottom manner



Figure 3.3: The regression analysis shows that the input-output relationship can be linearly fit, however there simply is not enough data

to the lack of enough training data.

As it can be seen from table 3.1, experiments 1 trough 3 are in agreement with the measured data. Clearly, given the small training set, one cannot expect to see an identical mapping, how-



Figure 3.4: The network's performance suggests that the data set has been properly divided which suggests that there should be little overfitting

ever no negative signs are present and the values are reasonably close. In addition it can be seen that estimating the standard deviation is much easier than estimating the correlation length of a given sample. One possible way to interpret this is to consider the way in which the standard deviation and the correlation length are calculated. From the work of Bergstrom (2008) it is seen that estimating the correlation length is a more complex mathematical procedure, whereas the standard deviation is more or less straightforward. A number of small investigations were made into this, in which surfaces with specific standard deviations and correlation lengths would be generated and their correlation lengths and standard deviations would be measured using the techniques of Bergstrom (2008). It was found that the error between the generated and predicted standard deviations was much smaller than that of the correlation lengths.

In addition to this it can be seen that experiments 4 and 5 have much larger estimation errors in comparison with the other experiments. This is to be expected, as these two experiments were performed using parameters that lie outside the maximum and minimum of those found in the training set. It is worth noting however, that multiple reevaluations of the network using the same number of hidden neurons and ratios, led to results which were occasionally much worse than those presented in this section. Thus it can be concluded that convergence is not always guaranteed when using such a small training set. Finally it is worth noting that the network will not be able to predict results which require parameter selection that lies outside of the established maximum and minimum parameters. This means that if the current fabricated samples are incapable of scattering light in the vicinity of 210 µm, then the ANN cannot be used to predict surfaces which will be capable of scattering in that range.

# Chapter 4 Light Band Scattering Simulation

This chapter is dedicated to the development of two light band scattering simulations based on the works of Bergstrom (2008); Tsang et al. (2001). The overall approach to computing the scattered field from a rough perfect electric conductor for a given band can be summarised as follows. First, a wave function which obeys the Helmholtz equation is created. Following this, an open form solution to this equation is found and the appropriate boundary conditions are chosen, in this case being the Dirichlet boundary conditions which represent the transverse electric case for the electromagnetic scattering problem. With the appropriate boundary condition chosen, the wave equation is converted into matrix form following the method of moments (MoM) technique. The scattered field is then calculated by using Huygen's princile and the numerical solution of the surface fields. Finally, the scattered field is expressed in terms of the bistatic scattering coefficient. Although changes have been made to the equations, so that they can represent a band instead of a single wavelength, they are minor in nature and do not alter the approach taken by Tsang et al. (2001). The second simulation relies on the small perturbation method (SPM) and has been presented without derivation. The same modification has been made to the SPM simulation to make it capable of simulating the scattering of entire bands of light.

## 4.1 Situational and Theoretical Analysis

The other major area of theoretical research which will be performed is of simulating and understanding how light scattering occurs. Several approaches have been briefly investigated, namely the works of (Oren and Nayar, 1996; Schroder et al., 2011; Ticconi et al., 2011; Harvey and Shack, 1978; Du and Liu, 2009). The problem of light scattering from a rough surface can be formulated in various ways starting from geometric ray tracing technique to stochastic models in combination with physical optics. The main difficulty in modeling this phenomena is that because of the complex geometry of the surface there are no analytic solutions to the governing equations . This means that numerical techniques must be employed extensively to approximate a solution (Hangartner, 2002). The task is further complicated by possible singularities which can occur during the evaluation of some of the functions as part of the numerical techniques. The avoidance of possible singularities is a field in applied mathematics which has also received substantial attention as numerical computation for simulations became more widely used (Knockaert, 1991; Nachamkin, 1990; Lee et al., 1980). Given the mathematical complexity of the field the author has been reluctant to cite equations directly out of context for the sake of clarity. Instead, references have been provided so that the curious reader can investigate the topics in full detail.

More complicated systems such as integrating spheres have been analyzed broadly as well (Crowther, 1996), or systems which use Gaussian beams. The case of Gaussian beam scattering (Gordon and Heyman, 2004) is a very interesting process for the development of waveguides, where the manipulation of beam patterns is vital to the performance of the optical system. Other works have focused on numerically investigating the curious process of backscattering in which light is reflected back towards the emitter at certain angles of incidence. The process is closely related to the specific surface roughness. It is a process which has numerous applications in fields such as astronomy. Understanding and controlling backscattering is crucial in waveguide design, as standing waves can degrade the quality of the signal (Soto-Crespo and Nieto-Vesperinas, 1988).

In (Oren and Nayar, 1996) an extensive reflectance model has been developed that relies on a geometric optics approximation which demonstrates that for certain rough surface geometries the Lambertian model is highly inaccurate. Despite this, the geometric optics approach is generally considered to be limited to certain geometries too, as it fails to take effects into consideration such as cross polarizations or the physical dimension of the wavelength. This is why the reflectance model does not account for wavelengths nor for the electromagnetic properties of the surface from which scattering occurs. The ray tracing approach is generally favored because of its simplicity as it avoids using higher level mathematics. A drawback of this approach is its limited domain of accurate applicability and the overall amount of the mathematics involved to take into consideration all the special cases such as shadowing and double angle reflections. The ray tracing techniques are compromise between mathematics and implementation - proper mathematical treatment of the theory at hand allows for more efficient and compact codes to be created, whereas simplifications to the theory at hand, such as the ray tracing, result in more complex codes from a programming standpoint (Bergstrom, 2008). In (Nordam et al., 2013) a two dimensional simulation is created based on the numerical solution of the Reduced Rayleigh Equation. Their work manages to include effects such as s- and p-polarization from a two dimensional metallic or dielectric rough random surface. They conclude that while within the validity of the Rayleigh hypothesis. This hypothesis has been criticized several times over the past century for lack of rigor and unrealistic assumptions (Wauer and Rother, 2009). Perhaps the most interesting optical theory is the attempt to create a matrix formulation for light scattering. This approach is known as Mueller calculus, after an MIT physics professor who developed the idea in 1943, which uses matrix algebra and calculus to manipulate the Stokes vectors. The theory deals with unpolarized and partially polarized light and this makes her suitable for tackling the phenomenon of light scattering (Letnes et al., 2012).

In (Schroder et al., 2011) the classical Rayleigh-Rice vector perturbation theory and the Generalized Harvey-Shack (GHS) theory are both assessed and compared. The Rayleigh-Rice vector perturbation theory relies on several far field approximations and limiting cases such as meeting the Rayleigh smooth surface criterion (Pinel et al., 2010) which help simplify the governing equations, whereas the Harvey-Shack theory tries to determine the transfer function of the surface when light is considered an input. The Generalized Harvey-Shack theory (Krywonos et al., 2006) is an improvement to the original theory of James Harvey (Harvey and Shack, 1978) which was developed during the late 1970s. The advantage of the GHS theory is that it allows the prediction of diffusely reflected light under any angle. Unfortunately the theory suffers in terms of computational inefficiency as it requires the evaluation of multiple, depending on the resolution one desires, Fourier transforms. Even with algorithms such as the Fast Fourier Transform (FFT), significant time can be lost. The key assumption made in the theory is that the rough surface would be perfectly Gaussian, which is one of the most general distributions encountered in roughness analysis. It has been shown that the theory has a wider domain of validity with respect to the Kirchoff approximation and the SPM. The main difference between the GHS theory and the MoM technique is that the latter provides a rigorous solution to the wave equations, whereas the former is only an approximate method (Choi and Harvey, 2013).

In (Ticconi et al., 2011) the reader can find an excellent summary of the various theories, both empirical and analytical, of light scattering. In this paper several of the widely used parameters, such as the Bidirectional Reflectance Distribution Function (BRDF), are derived and explained in detail. The advantages and disadvantages of all theories are discussed and summarized. In (Krywonos et al., 2006) the author provides another excellent discussion on the ranges of applicability of several semi-empirical light scattering theories and describes their domains of application. It is shown that due to the incredible complexity of the phenomenon of light scattering, empirical and semi-empirical approaches have very limited domains of application and that only rigorous solution of the wave equations are capable of describing the process in more detail. Finally, in (Du and Liu, 2009) the authors present a very innovative analysis of the process at hand. The authors have developed the scattering theory for various random surfaces by means of a stochastic functional approach combined with a group-theoretic consideration, which was originally introduced by one of the author in the theory of propagation in random media.

The method of moments is a very flexible computational method. It can be used to solve differential equations, integral equations, and integro-differential equations. One of the main advantages of this technique lies in its variational nature of the solution, which implies that even if the unknown function is modeled to first order accuracy, the solution is accurate to the second order. The technique involves a reformulation of Maxwell's equations because it makes use of the Green's function and Helmholtz's equation in the case of light scattering. This allows for open region problems such as radiation and scattering to be solved in an efficient manner. Unlike in other computational methods, the device domain is not discretized, and only the unknown function is discretized in the MoM approach. As a result, this method does not suffer from numerical dispersion and the matrix sizes are smaller. The main limitation in utilizing this technique is that the resulting matrix equations are notoriously singular. For one- and two-dimensional problems the singularities can be avoided, however higher dimensions are noticeably more difficult to efficiently avoid. (Garg, 2008). In summary, the MoM can be explained as follows. A general linear, inhomogenous equation may be described in operator form as

$$L(u) = b \tag{4.1}$$

where L is a linear operator and represents the system. The function b is known and represents the excitation of the system and the function u represents the system's response to a given input. The unknown function u is expanded as a series of known functions which have unknown amplitudes. The amplitudes are determined by a set of test functions w.

The first step in the process is to choose a set of linearly independent basis functions within the domain of L, and express the unknown function u in the form of a series as shown

$$u = \sum_{n} c_n u_n \tag{4.2}$$

where the constants  $c_n$  are unknown and are to be determined. The numerical performance of the simulation is dependent on the choice of proper basis functions and a poor choice can lead to a divergent solution and a waste of everybody's time. For an exact solution, the summation should be converted to an integral, which is computationally impossible. In addition to this it is assumed that a proper inner product has been defined for the problem. After this the set of already chosen test functions and the inner product between the test function and the discretized operator and the excitation functions are computed. This results in the following equation

$$\sum_{m} \sum_{n} c_n \langle w_m, L(u_n) \rangle = \sum_{m} \langle w_m, b \rangle$$
(4.3)

The above equation can be written in matrix form as

$$\mathbf{L}\mathbf{c} = \mathbf{b} \tag{4.4}$$

Should the operator matrix  $\mathbf{L}$  be well behaved and have low conditionality, the solution can be readily obtained. In most practical application this matrix is notoriously singular.

There are two main options for the choice of test functions - Galerkin's method and point matching. Both approaches compliment each other and depend on the problem at hand. Choosing Dirac delta functions as basis functions avoids the necessity to compute the inner product, which significantly reduces computational time and coding complexity. Of course, one has to make sure that the operator equation is not too badly violated that the solution becomes unreliable. The accuracy of the solution improves as the number of match points is increased. Galerkin's method makes the basis functions also the test functions. This simplifies the task of choosing proper test functions, which makes this a popular method (Garg, 2008).

# 4.2 Conceptual Model

#### 4.2.1 Method of Moments

We begin by modeling the tapered incident wave and combining it with a Planck distribution in order to simulate the power distribution of a light band emitted by a black body. In reality the calibration source of SAFARI is not a black body, however it can be approximated as one. The necessity to taper an incident wave is so that the surface current is driven to zero at the edges of the simulated surface. Should there be an abrupt change in the surface currents, artificial reflection will occur. In their book on numerical simulations, Tsang et al. (2001) have chosen to model a Gaussian tapering window. This is convenient because the pinhole pupil of the black body radiator in figure 1.4 generates the same light pattern. The function describing such behavior has been slightly modified to account for the wavelength dependent normalized power  $E_{\lambda}$  which follows a normalized, discretized Planck distribution. Since electric fields satisfy the superposition principle, the simulation's results can be superimposed for a band of different wavelengths, obeying the normalized Planck distribution

$$E_{\lambda} = 2 h c^2 \lambda^{-5} \left( e^{\frac{h c}{\lambda k_B T}} - 1 \right)^{-1}$$

$$\tag{4.5}$$

Where c is the speed of light in vacuum, h is Plank's constant, and  $k_B$  is Boltzmann's constant,  $\lambda$  is the wavelength of interest, and T is the absolute temperature. The consequent wave representation is given as

$$\Psi_{inc}\left(\bar{r}, E_{\lambda}\right) = E_{\lambda} \mathrm{e}^{jk\left(x\sin(\theta_{inc}) - z\cos(\theta_{inc})(1 + w(\bar{r})) - \frac{\left(x + z\tan(\theta_{inc})\right)^2}{g^2}\right)}$$
(4.6)

where g is the tapering parameter. The additional factor in the phase,  $w(\bar{r})$  is inserted so that  $\Psi_{inc}(\bar{r}, E_{\lambda})$  obeys the wave equation to to a second derivative.

$$w(\bar{r}) = \left(2\frac{\left(x + z\tan\left(\theta_{inc}\right)\right)^2}{g^2} - 1\right)k^{-2}g^{-2}\left(\cos\left(\theta_{inc}\right)\right)^{-2}$$
(4.7)

The next step is to create a surface model which is representative of the surfaces produced by sandblasting. Originally the surface generating technique from Bergstrom (2008) was used, however detailed analysis of processed samples showed that there still is room for improvement, albeit with negligible gain. The results from these small improvements will be discussed in Chapter 5.

A one-dimensional random rough surface can be represented as  $z = \zeta(x)$ , is described by a height probability distribution and an autocovariance function (ACF). The height probability distribution describes the height variations from a given reference, generally taken to be 0, whereas the ACF describes the variance of the heights laterally. This way it is possible to create surfaces which appear very smooth, yet still follow a certain distribution. The most common height distribution is chosen to be Gaussian, however depending on the process at hand different distributions can be used. For the purpose of this study it was assumed that the rough surface behaves like a Gaussian. This also makes the analysis of a surface easier, as one can only work with estimates of the correlation length and the standard deviation. The ACF function can follow any distribution, but the most commonly occurring ones in surface analysis have been found to be Gaussian and exponential. Bergstrom (2008) has chosen to use a Gaussian ACF

$$C(\tau) = \langle \zeta(x_1) \zeta(x_2) \rangle = \sigma^2 \mathrm{e}^{-\frac{(|x_1 - x_2|)^2}{\tau}}$$
(4.8)

Where  $\sigma$  is the height standard deviation and  $\tau$  is the correlation length and  $x_1$  and  $x_2$  are two consecutive surface points. Should the height distribution function and the ACF both be Gaussian, it has been shown in the work of Bergstrom (2008) that the root mean square (RMS) slope is  $\sqrt{2\sigma}/\tau$ .

The developed algorithm works in the following manner. We create a vector, or sequence, of random numbers  $\mathbf{Z} = [x_1, x_2, \dots, x_n]^T \sim \sigma N(0, 1)$ . The discrete Fourier transform of  $\mathbf{Z}$  is flat over the entire frequency range and if  $\mathbf{Z}$  were continuous the frequency range would span from negative infinity to positive infinity. The amplitudes of each frequency are normally distributed around the mean, 0 in this case, however every frequency is equally likely to occur. Thus,  $\mathbf{Z}$  is completely uncorrelated and every element is independent of the previous one. This can be changed by introducing a windowing function, which in this case is the ACF, and convolute it with  $\mathbf{Z}$ . The Fourier transform simplifies this task, following the convolution theorem, by transforming the process into a multiplicative one. The result is a band limited normal distribution, where some frequencies are more likely to occur than others, which introduces the desired correlation. Once the product has been computed, the inverse Fourier transform is taken, which converts

the spatial frequency into temporal frequency. Finally, a scaling constant is given such that the statistical representation remains unchanged. Such constants can be evaluated analytically depending on the type of probability distribution and windowing function. While this method produces surfaces that agree well with measurements, time series analysis has shown the presence of a truncated Laplacian distribution, which has been interpreted as micro cracks in the metal. Thus, the modified surface model is shown as

$$f(x) = \sqrt{\frac{2L}{N\tau}} \mathscr{F}^{-1}\left(\mathscr{F}\left(C(\tau, x)\right)\mathscr{F}\left(Z(x)\right)\right) + \epsilon_x$$
(4.9)

where  $\epsilon_x \sim Laplace(\mu, b)$ , where  $\mu$  is the mean and b is commonly referred to the diversity and for the case of a centered Laplace distribution is given simply by

$$b = \frac{\sum_{x=1}^{N} |x|}{N}$$
(4.10)

and the expected value is equal to 0 for all

$$E\left(\epsilon_{x}|e_{1},e_{2},\ldots e_{N}\right)=0$$

It is worth noting that the modified surface generator does not truncate the Laplace distribution for simplicity.

This numerical method is based on the formulation of integral equations for the Dirichlet problem and converting them into matrix equations using the MoM technique. Consider an incident wave  $\Psi_{inc}(\bar{r}, E_{\lambda})$  impinging upon a random rough surface with a height profile z = f(x). In two-dimensional scattering problems  $\bar{r} = x\hat{x} + z\hat{z}$  and the wavefunction  $\Psi(\bar{r}, E_{\lambda})$  is given by:

$$\Psi(\bar{r}, E_{\lambda}) = \Psi_{inc}(\bar{r}, E_{\lambda}) + \Psi_s(\bar{r}, E_{\lambda})$$
(4.11)

where  $\Psi_s(\bar{r}, E_\lambda)$  is the scattered wave distribution, where  $\bar{r}$  denotes the field coordinates and  $\bar{r}'$  denotes the source coordinates. The wavefunction obeys the Helmholtz equation:

$$(\nabla^2 + k^2)(\Psi) = 0 \tag{4.12}$$

The two-dimensional Green's function obeys the equation

$$\left(\nabla^2 + k^2\right)g\left(\overline{r},\overline{r}'\right) = -\delta\left(\overline{r} - \overline{r}'\right) \tag{4.13}$$

and

$$g(\bar{r},\bar{r}') = \frac{j}{4} H_0^{(1)} \left( k \left| \bar{r} - \bar{r}' \right| \right)$$
(4.14)

Let the spaces above and bellow the rough surface be denoted by region 0  $(V_0)$  and region 1  $(V_1)$ . By applying the Green's theorem the following equation is obtained

$$\iint_{V_{0}} \Psi\left(\overline{r}, E_{\lambda}\right) \nabla^{2} g\left(\overline{r}, \overline{r}'\right) - g\left(\overline{r}, \overline{r}'\right) \nabla^{2} \Psi\left(\overline{r}, E_{\lambda}\right) d\overline{r} = 
- \int_{S} \hat{n} \cdot \left(\Psi\left(\overline{r}, E_{\lambda}\right) \nabla g\left(\overline{r}, \overline{r}'\right) - g\left(\overline{r}, \overline{r}'\right) \nabla \Psi\left(\overline{r}, E_{\lambda}\right)\right) ds 
+ \int_{S_{\infty}} \hat{n} \cdot \left(\Psi\left(\overline{r}, E_{\lambda}\right) \nabla g\left(\overline{r}, \overline{r}'\right) - g\left(\overline{r}, \overline{r}'\right) \nabla \Psi\left(\overline{r}, E_{\lambda}\right)\right) ds \quad (4.15)$$

where  $S_{\infty}$  is the surface at infinity. Using 4.12 and 4.13 in the left hand side of 4.15, we have

$$\iint_{V_0} \Psi\left(\overline{r}, E_{\lambda}\right) \left(-k^2 g\left(\overline{r}, \overline{r}'\right) - \delta\left(\overline{r} - \overline{r}'\right)\right) + g\left(\overline{r}, \overline{r}'\right) k^2 \Psi\left(\overline{r}\right) d\overline{r} = -\iint_{V_0} \delta\left(\overline{r} - \overline{r}'\right) \Psi\left(\overline{r}, E_{\lambda}\right) d\overline{t} 4.16$$

To evaluate 4.16, it is necessary to define where  $\overline{r}'$  is. It is possible for  $\overline{r}'$  to be above or bellow the surface. It can also be infinitesimally close to the boundary. Depending on which region  $\overline{r}'$ is infinitesimally close to, it can be defined as  $\overline{r}'_+$  for region 0 and  $\overline{r}'_-$  for region 1. Also

$$-\iint_{V_0} \delta\left(\overline{r} - \overline{r}'\right) \Psi\left(\overline{r}, E_{\lambda}\right) d\overline{r} = \begin{cases} -\Psi\left(\overline{r}', E_{\lambda}\right) & \text{if } \overline{r}' \in V_0 \\ 0 & \text{if } \overline{r}' \in V_1 \end{cases}$$
(4.17)

The surface integral at infinity in 4.15 provides the solution for the incident wave. Thus

$$\Psi_{inc}\left(\overline{r}', E_{\lambda}\right) + \int_{S} \hat{n} \cdot \left(\Psi\left(\overline{r}, E_{\lambda}\right) \nabla g\left(\overline{r}, \overline{r}'\right) - g\left(\overline{r}, \overline{r}'\right) \nabla \Psi\left(\overline{r}, E_{\lambda}\right)\right) ds = \begin{cases} \Psi\left(\overline{r}', E_{\lambda}\right) & \text{if } \overline{r}' \in V_{0} \\ 0 & \text{if } \overline{r}' \in V_{1} \end{cases} (4.18)$$

The 0 in 4.18 corresponds to the extinction theorem. The resulting expression is interesting because  $\overline{r}$  is on the surface S, while  $\overline{r'}$  can be either in region  $V_0$  or  $V_1$ . There are two ways in which 4.18 can be solved, both requiring different assumptions and leading to different interpretations. The first solution arises when the Dirichlet boundary condition is enforced, whereas the second solution is the result of enforcing the Neumann boundary condition. The first solution is interpreted as the transverse electric (TE) case of electromagnetic scattering, whereas the second solution is interpreted as the transverse magnetic (TM) case of electromagnetic scattering. Since the SAFARI instrument, as well as the bolometer used in the experiments are optical detectors, the TE solution was used as the governing principle in the band scattering simulation. This also makes the implementation of the theory much simpler.

The Dirichlet boundary condition is

$$\Psi\left(\bar{r}, E_{\lambda}\right) = 0 \tag{4.19}$$

for  $\overline{r}$  on S. Then

$$\Psi_{inc}\left(\overline{r}', E_{\lambda}\right) - \int_{S} g\left(\overline{r}, \overline{r}'\right) \hat{n} \cdot \nabla \Psi\left(\overline{r}, E_{\lambda}\right) ds = \begin{cases} \Psi\left(\overline{r}', E_{\lambda}\right) & \text{if } \overline{r}' \in V_{0} \\ 0 & \text{if } \overline{r}' \in V_{1} \end{cases}$$
(4.20)

If we let  $\overline{r}'$  approach the surface, both cases in 4.20 will approach zero, which simplifies the problem further. For  $\overline{r}'$  and  $\overline{r}'$  on S, equation 4.20 becomes a surface integral equation for the surface unknown  $\hat{n} \cdot \nabla \Psi(\overline{r})$ 

$$\Psi_{inc}\left(\overline{r}', E_{\lambda}\right) = \int_{S} g\left(\overline{r}, \overline{r}'\right) \hat{n} \cdot \nabla \Psi\left(\overline{r}, E_{\lambda}\right) ds \tag{4.21}$$

In terms of applicability, equation 4.21 cannot be discretized just yet because as  $\overline{r}'$  approaches  $\overline{r}$ ,  $g(\overline{r},\overline{r}')$  becomes singular. This singularity is much easier to circumnavigate than the singularity resulting in the three-dimensional treatment of the same problem. This equation is referred to as the electric field integral equation (EFIE) for the TE case because the electric field is oscillating perpendicular to the x and z axis, which allows for its representation as a scalar wave.

Before the integral equation can be converted to a matrix one, a few final steps must be taken. Firstly, we must rewrite equation 4.21 into a more practical form. We do this by understanding that ds is a measure of the infinitesimal length that is  $\overline{r}$  and that z = f(x). That is to say, the surface profile is comprised of an infinite number of vectors  $\overline{r}$ , at least in the continuous sense. So in order to determine ds we need to determine the magnitude of the rate of change of  $\overline{r}$ . In other words

$$\frac{ds}{dx} = \left\| \frac{d\overline{r}}{dx} \right\| \tag{4.22}$$

Upon taking the derivative of  $\bar{r}$  and applying the Euclidean norm onto the derivative, we end up with a less general expression. Implementing the surface derivative into a simulation, requires that both "edges" of the derivative to be connected, so that surface discontinuities can be avoided as they cause false reflections on the sides (Garg, 2008) .Rewriting the integral equation and taking into consideration the fact that we can substitute the generalized vectors with more concrete arguments

$$\Psi_{inc}\left(x', f\left(x'\right), E_{\lambda}\right) = \int_{-\frac{L}{2}}^{\frac{L}{2}} \sqrt{1 + \left(\frac{df(x')}{dx}\right)^2} g\left(x, f(x); x', f(x')\right) \left(\hat{n} \cdot \nabla \Psi\left(\overline{r}, E_{\lambda}\right)\right)_{z=f(x)} dx \quad (4.23)$$

where S has been limited to between  $-\frac{L}{2}$  and  $\frac{L}{2}$ . The component  $\sqrt{1 + (\frac{df(x')}{dx})^2} (\hat{n} \cdot \nabla \Psi(\bar{r}, E_{\lambda}))_{z=f(x)}$ is treated as the surface unknown and it can be represented as  $u(x, E_{\lambda})$ , and  $\Psi_{inc}(x', f(x'), E_{\lambda})$ can be expressed as  $b(x', E_{\lambda})$  as it is a function of x' and  $E_{\lambda}$  only, however since the wave equations are independent of intensity within the range of interest, the  $E_{\lambda}$  is treated as a constant and effectively ignored in the derivations. The kernel of the integral equation is

$$K(x',x) = g(x, f(x); x', f(x'))$$
(4.24)

Combining all of this into a more compact form yields

$$\int_{-\frac{L}{2}}^{\frac{L}{2}} K(x',x) u(x) dx = b(x',E_{\lambda})$$
(4.25)

We now can begin converting equation 4.25 into a matrix one using the technique of MoM. The domain is -L/2 < x < L/2 is divided into N intervals, each of width  $\Delta = L/N$ . The intervals are centered at  $x_m, m = 1, 2, ..., N$ . Thus  $u(x, E_{\lambda}) = u_N$  in the Nth interval and we point match the integral equation at  $x' = x_m$ . Thus equation 4.25 becomes

$$\int_{-\frac{L}{2}}^{\frac{L}{2}} K(x_m, x) \, u(x, E_{\lambda}) dx = b(x_m, E_{\lambda}) \tag{4.26}$$

Finally, equation 4.26 can be replaced by a summation, assuming that  $u(x, E_{\lambda})$  is constant in each interval which means that pulse basis functions are employed.

$$\Delta x \sum_{\substack{n=1\\n \neq m}}^{N} K(x_m, x_n) u(x_{n, E_{\lambda}}) + \left( \int_m K(x_m, x) \, dx \right) u(x_{m, E_{\lambda}}) = b(x_{m, E_{\lambda}}) \tag{4.27}$$

The integral over the *m*th interval represents the diagonal of the newly created matrix equation. This is where the singularity, also known as self patch term or contribution, is contained. Luckily there is a way to outsmart the singularity with relatively little effort, in this case at least. The argument of the Green's function is small when x approaches  $x_m$ , which means that a series expansion is possible. In this case Tsang et al. (2001) have used a first order series expansion.

$$H_0^{(1)}(\omega) = j\frac{2}{\pi}\ln\left(\frac{\gamma\omega}{2}\right) \tag{4.28}$$

where  $\omega = 1.78107241$ . Further approximations can be made by taking the Taylor series of f(x) around the point  $x_m$ , which results in the following

$$f(x) \simeq f(x_m) + f'(x_m)(x - x_m)$$
 (4.29)

Remembering that  $\omega$  is equivalent to  $k |\bar{r} - \bar{r}'|$ , we can substitute the Taylor approximation into the vectors and evaluate the result. The result then becomes  $k |(x - x_m)^2 + (f'(x_m)(x - x_m)^2|$ . This allows for the following, somewhat confusing yet working, approximation to be made

$$\int_{m} K(x_{m}, x) dx = 2 \int_{x_{m}}^{x_{m} + \frac{\Delta x}{2}} K(x_{m}, x) dx$$

$$\simeq \frac{j}{2} \int_{0}^{\frac{\Delta x}{2}} 1 + j\frac{2}{\pi} \ln\left(\frac{\gamma}{2}kx\sqrt{1 + (f'(x))^{2}}\right) dx$$

$$= \frac{j\Delta x}{4} \left(1 + j\frac{2}{\pi} \ln\left(\frac{\gamma k}{4}\Delta x\sqrt{1 + (f'(x))^{2}}\right) - 1\right)$$
(4.30)

Thus the discretized kernel can be expressed as

$$A_{mn} = \begin{cases} \Delta x K(x_m, x) & \text{for } n \neq m\\ \frac{j\Delta x}{4} \left( 1 + j\frac{2}{\pi} \ln\left(\frac{\gamma k}{4} \Delta x \sqrt{1 + (f'(x))^2}\right) - 1 \right) & \text{for } n = m \end{cases}$$
(4.31)

Once the singularity has been taken care of, it is finally possible to construct the matrix equation as well as to account for the varying power intensities

$$\sum_{\lambda_{min}}^{\lambda_{max}} \mathbf{A} \mathbf{u} = \sum_{\lambda_{min}}^{\lambda_{max}} \mathbf{b}$$
(4.32)

Or in equation form

$$\sum_{\lambda_{min}}^{\lambda_{max}} \sum_{m=1}^{N} \sum_{n=1}^{N} A_{mn} u_{n\lambda} = \sum_{\lambda_{min}}^{\lambda_{max}} b_{m\lambda}$$
(4.33)

Higher order series expansions were investigated, going up to a fourth order approximation using Maple 18, however the gain in accuracy was immeasurable due to the Monte Carlo nature of the simulation (Sadiku, 2009) and thus the added numerical complexity was deemed unnecessary.

Now that the evaluation of  $u_{n\lambda}$  has been established it is possible to proceed with the calculation of the scattered field by using Huygen's principle. Given that  $\Psi(\bar{r}, E_{\lambda})$  and  $\hat{n} \cdot \nabla \Psi(\bar{r}, E_{\lambda})$  are known, it is possible to evaluate the scattered field  $\Psi_s(\bar{r}', E_{\lambda})$  by carrying out the integration of equation 4.18. The calculation of the bistatic scattering coefficients is made possible by setting  $\bar{r}'$  in the far field which yields the following approximation

$$g(\overline{r},\overline{r}') = \frac{j}{4}\sqrt{\frac{2}{\pi k\overline{r}'}} e^{-1/4 j\pi} e^{jk\overline{r}'} e^{-jk(x\sin(\theta) + z\cos(\theta))}$$
(4.34)

From which it follows that

$$(\hat{n} \cdot \nabla g\left(\bar{r}, \bar{r}'\right))_{z=f(x)} \sqrt{1 + \left(\frac{d}{dx} f\left(x\right)\right)^2} = \frac{j}{4} \sqrt{\frac{2}{\pi k \bar{r}'}} e^{-j\frac{\pi}{4}} e^{jk\bar{r}'} \cdot \left(\frac{d}{dx} f\left(x\right) \left(jk\sin(\theta_{sca}) - jk\cos(\theta_{sca})\right) e^{-jk(x\sin(\theta) + z\cos(\theta))} \right)$$
(4.35)

Inserting equation 4.35 and equation 4.34 into equation 4.18, one obtains

$$\Psi_{sca}\left(\overline{r}, E_{\lambda}\right) = \frac{j}{4} \sqrt{\frac{2}{\pi k \overline{r}}} e^{-1/4 j \pi} e^{j k \overline{r}} \Psi_{sca}^{(N)}\left(\theta_{sca}, E_{\lambda}\right)$$
(4.36)

Where

$$\Psi_{sca}^{(N)}\left(\theta_{sca}, E_{\lambda}\right) = -\int_{-\infty}^{\infty} -u\left(x, E_{\lambda}\right) + \Psi\left(x, E_{\lambda}\right) jk\left(\left(\frac{d}{dx}f\left(x\right)\right)\sin\left(\theta_{sca}\right) - \cos\left(\theta_{sca}\right)\right)\right) \cdot e^{-jk\left(x\sin\left(\theta_{sca}\right) + f\left(x\right)\cos\left(\theta_{sca}\right)\right)} dx$$
(4.37)

In which  $\Psi(x, E_{\lambda})$  is the surface field and contains both x and z = f(x). From a programming perspective this simply means that we keep track of the index and the corresponding value at

that index. And  $u(x, E_{\lambda})$  is proportional to the surface normal derivative, as it has been established earlier. From a programming point of view it is sufficient to know the value of  $u(x, E_{\lambda})$ . In order to calculate the bistatic scattering coefficient, Poynting's vector in the direction of the scattered field must be determined. Poynting's vector is given as

$$\overline{S}_{sca}\left(\overline{r}', E_{\lambda}\right) = -\frac{1}{2\eta k} Im\left(\Psi_{sca}\left(\overline{r}', E_{\lambda}\right) \nabla \Psi_{sca}^{*}\left(\overline{r}', E_{\lambda}\right)\right)$$
(4.38)

When the far field approximation is applied to equation 4.38 it becomes

$$\overline{S}_{sca}\left(\overline{r}', E_{\lambda}\right) = -\frac{\hat{r}'}{16} \frac{|\Psi\left(\overline{r}, E_{\lambda}\right)|^{2}}{\eta \pi k \overline{r}}$$

$$\tag{4.39}$$

where  $\hat{r}'$  represents the direction of the field. From the far field approximation of Poynting's vector it is possible to calculate the total scattered power  $P_{sca}$ , which is the integral of the product of equation 4.39 with  $\bar{r}'$  over the range from  $-\frac{\pi}{2}$  to  $\frac{\pi}{2}$  with respect to the scattered angle  $\theta_{sca}$ . The bistatic scattering coefficient  $\sigma(\theta_{sca}, E_{\lambda})$  is defined as

$$\frac{P_{sca}}{P_{inc}} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \sigma\left(\theta_{sca}, E_{\lambda}\right) d\theta_{sca} \tag{4.40}$$

By taking the derivative on both sides of equation 4.40 and remembering that  $P_{sca}$  is also integrated with respect to  $\theta_{sca}$ , an expression for  $\sigma(\theta_{sca}, E_{\lambda})$  can be directly obtained. The only left unknown is then  $P_{inc}$ . Luckily the definition of  $\sigma(\theta_{sca}, E_{\lambda})$  is that its integral over the range from  $-\frac{\pi}{2}$  to  $\frac{\pi}{2}$  is equal to unity for non-penetrable surfaces, which also includes rough surfaces. This means that a suitable "normalization" factor can be chosen to represent  $P_{inc}$ . In their work, Tsang et al. (2001) have chosen the following final expression to represent  $\sigma(\theta_{sca}, E_{\lambda})$  in the spatial domain

$$\sigma\left(\theta_{sca}, E_{\lambda}\right) = \frac{1}{8} \left(\left|\Psi\left(\overline{r}, E_{\lambda}\right)\right|\right)^{2} \sqrt{2\pi^{-3/2} k^{-1} g^{-1}} \left(\cos\left(\theta\right)\right)^{-1} \left(1 - \frac{1 + 2 \left(\tan\left(\theta\right)\right)^{2}}{2k^{2} g^{2} \left(\cos\left(\theta\right)\right)^{2}}\right)^{-1} \quad (4.41)$$

The simulation is then re-evaluated multiple times, along with new surfaces being generated each time, for every given  $E_{\lambda}$  and the results are averaged out using iterative averaging (Sadiku, 2009). Finally,  $\sigma(\theta_{sca}, E_{\lambda})$  is summed for all  $E_{\lambda}$  and this gives results in the evaluation of a scattered band of light.

#### 4.2.2 Small Perturbation Method

A second simulation using the same light band approach was created. The method employed was the Small Perturbation Method (SPM) which was originally developed by (Rice, 1951) for predicting scattered fields from slightly rough surfaces. This simulation was created before the processed samples were measured and as such was unknown whether the surface would be considered slightly rough or very rough as compared to the wavelength. The SPM is based on the Rayleigh hypothesis, which expresses the reflected and transmitted waves as going upwards

and downwards. The field amplitudes are then determined from the boundary conditions. The SPM is much faster to compute, requiring no Monte Carlo techniques whatsoever, however it has a more limited domain of applicability than the MoM. The bistatic scattering coefficient for the incoherent wave for the SPM is

$$\sigma\left(\theta_{sca}, E_{\lambda}\right) = 2 \frac{k^3 \left(\cos\left(\theta_{sca}\right)\right)^2 \cos\left(\theta_{inc}\right) \sigma^2 \tau e^{\frac{1}{4} \left(\sin\left(\theta_{sca}\right) - \sin\left(\theta_{inc}\right)\right)^2 k^2 \tau^2}}{\sqrt{\pi}} \tag{4.42}$$

where  $\sigma$  and  $\tau$  were found to be the surface standard deviation and correlation length. As it was not mentioned in Tsang et al. (2001), it was by experimental work that the simulation's output was compared to their original results. At this point it is worth mentioning that the website that has been provided by the authors has changed domain.

## 4.3 Research Design

Apart from the BioRad measurements, a new experimental setup was designed and assembled. A drawing of the setup is shown bellow, excluding the amplifiers and the data acquisition devices



Figure 4.1: Light scattering measurement setup alone

The encasing and chopper mechanism have been custom made at SRON. The infrared source is a 22 Watt ceramic element, model number 6575 from Instruments (2014). At 21 W this element emits at a temperature of approximately 850 K. In order to create a tapered beam which would be perpendicular to the surface, a plano-convex lens was used which had a focal distance of 3 cm. The sample is placed at an angle of  $\pi/2$  radians. The sample, lens, chopper, and 6575 encasing are mounted to an aluminum plate, which is attached to a stepper motor rotation mechanism. The sample is centered on the axis of rotation as to avoid creating parallaxing while rotating the plate. The surface of the rail was covered with sandpaper in order to reduce stray reflections from the infrared source. The distance between the sample and the 6575 encasing was measured to be 7.3 cm. The detector utilized in the experiments was a General Purpose 4.2K Bolometer



System from Infrared Laboratories, Laboratories (2013).

Figure 4.2: Infrared source incasing to the left, lens in center, and sample mounted onto rotating plate to the right

The bolometer has been mounted in a side-looking configuration with a Winston cone collector, a three position filter sector wheel, far-infrared cut-on type filters and an outer vacuum window. The bolometer is of the composite type and features a small silicon element thermally bonded to a suitably blackened 2.5 mm diamond absorber mounted in a cylindrical cavity. The absorbing layer thickness has been selected to minimize fringing effects. The mounting block in turn is bolted to an "L" type bracket which supports the cone exit aperture. The Winston cone features an entrance aperture of 12.7 mm at a focal ratio of 3.8 and an exit aperture of 1.6 mm and has been gold plated to prevent tarnish and to improve thermal properties. An 800 cm<sup>-1</sup> far infrared cut-on type filter consisting of 0.5 mil white polyethylene stretched tightly over holder rings and overlaid on one face with  $4-8 \,\mu\text{m}$  diamon scatter layer.

The aperture of the bolometer was limited to approximately 1 mm in order to avoid saturation. In addition to this, a combination of a low-pass and high-pass optical filters was used, which created a band ranging from  $30 \,\mu\text{m}$  to  $65 \,\mu\text{m}$ . Unfortunately the transmission of air within this band could not be measured. Experimental data and simulations suggest that the transmittance of air should be low and uniform within the measured band, however no further inferences could be made.

The chopper mechanism and the bolometer were connected to an SR-830 lock-in amplifier, which in turn was connected to a desktop PC. The control program was written (drawn) in LabView by Darren Hayton, SRON. The control program would change the angle between the sample and the bolometer and then readout the length of the amplitude vector from the SR-830 lock-in amplifier.



Figure 4.3: Optical filters elegantly mounted onto the bolometer's window. The silicon hose is used to recapture evaporating helium

As it can be seen from figure 4.5 the setup was not firmly connected. Due to the custom nature of all the components utilized and the lack of an available optical bench at the time, the position of the components had to be measured using a micrometer and a straight angle. This has definitely led to certain angle deviations, which could not be determined. However, it is presumed that the errors are small, in the range of a degree or two, because that was the smallest angular deviation that was possible to measure.

## 4.4 Research Results

For the simulation part, 10 wavelengths were chosen, starting from 30 µm to 65 µm. To each of them a normalized intensity value was assigned ranging from 1 to 0. The descriptive statistics obtained from the surface analysis tool are first normalized by the wavelength, as done in (Tsang et al., 2001) in order to reduce the number of necessary surface points. The wavelength gets normalized to itself, after which all surface control parameters are divided by a factor of 10. This allows to further increase the efficiency of the simulation because as long as the ratios between the wavelength, correlation factor, standard deviation, and Laplace diversity factor are the same, everything else can be scaled.

The generated surface profile is a fixed multiple of the normalized wavelength. The surface parameters are scaled accordingly to the wavelength and this is how the profile can change it's statistics while having a fixed resolution.



Figure 4.4: Desktop PC connected to the SR-830 lock-in amplifier. Rotating setup and bolometer are to the right



Figure 4.5: Overview of the entire experimental setup. Emphasis on the lack of connection between some of the components



Figure 4.6: Normalized discretized Planck distribution



Figure 4.7: A simulated surface having  $\sigma = 21.8 \times 10^{-5}$  m,  $\tau = 7.83 \times 10^{-5}$  m,  $b = 2.26 \times 10^{-7}$  m,  $\lambda = 65 \,\mu\text{m}$ 

The tapered wave is then projected onto the surface at an angle of  $\pi/2$ .

When the matrix equations have been solved, the final result is the scattered band as shown in figure 4.9.



Figure 4.8: The incident field distribution shows where the majority of power is concentrated along the surface. Tapering parameter g = L/7



Figure 4.9: Scattered band of 2D spherical coordinates (represented in Cartesian space) with  $\sigma = 21.8 \times 10^{-5}$  m,  $\tau = 7.83 \times 10^{-5}$  m,  $b = 2.26 \times 10^{-7}$  m,  $\lambda = 30 - 65 \,\mu\text{m}$ 

It is important to note the conditionality of the operator matrix. This is a direct measure of how erroneous the solution is. The lower the conditionality, the better.

The simulation results are compared with the measured light scatter patterns of the type 1 samples and are superimposed one over the other. A constant offset of  $4^{\circ}$  was observed in all measurements. Based on this it was concluded that the error was systematic and it has been compensated for in all figures.



Figure 4.10: Conditionality of operator matrix. It can be seen that as the simulation proceeds, the conditionality converges



Figure 4.11: Beam pattern measured from an unprocessed sample

The beam pattern generated by the 9575 element is by no means close to what has been simulated. This is a definite experimental design flaw and a more precise optical setup must be assembled if experiments with higher accuracy should be performed. The specular case was not tested simply because such a pattern could not be generated easily and given that the case is specular, the beam patterns will be the same.



Figure 4.12: Type 1, experiment 1:  $\theta_{inc} = 45^{\circ}$ ,  $\tau = 9.26 \times 10^{-5}$  m,  $\sigma = 2.58 \times 10^{-5}$  m,  $b = 2.02 \times 10^{-7}$  m. Measured data has been normalized. Square data points represent the simulation. Angle error of  $4^{\circ}$ 



Figure 4.13: Type 1, experiment 2:  $\theta_{inc} = 45^{\circ}$ ,  $\tau = 9.96 \times 10^{-5}$ ,  $\sigma = 2.6 \times 10^{-5}$  m,  $b = 2.031 \times 10^{-7}$  m. Measured data has been normalized. Square data points represent the simulation. Angle error of  $4^{\circ}$ 

The measured fields are in agreement with the simulation, except in figure 4.15. Under closer inspection it was found that the sample had many smooth "islands", areas which were not roughened. This is to be expected, as experiment 4 was processed for only 20 seconds. What is



Figure 4.14: Type 1, experiment 3:  $\theta_{inc} = 45^{\circ}$ ,  $\tau = 1.08 \times 10^{-4}$  m,  $\sigma = 1.93 \times 10^{-5}$  m,  $b = 2.05 \times 10^{-7}$  m. Measured data has been normalized. Square data points represent the simulation. Angle error of  $4^{\circ}$ 



Figure 4.15: Type 1, experiment 4:  $\theta_{inc} = 45^{\circ}$ ,  $\tau = 7.61 \times 10^{-5}$  m,  $\sigma = 2.02 \times 10^{-5}$  m,  $b = 2.09 \times 10^{-7}$  m. Measured data has been normalized. Square data points represent the simulation. Angle error of  $3.5^{\circ}$ 

interesting however is the fact that the analysis tool did not adjust the correlation length accordingly. One way to interpret this is to take the discontinuities into account. The small smooth "islands", though small, still represented a roughness discontinuity. This is the equivalent of generating a rough surface of a given length, and connecting it to another rough surface via a



Figure 4.16: Type 1, experiment 5:  $\theta_{inc} = 45^{\circ}$ ,  $\tau = 1.08 \times 10^{-4}$  m,  $\sigma = 1.93 \times 10^{-5}$  m,  $b = 2.058 \times 10^{-7}$  m. Measured data has been normalized. Square data points represent the simulation. Angle error of  $4^{\circ}$ 



Figure 4.17: Type 1, experiment 5:  $\theta_{inc} = 30^{\circ}$ ,  $\tau = 7.83 \times 10^{-5}$  m,  $\sigma = 2.18 \times 10^{-5}$  m,  $b = 2.26 \times 10^{-7}$  m. Measured data has been normalized. Square data points represent the simulation. Angle error of  $4^{\circ}$ 

smooth path. A similar specular peak, though much smaller, can be seen in figure 4.17. Thus, processing times below 40 seconds are to be avoided, as their analysis will be flawed. However it can be seen that the beam pattern converges to the simulation's predictions along the edges.

Additionally it can be concluded that the scattering surfaces are very absorbing, as evidenced by the noisy tails found in the measured profiles. If a comparison is made between figure 4.11 and any other original measurement, a difference of approximately 40 dB was found from peak to peak, before normalization. One could argue that this is due to the redistribution of scattered power over a wider angle and indeed this argument cannot be disproved because the measurement spans over half the plane only. However, if the simulation can be trusted then it can be used to show that there are indeed strong signal losses. The total normalized incident power measured from figure 4.11 is approximately  $-86 \,\mathrm{dB}$ , whereas that of all simulated surfaces is roughly  $-1300 \, \text{dB}$ . And since all of these measurements were made under the same conditions, this absorption effect cannot be directly related to the transmission spectra of air. Aluminum is generally considered as an excellent reflector for IR light, however no predictions whatsoever can be made regarding the light losses if the sample were placed in vacuum and at cryogenic temperatures. In addition to this, it has been planned to have the aluminum integrating sphere coated with a thin layer of gold once the fabrication is complete in order to further reduce losses. It the work of Klandermans (2013) an integrating sphere was coated, however this did not lead to any significant improvement whatsoever. No analysis has been made into the effects of losses.

The SPM simulation could not predict the light scattering because it was used outside of its domain of validity. The performance of the simulation depends on either the Rayleigh or Frauenhofer smooth surface criterion, where the difference being a factor of 4 in the denominator. Equation 4.43 shows the Rayleigh criterion. For the case of  $\lambda = 65 \,\mu\text{m}$ ,  $\sigma = 19.3 \,\mu\text{m}$  and  $\theta = 45^{\circ}$ , the inequality is violated and it can be seen in figure 4.18 that the pattern differs a lot from those computed using MoM. In fact, the difference is so big that both fields have not been shown together, because the field computed using the MoM technique will resemble a flat line.

$$\sigma < \frac{\lambda}{8\cos\left(\theta\right)} \tag{4.43}$$

Finally, the incident angle was changed to  $30^{\circ}$  to show that the simulation is not designed just for a single case. The simulation and measurements are in good agreement, however once again there was an error of  $4^{\circ}$ , which indicates that the error is systematic. Such an error could be explained by the fact that there was no way to align the bolometer with the rest of the setup.

## 4.5 Extrapolation

After proving that the simulation can accurately predict the scattering of a light band from a rough surface with a certain profile, it was used to extrapolate the scattering band. Since the SAFARI instrument can detect wavelengths as far as  $210 \,\mu\text{m}$ , it was investigated whether the current scattering surfaces, especially those produced with the F-16 and F-12 grains could scatter light close to a Lambertian pattern. The advantage of using a light band scattering simulation is that based on the Planck law, the contribution of the longer wavelengths can be directly compared to that of the shorter ones. Originally, the complete SAFARI band was simulated, however due to the Plank distribution, the contribution of the 210 µm wavelength was indistinguishable. Two comparisons have been made. One assuming the experiments have been done at room temperature, thus having a hot source temperature of 850 K and one assuming a cryogenic environment in which case the temperature would be close to 90 K. Absorption cannot



Figure 4.18: SPM simulation for  $\theta_{inc} = 45^{\circ}$ ,  $\tau = 1.08 \times 10^{-4}$  m,  $\sigma = 1.93 \times 10^{-5}$  m,  $b = 2.06 \times 10^{-7}$  m. The wavelength range is from 30 µm to 65 µm

be modeled, however it is presumed that in the cryogenic case it will be significantly lower.

Despite the big temperature differences, the normalization shows little change. This suggests that it might be possible to perform testing at room temperatures, which is much easier than in a vacuumed, cryogenic environment.

As it can be seen from figure 4.20, both fields are remarkably similar. Of course, many effects have been excluded, however this raises the possibility of comparing experiments carried out in a room to those made in a cryostat, a much more complex environment. In addition to this, it can be seen that in both cases the surfaces will fail to scatter. Finally, a surface that can scatter at these wavelengths was found via simulations to have a surface roughness of  $\sigma = 1.59300 \times 10^{-4}$  m and correlation length of  $\tau = 5.88000 \times 10^{-4}$  m. The effects of the Laplace diversity were too minor to cause any severe changes and in general there is no way to control them, thus they were kept at  $b = 2.05890 \times 10^{-7}$  m. Unfortunately such parameters were not achieved in the Taguchi training set and as such the NN cannot predict how to obtain such a surface.

# 4.6 Comparison of the SPM and MoM Simulations

At these wavelengths, the Rayleigh criterion is just barely satisfied. Thus, the SPM simulation can be compared more easily with the MoM technique as shown in figure 4.21. As it can be seen, the difference is much smaller once the Rayleigh criterion is satisfied. It is worth noting that the Rayleigh criterion is an empirical estimate and as such it's accuracy deteriorates when both sides of the inequality get closer. The advantage of the MoM approach over the SPM can be immediately seen - the SPM simulation cannot predict specular reflections. This should not be confused as a drawback of the theory, as it was originally designed to compute only the diffuse component of the scattered field.



(b) Planck distribution at 90 K

Figure 4.19: Comparison between the two power distributions



(b) Planck distribution at 90 K.  $\theta_{inc} = 45^{\circ}$ ,  $\tau = 1.08e^{-4}$ ,  $\sigma = 1.93e^{-5}$ ,  $b = 2.058e^{-7}$ Figure 4.20: Comparison between the two scattered fields from 180 µm to 210 µm



Figure 4.21: SPM simulation for  $\theta_{inc} = 45^{\circ}$ ,  $\tau = 1.08 \times 10^{-4}$  m,  $\sigma = 1.93 \times 10^{-5}$  m,  $b = 2.06 \times 10^{-7}$  m. The wavelength range is from 180 µm to 210 µm for a Planck temperature of 850 K

# Chapter 5 Complex Scattering Surfaces

In this chapter an investigation was made to determine whether samples which have been processed with different sized grains, under the same conditions, can create a surface which can scatter a wider ling band. Unfortunately the optical filters which were available for the radiometric measurements restricted the band between  $30 \,\mu\text{m}$  and  $65 \,\mu\text{m}$ , so the focus shifted on investigating whether the individual surface distributions could be separated for further analysis. Should it be possible to separate the individual contributions of each grain size, then simulations could easily be performed. For this purpose ICA was chosen. This investigation was only preliminary in nature and as such cannot be considered thorough by any means.

# 5.1 Situational and Theoretical Analysis

ICA is a numerically efficient blind source separation (BSS) technique and over the years has found several applications. The task of separating a known signal into more than one source signals without any further information is generally referred to as an inductive inference problem. The lack of information forces one to attempt to induce the most probable solution, which may or may not exist. The term blind is used to indicate that there is no prior information about how the signals have been mixed or even generated. Yet despite this the theory is capable of separating mixed source signals, and what makes this process even more fascinating is the mathematical elegance with which this is done. The reader is directed towards Naik and Kumar (2011) for an in depth overview of ICA.

Of course, as the process is statistically inductive in nature, there are several factors such as scaling, delay, and even causality which cannot be predicted. Luckily in most practical applications this information is not always necessary, as the goal is to recognize a pattern and not the signal's amplitude amplitude. The issue with causality is not favored in the biomedical field especially in analyzing brain patterns, however there are different theories which can circumnavigate the issue. The core of this theory is based on the assumption that different physical processes are unrelated. An example of the theory in practice is shown in figures 5.1, 5.2, 5.3.



(b) Original Signal 2

Figure 5.1: Two arbitrarily generated signals

The two arbitrary signals are randomly and linearly added together, after which ICA is applied to estimate the original two sources. Of course, the theory can be extended to any number of sources. As it can be seen, the ICA theory is capable of separating the two signals, however both signals appear to be inverted. This is a clear illustration of one of the drawbacks of this theory, however given the simplicity of it, such drawbacks can be neglected and worked around using conventional programming. This theory has not been applied to sandblasted surface profiles to determine whether multiple processings can be recognized.


(b) Original Signal 2

Figure 5.2: Two independent combinations of the arbitrary signals

#### 5.2 Conceptual Model

In addition to the surface analysis and simulation, the Infomax algorithm from Bell (1996) was converted to an analysis tool with minor peripheral modifications and was used. The goal was to investigate whether the individual contributions of a sample which has been processed with two different grain types sequentially could be analyzed. It was uncertain whether the final surface distribution could be considered as a linear, independent combination or a more complex mechanism was at hand. The algorithm that used was developed by Bell and Sejnowski (1995)



(b) Original Signal 2

Figure 5.3: The induced most likely original signals

and was taken straight from Bell (1996). It was slightly modified so that it can work directly with collected data from an Excel Workbook.

The basic working principles of the Informax algorithm will now be explained. An observation can be assumed as a vector, which is presumed to be random for there is no information about its composition,  $\mathbf{X} = [x_1, x_2, \dots, x_m]^T$  whose *m* elements are mixtures of *m* independent elements of another vector which must be presumed random  $\mathbf{S} = [s_1, s_2, \dots, s_m]^T$  given by:

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{S} \tag{5.1}$$

where A represents an  $m \times m$  mixing matrix. This matrix is unknown and it represents the environment in which the signals have interacted and as such can be of even more interest than the signals themselves. The goal of ICA is to find **W**, which is the inverse of A, that will give **Y**, the closest possible approximation of **S**.

$$\mathbf{Y} = \mathbf{W} \cdot \mathbf{X} \tag{5.2}$$

There are five assumptions which have made ICA theory possible. First, the vector **S** is assumed to contain only statistically independent sources. Independence allows for a probability density function (pdf) of m random variables,  $f(x_1, x_2, \ldots, x_m)$  to be represented as the product of m independent pdfs -  $f_1(x_1) f_2(x_2) \ldots f_m(x_m)$ . Statistical independence is then formulated as the equality between the expectation of the combined pdf and the product of the expectations of the individual pdfs. This can be expressed mathematically as follows:

$$E(f_1(x_i) f_2(x_j)) = E(f_1(x_i)) E(f_2(x_j))$$
(5.3)

Second, the mixing matrix must be square and have full rank, which can be interpreted as having the same number of linearly independent mixtures and sources. Third, the only source of noise in the model is the source vector  $\mathbf{S}$ . Fourth, it is assumed that the mean of the data is equal to 0. And finally, the source signals should not have a Gaussian pdf except for a single source only (Langlois et al., 2010).

The same code from Bell (1996) was combined with the surface generator of Bergstrom (2008) in order to determine whether the concept will work when assuming a linear combination of the surface processes.

#### 5.3 Research Design

Two different samples were fabricated. The first sample, referred to as "A" was processed for 3 minutes, at a pressure of 5 bar, at a distance of 6 cm first with grain type F-12 and later with grain type F-16. The second sample, referred to as "B" was processed for 4 minutes, at a pressure of 4 bar, at a distance of 7 cm also first with grain type F-12 and later with grain type F-16. The profiles were then measured using the Dektak instrument in the same manner as the type 1 and 2 samples. One measurement were made for each sample with a length of 3 cm. This measurement was then split in half to represent two different signals, because the distributions are ergodic.

#### 5.4 Research Results

First, the profiles were analyzed using the existing surface analysis tool. The KPSS test rejected the null hypothesis at all lags for sample "A", while failing to reject the last lag with sample "B", showing that the polynomial completely failed to detrend the data. Thus, a 5<sup>th</sup> order polynomial

was used in order to improve the detrending. Figure 5.4 shows the two detrended profiles and it can be observed that the profile of sample "B" is more uniform than that of sample "A"



(a) Detrended surface profile of sample "A"



(b) Detrended surface profile of sample "B"

Figure 5.4: Surface profiles of sample "A" and "B" after being detrended by a  $5^{\text{th}}$  order polynomial

As it can be seen in figure 5.5, sample "A" has a clear sinusoidal component, which can be attributed to the prolonged processing time of roughly 6 minutes. The KPSS test rejects the null hypothesis up to the  $30^{\text{th}}$  lag, while for sample "B" - up to the  $20^{\text{th}}$  lag. And this is to be expected, as the sinusoidal component is a sign of consistency among the noise, i.e. a drift.



(b) ACF of sample "B"

Figure 5.5: Comparison of the ACFs of sample "A" and "B". Emphasis on the sinusoidal component of "A"

Both histograms appear Gaussian in nature but upon closer inspection, especially at their normality plots in figure 5.7, it can be seen that the distributions are truncated and slightly asymmetric. The general fact that their distributions appear Gaussian suggests that the F-16 and F-12 grains are too similar and their distributions are indistinguishable from one another. Should grain wear be taken into consideration, larger grains wear out faster than smaller ones by either breaking into smaller grains, while smaller grains have been observed to have a higher durability.





Figure 5.6: Comparison between the extrapolated distributions of samples "A" and "B". Sample "A" has a truncated Gaussian

When the modified Infomax algorithm was applied to the profile measurements of samples "A" and "B", it failed to separate the two distributions. The reason why the algorithm failed is because both processes had identical parameters, with the grain types being the only difference. It was expected that this should make the distinction more apparent, however it turned out not to be the case. In conclusion it was found that using more than one grain type in the manufacturing process of a scattering surface does create slight asymmetries within the surface distributions.



(a) Histogram of AR model of sample "A"



(b) Histogram of AR model of sample "B"

Figure 5.7: Comparison between the normality plots of samples "A" and "B"

The Infomax algorithm might not be capable of separating the contributions of the two grain types either due to their similarity, due to the fact that both distributions are Gaussian in nature, or due to the fact that the process is not linear. The last two reasons violate the basic assumptions behind ICA and make it inapplicable for this analysis.



(a) Half of profile "B" treated as a linear sum of two unknown signals



Figure 5.8: Other half of profile "B" treated as a linear sum of two unknown signals

As it can be seen, figure 5.8 is almost identical to figure 5.9, which means that no separation has occurred. The Infomax algorithm was also implemented into a surface generator simulation, where it was found to be capable of separating surfaces with different truncated Gaussian dis-



(b) Second estimated signal

Figure 5.9: The two new estimated signals are the original signals. No separation has occurred

tributions, as shown in section 5.1. Unfortunately such distributions cannot be created using sandblasting so other techniques must be investigated.

# Chapter 6 Conclusion

The theories of scattering of electromagnetic waves, numerical techniques, neural networks, independent component analysis, statistical investigations, and time-series analysis have been implemented to create a set of basic tools which will serve as the starting point for the development of the integrating sphere, as part of the SAFARI calibration process. These tools have been implemented and tested, and apart from ICA can be directly used in conjunction with sandblasting.

It has been shown that care must be taken when analyzing surface profiles, despite the task's simplistic nature, due to the nonuniform drift component distributions. Thus a choice was made to create a very basic self adjusting detrending approach, instead of relying on rough surface standard filters. This approach was not perfect, especially when dealing with higher frequency drift components, however by carefully implementing other monitoring statistics the detrending performance could be evaluated. Multiple statistical tools were used in the analysis of the scattering surface samples so that as much information as possible could be gathered that would reveal the properties of interest.

A very high frequency truncated Laplace distribution was found after taking the difference of the surface profile. The distribution was modeled as noise and was added to the surface generating algorithm. The result was a very slight increase in accuracy, however the implementation of this error term requires higher resolutions and more precise experimental setup. The difficulty with implementing the noise component was the fact that it was two orders of magnitude smaller than the other structures. This meant that higher surface resolutions would have to be used in order to include the effect. This the improvement is considered to be very minor.

Several experiments were performed to validate the assumptions, beginning from fabricating scattering surfaces, using the Taguchi's orthogonal arrays, and profile measurements, to a scattered band measurement from  $30 \,\mu\text{m}$  to  $65 \,\mu\text{m}$ . The sandblasting and band scattering experimental setups had several flaws whose presence was recognized from the obtained information, despite the author's best efforts.

Based on the analyzed data from the surface analysis tool, a neural network was created with the aim of aiding a human operator with the manufacturing process. It was shown that the network is capable of estimating system parameters within the same order of magnitude without any negative numbers, given the minimum of statistical information with which it was trained. Should the network be trained with more data, it is expected that it's performance will increase, thus

#### CHAPTER 6. CONCLUSION

making it estimate surface control parameters more and more accurately. This will eliminate the need to guess the manufacturing controls and will help reduce the number of faulty components and material waste.

The developed simulation from the works of Tsang et al. (2001); Bergstrom (2008) was shown to agree with experimental data, except for experiment 4 where roughness discontinuities were found to be present upon optical inspection. In addition to this a systematic measuring error of  $4^{\circ}$  was found present in all samples. In addition to this it was shown that with the current fabrication capabilities it is not possible to create a scattering surface which can scatter up to 210 µm. Thus a new manufacturing technique must be chosen. A comparison was made between a 90 K and a 850 K band scattering simulation and it was found that they are nearly identical. This was due to the normalized Planck distribution being employed. This suggests that it might be possible to completely forgo light scattering measurements in a cryogenic environment, depending on how severe the absorption of light becomes within the material of interest. The SPM simulation could be used successfully only in the longer wavelengths range. It was compared to the MoM technique and it was found to still differ. The SPM simulation was partially used out of context, as it only approximated the diffuse component of a scattered wave. Regardless, it can still be useful if the Rayleigh criterion is satisfied. A surface which could scatter well up to 210 µm was found, however there are no current means of manufacturing it.

The Infomax algorithm by Bell and Sejnowski (1995) was implemented into a simulation and an analysis tool and was tested on multigrain samples, however it could not separate the two profiles. This is because both profiles were of a Gaussian nature, which contradicted one of the five assumptions around ICA, and also because the process in general was not linear. However the Infomax simulation could properly separate sufficiently different profiles, thus showing that should a more careful and thought out experiment be created, ICA might produce meaningful results.

# Chapter 7 Discussions

The development of a 3D simulation is much more difficult than a 2D one. Several attempts were made at creating such, however the avoidance of the singularity of the 3D Green's function proved too complex. As future recommendation, the development of a vector field 3D simulation would provide a much more robust tool in the analysis of the process of light distribution and it could be used to analyze other effects as well, such as light behaviour in waveguides. The development of a 3D simulation will also require more processing power. Thus it is recommended to obtain more powerful computers that can reduce the simulation time as this will significantly reduce the development process. The creation of a dedicated test setup for the measurement of light scattering will also contribute to the further understanding of the phenomenon which will lead to more accurate design criteria. As discussed in this thesis, the experimental setup had many possible areas of failure. In addition to this other manufacturing techniques must be investigated in order to obtain a longer correlation length and a larger standard deviation. Areas such as spark erosion or chemical treatment should be investigated and properly modeled.

Other areas that could benefit from these suggestions include the development of scattering or absorbing coatings, as currently there is no other means to verify then apart from trial and error testing. Such coatings can be used to improve the signal to noise ratio in certain setups.

# Appendices

#### Appendix A

## Data Analysis Code

```
1 % Code used to analyze the profile measurements of roughened samples.
       Tt.
 \% also performs a series as tests to analyze the information.
2
3
  clear all
4
  close all
\mathbf{5}
6
  directory_input = uigetdir('', 'Choose directory with input data
7
      files:');
                         % Select input directory
  f_det = (xlsread(strcat(directory_input, '\MG_2_2_1.xlsx'), 'MG_2_2_1
8
      ', 'B21:B54020').*1e-9)'; %Choose your own files. Convert to nm
     from m
9
```

```
10 %
```

18

```
<sup>19</sup> L = 30000e-6; %Measured length. Obtained from the original sample.
Dont forget to change!
<sup>20</sup> N = length(f_det); %Data points
<sup>21</sup> dx = L/N; %Original step
<sup>22</sup> x = linspace(-L/2,L/2,N); %Split in two halves
<sup>23</sup>
<sup>24</sup> %
```

```
%Detrend the surface. Very basic, based on assumption that the drift
25
      is of
  %low frequency.
26
27
   [P,S,mu_hat] = polyfit(x,f_det,5);
^{28}
   f1 = polyval(P, x, S, mu_hat);
29
30
   figure ('name', 'Original Surface and 2nd Order Detrending Polynomial')
31
   plot(x, f_det, '-', x, f1, '--');
32
  xlabel('Lenght [m]');
ylabel('Heigth [m]');
33
34
   title('Original Surface and 2nd Order Detrending Polynomial');
35
   grid on
36
   print (format, resolution, streat (directory_output, '\Original Surface
37
       and 2nd Order Detrending Polynomial'))
38
   f_det = f_det - f1; \% Detrending occurs here
39
40
  %Plot the detrended surface
41
  figure('name', 'Detrended Surface Profile');
42
  plot(x, f_det);
43
  xlabel('Lenght [m]');
44
  ylabel('Heigth [m]');
45
  title('Detrended Surface Profile');
46
   grid on
47
   print (format, resolution, streat (directory_output, '\Detrended
48
      Surface Profile '))
49
  %
50
  %Plot partial correlation, to compare with the normalized
51
      autocorrelation
  %function taken from David Bergstrm. The difference in gradient is
52
  % evidence of a stationary process, and the ADF test (see bellow)
53
      rejects the null
  %hypothesis at the 5% confidence interval when the lags are more
54
  % than 1 (^{3}0 and greater). This is due to the imperfect detrending.
55
56
   figure
57
   parcorr(f_det,100)
58
   xlabel('Lag');
59
  ylabel('Partial Autocorrelations');
60
  title ('Sample Partial Autocorrelation Function');
61
  grid on
62
```

```
63 print (format, resolution, streat (directory_output, '\Sample Partial
```

```
Autocorrelation Function'))
```

64 65 %

```
%Estimate an AR model and extrapolate
66
67
   N_{max} = 5e6; %Evolve the process further
68
   L_{max} = N_{max} \cdot dx;
69
   x_{AR} = linspace(-L_max/2, L_max/2, N_max); %Only used for plotting
70
71
   [A, NoiseVariance] = arburg(f_det, 10);
72
   f_AR = filter(1, A, sqrt(NoiseVariance) * randn(N_max, 1));
73
74
   figure('name', 'Extrapolated estimated AR model');
75
   plot(x_AR, f_AR);
76
   xlabel('Lenght [m]');
77
   ylabel('Heigth [m]');
78
   title('Extrapolated estimated AR model');
79
   grid on;
80
   print (format, resolution, streat (directory_output, '\Extrapolated
81
       estimated AR model'));
82
   %
83
   %Histograms of the original data and of the extrapolated AR model.
84
85
   m = diff(f_det); %Difference the detrended surface.
86
87
   Laplace_b = sum(abs(m))/(N-1); % Estimate the b parameter for the
88
       Laplacian distribution. N-1 because diff takes away 1 index
89
   bins = 80; %Number of bins
90
91
    [hdf_raw, bc_raw] = hist(f_det, bins);
92
    [hdf_AR, bc_AR] = hist(f_AR, bins);
93
    [hdf_m, bc_m] = hist(m, bins);
94
95
   figure('name', 'Bar Graph of the Historgram');
96
   bar(bc_raw, hdf_raw);
97
   xlabel('Surface Height');
98
   ylabel('Probability');
99
   title ('Histogram of Height Distribution Function for Height of
100
       Surface Profile');
   grid on;
101
   print(format, resolution, strcat(directory_output, '\Histogram of
102
       Height Distribution Function for Height of Surface Profile '));
103
```

```
figure ('name', 'Bar Graph of the Histogram of the AR Model');
104
   bar(bc_AR,hdf_AR);
105
   xlabel('surface height');
106
   ylabel('probability');
107
   title ('Histogram of Height Distribution Function for Height of AR
108
       Model Surface Profile ');
   grid on;
109
   print (format, resolution, strcat (directory_output, '\Histogram of
110
       Height Distribution Function for Height of AR Model Surface
       Profile '));
111
   figure ('name', 'Bar Graph of the Difference Historgram');
112
   bar(bc_m, hdf_m);
113
   xlabel('Surface Height');
114
   ylabel('Probability');
115
   title ('Histogram of Height Distribution Function for Difference of
116
       Surface Profile');
   grid on;
117
   print (format, resolution, strcat (directory_output, '\Histogram of
118
       Height Distribution Function for Difference of Surface Profile'));
119
   %
120
   %Autocovariance function as defined by David Bergstrm from his
121
       thesis. It
   %shows that the oscillations are slow compared to the number of data
122
   %points, because of the high resolution.
   lags = linspace(0, x(N)-x(1), N); %Lag lengths
   c = xcov(f_det, 'coeff'); %Autocovariance function
   acf = c(N:2*N-1); %Right sided version
```

```
123
124
125
126
127
128
   k = 1;
129
   while (acf(k) > 1/exp(1)) %Correlation lenght calculation
130
        k = k+1;
131
   end
132
133
   correlation_lenght = 1/2*(x(k-1)+x(k)-2*x(1));
134
135
    figure ('name', 'Normalized ACF');
136
   plot(lags, acf);
137
   xlabel('Lag Length');
138
    ylabel('Normalized ACF');
139
   title('Normalized ACF');
140
    grid on;
141
    print (format, resolution, strcat (directory_output, '\Normalized ACF')
142
       );
143
```

```
[muhat, sigmahat] = normfit(f_det); %Meand and standard deviation of
144
       the original data
145
   %
146
   %The Normalized ACF of the extrapolated AR model is also computed.
147
148
   lags_AR = linspace(0, x_AR(N_max) - x_AR(1), N_max); % lag lengths
149
   c_AR = xcov(f_AR, 'coeff'); %Autocovariance function
150
   acf_AR = c_AR(N_max:2*N_max-1); %Right sided version
151
152
   k = 1;
153
   while (acf(k)>1/exp(1)) %Correlation lenght calculation
154
        k = k+1;
155
   end
156
157
   correlation_lenght_AR = 1/2*(x_AR(k-1)+x_AR(k)-2*x_AR(1));
158
159
   figure('name', 'Normalized AR Model ACF');
160
   plot(lags_AR, acf_AR);
161
   xlabel('lag length');
162
   ylabel('Normalized AR Model ACF');
163
   title('Normalized AR Model ACF');
164
   grid on;
165
   print (format, resolution, strcat (directory_output, '\Normalized AR
166
       Model ACF'));
167
   [muhat_AR, sigmahat_AR] = normfit(f_AR); %Meand and standard deviation
168
        of the extrapolated surface
   surface_ratio = sigmahat_AR/correlation_lenght_AR; %The scattering
169
       ratio is computed
170
   %
171
   \% Z-Scoring of the sample data so that it can be compared to a
172
       standard normal deviate.
   \% In addition the ADF test is performed to
173
   step = 1; %Introduce a lag of 100 steps. Rationale is that the
174
       correlation created by the grains spans over several hundred data
       points and the kstest and jbtest functions ' lag lenght cannot be
       adjusted internally.
175
   Y = (f_det - muhat)/sigmahat; %Real sample
176
   Y_AR = (f_AR - muhat_AR) / sigmahat_AR; % AR model sample
177
178
179
   Y = Y(1:step:N); %Introduce a lag of 'step' points.
   Y_AR = Y_AR(1: step:N_max);
180
```

```
181
182
```

183

184

185

186

187

189 190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

208 209

210

211

212

213

214215

216

% 207

%

```
%Perform a bunch of hypothesis tests on the detrended sample
   [h1, pValue1, stat1, cValue1] = kpsstest(f_det, 'lags', 10:10:100); \%
       Test for lags starting from 10 to 100 using a step of 10
   [h2, pValue2, stat2, cValue2] = adftest(f_det, 'model', 'ts', 'alpha'
       ,0.05, 'lags',100); %Here a 100 lags are chosen and the number can
       be changed, obviously.
   [h3, pValue3, k3, cValue3] = kstest(Y, 'Alpha', 0.05); %The kstest cannot
       have its lags set directly, so the undersampled z-values are used.
   [h4, pValue4, jbstat, cValue4] = jbtest(f_det, 0.05); %Second test to
       check whether the data has a normal distribution of some sort.
   %
188
   %Visual normality test for real data
   figure ('name', 'Normality Test Visualisation for Measured Data');
   subplot(2,1,1)
   [f_dist, Y_values] = ecdf(Y);
   J = plot(Y_values, f_dist);
   grid on;
   hold on;
```

legend ([J K], 'Empirical CDF', 'Standard Normal CDF', 'Location', 'SE');

print (format, resolution, strcat (directory\_output, '\Normality Test

figure ('name', 'Normality Test Visualisation for Extrapolated AR Model

 $K = plot(Y_values, normcdf(Y_values), 'r_{values});$ 

Visualisation for Measured Data'))

 $K = plot(Y_values, normcdf(Y_values), 'r-');$ 

%Visual normality test for extrapolated AR model

set(J, 'LineWidth', 2);

set(K, 'LineWidth', 2);

set(np, 'LineWidth',2);

 $[f_AR, Y_values] = ecdf(Y_AR);$ 

 $J = plot(Y_values, f_AR);$ 

subplot(2,1,2)

grid on;

'); subplot(2,1,1)

grid on;

hold on;

np = normplot(Y);

88

```
set(J, 'LineWidth', 2);
217
   set(K, 'LineWidth',2);
218
   legend ([J K], 'Empirical CDF', 'Standard Normal CDF', 'Location', 'SE');
219
   subplot (2,1,2)
220
   np = normplot(Y_AR);
221
   set(np, 'LineWidth',2);
222
   print (format, resolution, streat (directory_output, '\Normality Test
223
       Visualisation for Extrapolated AR Model'))
   %
224
```

 $_{225}$  %Save the test results as text.

226

- Names = ['Lag coefficients: %1.4f, %1.4f, %1.4f, %1.4f, %1.4f, %1.4f 227 , %1.4f, %1.4f, %1.4f, %1.4f, %1.4f \n\n', 'Noise Variance: %1.5e [m] \n\n', 'Estimated Correlation Lenght: %1.5e [m] \n', 'Estimated Mean:  $\%1.5e [m] \ n'$ , 'Estimated Standard Deviation:  $\%1.5e [m] \ n \ n'$ , 'Laplace diversity: %1.5e [m]\n\n', 'Estimated AR Model Correlation Lenght:  $\%1.5e\ [m]\n',$  'Estimated AR Model Mean: %1.5e $[m] \ n'$ , 'Estimated AR Model Standard Deviation: %1.5e  $[m] \ n'$ , Surface Ratio: %1.5f\n\n', 'KPSS Test Decissions: %1.0f %1.0f %1.0 f %1.0f %1.0f %1.0f %1.0f %1.0f %1.0f %1.0f  $\n$  , 'KPSS Test  $\rm p$ values: %1.5f %1.5f %1.5f %1.5f %1.5f %1.5f %1.5f %1.5f %1.5f %1.5 'KPSS Test Statistics: %1.5f %1.5f %1.5f %1.5f %1.5f %1.5f  $f \mid n'$ , %1.5f %1.5f %1.5f %1.5f \n', 'KPSS Test Critical Values: %1.5f \n\n', 'ADF Test Decission: %1.0f n', 'ADF Test p-value: %1.5f n', 'ADF Test Statistic: %1.5f  $\n',$  'ADF Test Critical Value: %1.5f  $\n',$  'KS Test Decission: %1.0f \n', 'KS Test p-value: %1.5f \n', 'KS Test k -value: %1.5f \n', 'KS Test Critical Value: %1.5f \n\n', 'Jarque-Bera Test Decission: %1.0f \n', 'Jarque-Bera Test p-value: %1.5f  $\backslash$ n', 'Jarque-Bera Statistic: %1.5f \n', 'Jarque-Bera Test Critical Value:  $\%1.5 f \ln n'$ ;
- M = [A, NoiseVariance, correlation\_lenght, muhat, sigmahat, Laplace\_b, correlation\_lenght\_AR, muhat\_AR, sigmahat\_AR, surface\_ratio, h1, pValue1, stat1, cValue1, h2, pValue2, stat2, cValue2, h3, pValue3, k3, cValue3, h4, pValue4, jbstat, cValue4];

- <sup>231</sup> fprintf (fid , Names , M);
- <sup>232</sup> fclose (fid);

<sup>229</sup> 

<sup>230</sup> fid = fopen(streat(directory\_output, '\Parameters.txt'), 'wt');

#### Appendix B

### **ICA Simulation Code**

```
<sup>1</sup> %This is an attempt to try the Independent Component Analysis (part
      of the
  %Blind Source Separation theory) using the algorithm designed by Tony
2
       Bell
  %et al.
3
4
  %setup
5
6
  %
       sweep - How many times I've gone through the data
7
  %
      P - How many timepoints in the data
8
  %
      N - How many input (mixed) sources there are
9
  %
      M - How many outputs there are
10
  %
11
      L - Learning rate
  %
      B – Batch block size (ie: how many presentations per weight
12
      update)
  %
      t - time step
13
  %
       sources - NxP matrix of the N sources I read in
14
15
  %
      x - NxP matrix of mixtures
  %
      u - MxP matrix of hopefully unmixed sources
16
17 %
      A – NxN mixing matrix
 %
      w – MxN unmixing matrix
18
  %
      wz - Zero-phase whitening: a matrix used to remove correlations
19
      from
  %
       between the mixtures x. Useful as a preprocessing step.
20
  %
       noblocks - How many blocks in a sweep
21
  %
       oldw - Value of w before the last sweep
^{22}
  %
       delta - w-oldw
23
 %
       olddelta - Value of delta before the last sweet
^{24}
  %
       angle - Angle in degrees between delta and olddelta
25
  %
       change - Squared length of delta vector
26
 %
       Id – An identity matrix
27
28 %
       permute - A vector of length P used to scramble the time order of
       the
29 %
       sources for stationarity during learning.
```

```
30
^{31}
   clear all
32
   close all
33
34
35
   P = 10000;
   N = 2;
36
   L = 10;
37
38
   %Low frequency large amplitude stochastic controls.
39
   h1 = 0.3; %Height or standard deviation 0.5
40
   c1 = 0.2; %Grain size descriptor
^{41}
^{42}
   %High frequency small amplitude stochastic controls.
43
   h2 = 0.4; %Penetration depth descriptor (related to grain geometry
44
       possibly – a measure of the roundness of the grain(?) ) 0.05
   c2 = 0.09; %Irregularity size descriptor
^{45}
46
   format = '-dpng';
47
   resolution = '-r300';
^{48}
   directory_output = uigetdir('', 'Choose directory with output data
49
       files:');
50
   %
51
   %Distributions control
52
53
   t = linspace(-L/2, L/2, P);
54
55
   pd1 = makedist('Normal');
56
   t1 = truncate(pd1, -inf, inf);
57
58
   pd2 = makedist('Normal');
59
   t2 = truncate(pd2, -inf, inf);
60
61
  %
62
   %Low frequency stochastic process
63
64
   Z1 = h1.*random(t1, 1, P);
65
   F1 = \exp(-(2*((t).2)/(c12)));
66
   f1 = 2/\operatorname{sqrt}(\operatorname{pi}) * L/P/c1 * \operatorname{ifft2}(\operatorname{fft2}(\operatorname{Z1}) * \operatorname{fft2}(\operatorname{F1}));
67
68
   %
69
```

70 %High frequency stochastic process

71

```
Z2 = h2.*random(t2, 1, P);
  72
        F2 = \exp(-(2*((t).2)/(c22)));
 73
        f2 = 2/sqrt(pi)*L/P/c2*ifft2(fft2(Z2).*fft2(F2));
 74
  75
        %
 76
 77
         figure('name', 'Surface Profile 1');
 78
         plot(t, f1);
  79
        xlabel('Data Points')
  80
         ylabel('Amplitude [Arbitrary Units]');
  81
         title('ICA Simulation Surface Profile 1');
 82
         grid on
  83
        hold on;
  84
         print(format, resolution, strcat(directory_output, `\
 85
                  ICA_Simulation_Surface_Profile_1'))
 86
         figure('name', 'Surface Profile 2');
  87
         plot(t, f2);
 88
         xlabel('Data Points')
 89
         ylabel('Amplitude [Arbitrary Units]');
 90
         title ('ICA Simulation Surface Profile 2');
 91
         grid on
 92
        hold on;
 93
         print(format, resolution, strcat(directory_output, `\
 ^{94}
                  ICA_Simulation_Surface_Profile_2'))
 95
        %Randomly combined sources, similar signals results in poor
 96
                  performance of the system
         sources = [5*rand(1,1)*f1-7*rand(1,1)*f2; -6*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand(1,1)*f1+3*rand
 97
                 f2];
 98
         figure('name', 'Combined Profile 1')
 99
         plot(t, sources(1,:)')
100
         xlabel('Data Points')
101
         ylabel('Amplitude [Arbitrary Units]');
102
         title('ICA Simulation Combined Profile 1');
103
         grid on
104
        hold on;
105
         print (format, resolution, streat (directory_output, '\
106
                  ICA_Simulation_Combined_Profile_1 '))
107
         figure('name', 'Combined Profile 2')
108
         plot(t, sources(2,:)')
109
         xlabel('Data Points')
110
         ylabel('Amplitude [Arbitrary Units]');
111
         title ('ICA Simulation Combined Profile 2');
112
```

```
grid on
113
   hold on;
114
   print(format, resolution, strcat(directory_output, `\
115
       ICA_Simulation_Combined_Profile_2'))
116
117
   %
118
119
   permute = randperm(P);
                                  %generate a permutation vector
120
                                  %time scrambled inputs for stationarity
   s = sources(:, permute);
121
122
                                %Mix input signals (permuted)
   x = s;
123
                                %Make mixed sources (not permuted)
   mixes = sources;
124
125
   Sphere the data
126
   mx = mean(mixes');
127
   c = cov(mixes');
128
129
   x = x - mx' * ones(1, P); %Subtract means from mixes
130
131
                             % Get decorrelating matrix
   wz = 2 * inv(sqrtm(c));
132
                             % Decorrelate mixes so cov(x') = 4*eye(N)
   x = wz * x;
133
134
   %
135
136
   w = eye(N); %Initial unmixing matrix. Can be random numbers
137
   M = size(w, 2); % M=N usually
138
139
   sweep = 0;
140
   oldw = w;
141
   olddelta = ones(1,N*N);
142
   Id = eye(M);
143
144
   %
145
146
   T = 0.0001;
147
   B=30;
148
149
   sweep = sweep +1;
150
   t = 1;
151
152
   noblocks = fix(P/B);
153
```

```
BI = B*Id;
154
155
    for t = t:B:t-1+noblocks*B
156
157
158
        u = w * x (:, t : t + B - 1);
        w = w+T*(BI+(1-2*(1./(1+exp(-u))))*u')*w;
159
160
   end
161
162
    delta = reshape(oldw-w, 1, M*N)
163
    change = delta * delta '
164
    angle = acos((delta*olddelta')/sqrt((change)*(olddelta*olddelta'))))
165
166
   oldw = w;
167
168
   %
169
```

```
170
   uu = w*wz*mixes; %Separated sources
171
172
   f_{reconstructed} = uu(1,:)+uu(2,:); %Reconstruct the original signal
173
174
   ratio = mixes (2, :) . / f_reconstructed;
                                               %Measure the ratio between
175
       them
   mu = mean(ratio);
                          %Find the mean
176
177
   figure('name', 'Error Ratio');
                                       %Plot for entertainment
178
   plot(ratio);
179
180
   new_uu(1,:) = mu*uu(1,:);
                                  %Renormalize
181
                                  %Renormalize
   new_uu(2,:) = mu*uu(2,:);
182
183
184
   figure ('name', 'Renormalized Separated Profile 1'); %Renormalized
185
       unmixed sources
186
   plot(new_uu(1,:));
   xlabel('Data Points')
187
   ylabel('Amplitude [Arbitrary Units]');
188
   title ('ICA Simulation Renormalized Profile 1');
189
   grid on
190
   hold on;
191
   print (format, resolution, streat (directory_output, '\
192
       ICA_Simulation_Renormalized_Separated_Profile_1'))
193
   figure ('name', 'Renormalized Separated Profile 2'); %Renormalized
194
       unmixed sources
195
   plot (new_uu (2,:));
   xlabel('Data Points')
196
```

```
197 ylabel('Amplitude [Arbitrary Units]');
198 title('ICA Simulation Renormalized Profile 1');
199 grid on
200 hold on;
201 print(format, resolution, strcat(directory_output, '\
ICA_Simulation_Renormalized_Separated_Profile_2'))
202
203 %
```

#### Appendix C

## ICA Analysis Code

```
1 %ICA Analysis code, originally developed by Tony Bell, slightly
      modified to
  %be used for surface analysis purposes
2
  clear all
3
  close all
4
5
  directory_input = uigetdir('', 'Choose directory with input data
6
                                                        % Select input
      files:');
      directory
  data1 = xlsread(strcat(directory_input, '\MG_2_2.xlsx'), 'MG_2_2', '
7
      B21:B27020');
                           %Choose your own files
  data2 = xlsread(strcat(directory_input, '\MG_2_2 - Copy.xlsx'), '
8
      MG_2_2', 'B21:B27020');
9
  N = length(data1);
10
  t = (1:N)';
11
^{12}
  format = '-dpng';
^{13}
  resolution = '-r300';
14
  directory_output = uigetdir('', 'Choose directory with output data
15
      files:');
16
  %
17
```

```
18 %Detrend here
19
20 ica_data = [data1,data2];
21
22 for g =1:2
23
24 [P,S,mu_hat] = polyfit(t,ica_data(:,g),3);
25 f1 = polyval(P,t,S,mu_hat);
26
```

```
27
   plot(t,ica_data(:,g), '-', t, f1, '---');
28
  xlabel('Lenght [m]');
^{29}
   ylabel('Heigth [m]');
30
   title(strcat('Original Surface and 2nd Order Detrending Polynomial',
31
      \operatorname{num2str}(g)));
   grid on
32
   print(format, resolution, strcat(directory_output, strcat(`\
33
      Original_Surface_and_2nd_Order_Detrending_Polynomial_', num2str(g))
      ))
34
   ica_data(:,g) = ica_data(:,g) - f1; \% Detrend. Very primitive.
35
36
  %Plot the detrended surface
37
  figure('name', strcat('Detrended Surface Profile ', num2str(g)));
38
   plot(t, ica_data(:,g));
39
  xlabel('Lenght [m]');
40
  ylabel('Heigth [m]');
41
  title(strcat('Detrended Surface Profile', num2str(g)));
42
   grid on
43
   print (format, resolution, strcat (directory_output, strcat (')
44
      Detrended_Surface_Profile_', num2str(g))))
45
  end
46
47
  %
48
  P = 2;
49
   sources = ica_data'; %These are the sources that get mixed up
50
      together, right?
51
  permute = randperm(N);
                               %generate a permutation vector
52
  s = sources(:, permute);
                               %time scrambled inputs for stationarity
53
54
                             %Mix input signals (permuted)
55
  x = s;
  mixes = sources;
                             %Make mixed sources (not permuted)
56
57
  %=
           _____Sphere the data
58
  mx = mean(mixes');
59
  c = cov(mixes');
60
61
  x = x - mx' * ones(1,N); %Subtract means from mixes
62
63
                           % Get decorrelating matrix
  wz = 2 * inv(sqrtm(c));
64
65
  x = wz * x;
                           % Decorrelate mixes so cov(x') = 4*eye(N)
66
```

```
67 %
```

```
68
  w = eye(P); %Initial unmixing matrix. Can be random numbers
69
70
  M = size(w, 2); % MeN usually
71
   sweep = 0;
72
   oldw = w;
73
   olddelta = ones(1, P*P);
74
  Id = eye(M);
75
76
  %
77
```

```
78
   L = 0.0001;
79
   B = 30;
80
81
   sweep = sweep +1;
82
83
   r = 1;
84
   noblocks = fix(N/B);
85
   BI = B*Id;
86
87
   for r = r:B:r-1+noblocks*B
88
89
        u = w * x (:, r : r + B - 1);
90
        w = w+L*(BI+(1-2*(1./(1+exp(-u))))*u')*w;
91
92
   end
93
94
   \%[M,N] = size(w);
95
   delta = reshape(oldw-w, 1, M*P);
96
   change = delta*delta';
97
   angle = acos((delta*olddelta')/sqrt((change)*(olddelta*olddelta')));
^{98}
99
   old w = w;
100
   fprintf('****sweep = %d, change = %.4f angle=%.1f deg., [N%d, M%d, P%
101
       d, B\%d, L\%.5f]\n', sweep, change, 180* angle/pi, N,M,P,B,L);
102
   uu = w*wz*mixes; %Separated sources
103
104
   f-reconstructed = uu(1,:) + uu(2,:); %Reconstruct the original
105
       signal
106
   ratio = mixes (2,:)./f_reconstructed;
                                                %Measure the ratio between
107
       them
108
```

```
mu = mean(ratio)
                         %Find the mean
109
110
   figure('name', 'Error Ratio');
                                       %Plot for entertainment
111
   plot(ratio);
112
   xlabel('Lenght [m]');
113
   ylabel ('Heigth [m]');
114
   title('Error_Ratio');
115
   grid on
116
   print(format, resolution, strcat(directory_output, '\Error_Ratio'))
117
118
   new_uu(1,:) = mu*uu(1,:);
                                  %Renormalize
119
   new_uu(2,:) = mu*uu(2,:);
                                  %Renormalize
120
121
122
   figure ('name', 'Renormalized Separated Profile 1'); %Renormalized
123
       unmixed sources
   plot (new_uu (1,:));
124
   xlabel('Lenght [m]');
125
   ylabel('Heigth [m]');
126
   title('Renormalized Separated Profile 1');
127
128
   grid on
   print(format, resolution, strcat(directory_output, '\
129
       Renormalized_Separated_Profile_1'))
130
   figure ('name', 'Renormalized Separated Profile 2'); %Renormalized
131
       unmixed sources
   plot (new_uu (2,:));
132
   xlabel('Lenght [m]');
133
   ylabel ('Heigth [m]');
134
   title ('Renormalized Separated Profile 2');
135
   grid on
136
   print(format, resolution, strcat(directory_output, `\
137
       Renormalized_Separated_Profile_2'))
138
139
   %
140
   %Height distribution function for first component
141
   bins = 80; %Number of bins
142
143
   [hdf1, bc1] = hist(new_uu(1, :), bins);
144
145
   figure ('name', 'Bar graph of the Historgram of Surface 1');
146
   bar(bc1, hdf1);
147
   xlabel('surface height')
148
   ylabel('probability')
149
```

```
150 title('Histogram of height distribution function for height of surface profile y = f(x)')
```

```
%Height distribution function for second component
153
154
    [hdf1, bc1] = hist(new_uu(2, :), bins);
155
156
    figure ('name', 'Bar graph of the Historgram of Surface 2');
157
    bar(bc1, hdf1);
158
    xlabel('surface height')
159
    ylabel('probability')
160
    title ('Histogram of height distribution function for height of
161
        surface profile y = f(x)')
    print \,(\, format \,, \ resolution \,, \ strcat \,(\, directory\_output \,, \ ` \backslash
162
        Bar_graph_of_the_historgram_of_Surface_2'))
   %
163
```

```
%Autocovariance function for first component
164
165
   lags = linspace(0, t(N)-t(1), N); % lag lengths
166
167
   %Autocovariance function calculation
168
169
   c = xcov(new_uu(1,:), 'coeff'); %Autocovariance function
170
   acf = c(N:2*N-1); %Right sided version
171
172
   %Correlation lenght calculation
173
   k = 1;
174
    while (\operatorname{acf}(k) > 1/\exp(1))
175
        k = k+1;
176
   end
177
178
   c1 = 0.5*(t(k-1)-t(k)-2*t(1)); %Correlation length
179
180
   figure('name', 'Normalized ACF of Surface 1');
181
   plot(lags, acf);
182
    xlabel('lag length')
183
   vlabel('Normalized ACF')
184
    title ('Plot of the Normalized ACF of Surface 1')
185
   grid on
186
   print(format, resolution, strcat(directory_output, '\
187
       Normalized_ACF_of_Surface_1'))
   %
188
```

189 %Autocovariance function for second component

```
190
   lags = linspace(0, t(N)-t(1), N); \% lag lengths
191
192
   %Autocovariance function calculation
193
194
   c = xcov(new_uu(2,:), 'coeff'); %Autocovariance function
195
   acf = c(N:2*N-1); %Right sided version
196
197
   %Correlation lenght calculation
198
   k = 1;
199
   while (\operatorname{acf}(k) > 1/\exp(1))
200
        k = k+1;
201
   end
202
203
   c1 = 0.5*(t(k-1)-t(k)-2*t(1)); %Correlation length
204
205
   figure('name', 'Normalized ACF of Surface 2');
206
   plot(lags, acf);
207
   xlabel('lag length')
208
   ylabel ('Normalized ACF')
209
   title ('Plot of the Normalized ACF of Surface 2')
210
   grid on
211
   print(format, resolution, strcat(directory_output, `\
212
       Normalized_ACF_of_Surface_2'))
213 %
```

#### Appendix D

19

## MoM Band Scatter Simulation Code

```
<sup>1</sup> %This is a 2D scalar wave random surface scattering simulation
2
  clear all;
3
  close all;
^{4}
5
  scale = 10; %Dangerous idea
6
  L = 100/scale; %Scale everything
7
               %Show me the number of sampling points per new surface
8
  N = 700;
9
                                     %Number of wavelenghts. Minimum is 2.
  N_w = 10;
10
       Bigger is better.
                                     %Number of scattered points. Bigger
  N_{f} = 180;
11
      is better
                                     %Number of Monte Carlo simulations.
  N_{mc} = 100;
12
      Bigger is better.
13
                                     %Surface resolution
_{14} dx = L/N;
15
_{16} x = linspace(-L/2, L/2, N);
_{17} g = L/7; %Tapering Parameter
18 %
```

```
% Planck distribution parameters

h = 6.626e-34; % Planck's Constant = 4.135 x 10<sup>-15</sup> eV s

c = 3e8; % speed of light

T = 90; % Temperature of the hot source

k = 1.38066e-23; % Boltzmann constant in J/K
```

```
lambda_min = 180e-6; \% Starting wavelenght
27
  lambda_max = 210e-6;% Ending wavelenght
28
29
  waves = linspace(lambda_min, lambda_max, N_w); %Wavelength resolution.
30
      Can simulate the detector's frequency range
31
  p = 2*pi*h*(c^2)./(waves.^5);
                                                      %Numerator
32
  bT = p./(exp(h*c./(waves*k*T)-1));
                                                      %Planck Radiation
33
      Distribution
34
  bT = bT./max(bT);
                                                      %Normalized Planck
35
      Radiation Distribution
36
  %
37
  %Scalar wave parameters
38
39
                                               %Incident angle in degrees (
  theta_deg = 45;
40
      because we're all used to them)
   theta_rad = theta_deg * pi / 180;
                                            %Incident angle in radians,
41
      the magic stuff
42
  %
^{43}
  %Surface control parameters
44
^{45}
   cl = 1.08E - 04;
                      %Correlation length in m. This parameter is
46
      obtained from the profile analysis code
   height = 1.53E-04; %Height std in m. This parameter is obtained from
47
      the profile analysis code
  laplacian_height = 2.05890e - 07; % Empirical guess
48
  %
49
50
   height_normalized = (height./waves)./scale; %Normalize
51
   cl_normalized = (cl_vaves)./scale;
                                                  %Normalize
52
   waves_normalized = (waves./waves)./scale;
                                                  %Normalize
53
   laplacian_height_normalized = (laplacian_height./waves)/scale; %
54
      Normalize
55
  k_number=2*pi./waves_normalized;
                                               %Wavenumber!
56
57
  %
58
```

59 %Image printing settings

```
60
61 format = '-dpng';
62 resolution = '-r500';
63 directory_output = uigetdir('', 'Choose directory with output data
64
65 %
```

```
%Initialized
66
   new\_light = zeros(N_f, 1);
67
   final_light = zeros(N_f, 1);
68
   final_spm = zeros(N_f, 1);
69
   singularity\_check\_G = zeros(N\_mc,1);
70
   incident_e_temp = zeros(N_w, 1);
71
   time = zeros(N_w, 1);
72
   final_conditionality = zeros(N_w, 1);
73
74
   step = pi/(N_f-1);
75
   degrees = -(pi/2): step:(pi/2);
76
   degs = degrees.*180/pi;
77
78
  %This allows to plot only half of the scattered field,
79
  % in order to make it easier to compare with the experimental setup
80
      which can only see half of the scattered field
   half = (N_f/2) : N_f;
81
   gamma=1.78107241;
82
83
84
  %
85
   tic
86
   for j = 1:N_w
87
88
       text = 'Iteration number - %1.0f. Lenght - %1.0f (multiple of
89
           wavelenght). Surface points -\%1.0 f.\n';
       fprintf(text,j,L,N)
90
91
       scaling = 8*pi*k_number(j)*g*sqrt(pi/2)*cos(theta_rad)*(1-(1+2*tan(j)))
92
           theta_rad)^2)/(2*(k_number(j)*g*cos(theta_rad))^2)); %
           Conservation of energy scale
93
       for i=1:N_mc
94
95
           %
96
```

97

%Here the surface is generated, following the above

98	parameters. %Idea taken from the thesis of David Bergstrm and modified
99	to %include a Laplacian noise component
100 101 102 103	$ u = rand(1,N) - 0.5; b = laplacian_height_normalized(j)/sqrt(2); laplace_noise = 0 - b*sign(u).*log(1-2*abs(u)); $
104 105	%
106 107 108 109 110	<pre>Filter = exp(-abs(x)/(cl_normalized(j)/2)); Z = height_normalized(j)*randn(1,N); %Generate random numbers with standard deviation (height) F = sqrt(2/sqrt(pi))*sqrt(dx/cl_normalized(j))*ifft(fft(Z).* fft(Filter)) + laplace_noise; %Empirical correction</pre>
111	%
112 113 114 115 116	%The first derivative is being calculated dF = (1/dx)*gradient(F); $dF(1) = (F(2)-F(N))/(2*dx);$ %They connect around the edges in order to avoid discontinuities around the edges of the surface which cause false reflections dF(N) = (F(1)-F(N-1))/(2*dx);
117 118	%
119	%Tapered incident scalar beam is generated here, following the ideas of
120	% Tsang, et al.
121 122	$w = (2*(((x+F*tan(theta_rad))/g).^2)-1)./((k_number(j)*g*cos(theta_rad))^2))$
123	<pre>Psi_inc = bT(j)*exp(1i*k_number(j)*(x*sin(theta_rad)-F*cos( theta_rad)).*(1+w)-((x+F*tan(theta_rad))/g).^2)'; %Every wavelenght has a certain energy</pre>
124	
125	%
126 127	%Perform the point matching and computing of the argument
128	[xm, xn] = meshgrid(x);
129	[Fm, Fn] = meshgrid(F);
-----	---
130	
131	distance=k_number(j) * sqrt((Fm-Fn). $^2$ + (xm-xn). $^2$ );
132	distance $(1:N+1:N 2) = NaN;$ % Replace the diagonal of this
	matrix with anything but 0 to avoid 0s in the denominator
	of the Green's function.
133	0
134	70
135	%2D Green's function defined and used here
136	
137	G=1i/4*besselh(0,1,distance)*dx;
138	$taylor = sqrt(1+dF.^2) * dx;$
139	
140	$G(1:N+1:N^2) = (1i/4) * dx * (1+(2*1i/pi)) * (\log(gamma*k_number(j))*)$
	taylor/4)-1); % The badly conditioned diagonal is
	replaced by the approximation
141	
142	$singularity_check_G(i) = cond(G,2); \%$ The conditionality of
	the Green's function is recorded for later analysis.
143	
144	scattered_vector=G\Psi_inc;
145	%
	<sup>o</sup> Sum up all the scattered source points at every field point
146	70 Sum up all the scattered source points at every field point
147	for $m-1 \cdot N$ f
148	
149	Psi scat = $sum(exp(-1i*k number(i)*(sin(degrees(m))*x+F*))$
150	$\cos(\deg \operatorname{rees}(m)))$ * scattered vector '*dx): %Compute the
	numeric integral of the scattered field
151	normalized = $(abs(Psi scat))^2/scaling$
101	Make real and normalize
152	new light (m) = normalized / i + $((i-1)/i)$ * new light (m):
	%Iterative mean for the Monte Carlo
153	
154	$\operatorname{end}$
155	end
156	
157	final_conditionality(j) = mean(singularity_check_G);
158	final_light = final_light + new_light;
159	time(j) = toc;
160	
161	end
162	
163	$final_light = 10*log10(final_light);$
164	

165 %

```
%Plot the Planck distribution
166
167
   figure('name', 'Planck Radiation Distribution');
168
   plot (linspace (lambda_min, lambda_max, N_w), bT, '*');
169
   xlabel('Wavelength [m]')
170
   ylabel ('Normalized Spectral Radiance (originally [Wm^{\{-2\}} sr^{\{-1\}}] nm
171
        (-1]);
    title('Planck Radiation Distribution');
172
   grid on
173
   hold on;
174
   legend(strcat(num2str(T), 'K')); %See if this works
175
   print (format, resolution, strcat (directory_output, '\Planck Radiation
176
       Distribution '))
177
   %
178
       %
   %Plot the last surface profile
179
180
   figure('name', 'Last Simulated Surface');
181
   plot(x,F);
182
   xlabel('Lenght [multiple of wavelenght]')
183
   ylabel('Height [multiple of wavelenght]');
184
   title('Last Simulated Surface');
185
   grid on
186
   print (format, resolution, streat (directory_output, '\Last Simulated
187
       Surface '))
188
   %
189
       %=
   %Plot the final incident beam
190
191
   figure ('name', 'Last Simulated Incident Field Distribution');
192
   plot(x, abs(Psi_inc));
193
   xlabel('Lenght [multiple of wavelenght]')
194
   ylabel ('Normalized Intensity (originally [Wm^{-2}] sr^{-1} mm^{-1}])'
195
       );
    title ('Last Simulated Incident Field Distribution');
196
   grid on
197
   print (format, resolution, streat (directory_output, '\Last Simulated
198
       Incident Field Distribution '))
199
   %
200
```

```
%Plot the sum of all the scattered waves summed together as one band
201
202
   figure ('name', 'Scattered Band in 2D Spherical Coordinates');
203
   plot(degs,final_light, degs(half), final_light(half),'r');
204
   xlabel('Angle [deg]')
205
   ylabel('Bistatic Scattering Coefficient (dB)');
206
   title('Scattered Band in 2D Spherical Coordinates');
207
   grid on
208
   print (format, resolution, streat (directory_output, '\Scattered Band
209
       in 2D Spherical Coordinates'))
210
   %
211
   %Plot the conditionality of the Green's function to determine whether
212
        the
   %simulation is reliable or not. Higher values indicate a badly
213
       behaved
   %matrix
214
215
   figure ('name', 'Conditionality of Green's function matrix');
216
   plot (1:N_w, final_conditionality, '*');
217
   xlabel('Number of Iterations')
218
   ylabel('2-norm Condition Number');
219
   title('Conditionality of Green's function matrix');
220
   grid on
221
   print(format, resolution, strcat(directory_output, '\setminus Conditionality)
222
       of Green's function discretization'))
223
   %
224
   %Plot the time series
225
226
   figure('name', 'Time Iterations');
227
   plot (1:N_w, time, '*');
228
   xlabel('Iteration Number')
ylabel('Time per Iteration [s]');
229
230
   title('Time Iterations');
231
   grid on
232
   print (format, resolution, strcat (directory_output, '\Time Iterations'
233
       ))
^{234}
   %
235
```

236

237 %The simulated surface in 2D spherical coordinates is saved to an Excel

```
%document. Could be changed to anything else
238
239
   degrees = num2cell((degs))';
240
   degrees = ['Degrees'; degrees];
241
242
    values = num2cell(final_light);
243
    values = ['MoM'; values];
244
245
   data = [degrees, values];
246
    xlswrite(strcat(directory_output, '\Scattered_field.xlsx'), data, 1, 'A1
247
        ');
^{248}
   %
249
```

```
<sup>250</sup> % Save the simulation parameters to a text file in the same folder as the
<sup>251</sup> % simulation images.
<sup>252</sup>
<sup>253</sup> headers = ['Scale: %1.1f\n', 'Number of Surface Points: %1.0f\n', '
```

```
Surface Lenght: \%1.5e [*waves]\n', 'Surface Resolution: \%1.3f [wavelenghts per point] \n\n', 'Number of Wavelenghts: \%1.0f\n',
         Number of Field Points: \%1.0\,f\,\mbox{\sc n} , 'Number of Monte Carlo
         simulations: \%1.0 f \setminus n \setminus n', 'Surface Standard Deviation: \%1.5e [m] \setminus n'
           'Surface Correlation Lenght: %1.5e [m]\n', 'Laplace diversity:
         1.5e [m] n^{, ', 'Hot Source Temperature: <math>1.1f [K] n^{, ', 'Starting}
         Wavelenght: %1.5e [m] \ n', 'Final Wavelenght: %1.5e [m] \ n',
    Incident Angle: %1.3f [Degrees]\n', 'Total Time: %1.3f [s]\n'];
vals = [scale, N, L, dx, N_w, N_f, N_mc, height, cl, laplacian_height
254
         , T, lambda_min, lambda_max, theta_deg, toc];
255
    fid = fopen(streat(directory_output, '\Simulation Parameters.txt'),'
256
         wt ');
    fprintf(fid , headers , vals);
257
    fclose(fid);
258
259
260
```

261 %

```
109
```

#### Appendix E

# SPM Band Scatter Simulation Code

```
1 clear all
2 close all
3
  N_w = 50;
                                     %Number of wavelenghts. Minimum is 2.
4
       Bigger is better.
                                     %Number of scattered points. Bigger
  N_{-}f = 180;
5
      is better
6
  scale = 1;
\overline{7}
  %
8
9 %Scalar wave parameters
10
                                               %Incident angle in degrees (
  theta_deg = 45;
11
      because we're all used to them)
                                             %Incident angle in radians,
   theta_rad = theta_deg * pi / 180;
12
      the magic stuff
13
  %
14
  %Surface control parameters
15
16
                          %Correlation length in m. This parameter is
   cl = 1.08000e - 04;
17
      obtained from the profile analysis code
   height = 1.93000e-05; %Height std in m. This parameter is obtained
18
      from the profile analysis code
^{19}
20 %
```

```
%Planck distribution parameters
21
22
                             \% Planck's Constant = 4.135 x 10^-15 eV s
  h = 6.626 e - 34;
^{23}
   c = 3e8;
                        % speed of light
^{24}
                    % Temperature of the hot source
25
  T = 850;
   k = 1.38066 e - 23;
                        % Boltzmann constant in J/K
26
27
   lambda_min = 180e-6; \% Starting wavelenght
^{28}
   lambda_max = 210e-6;% Ending wavelenght
29
30
   waves = linspace(lambda_min, lambda_max, N_w); %Wavelength resolution.
31
      Can simulate the detector's frequency range
32
  p = 2*pi*h*(c^2)./(waves.^5);
                                                       %Numerator
33
  bT = p./(\exp(h*c./(waves*k*T)-1));
                                                       %Planck Radiation
34
      Distribution
35
  bT = bT./max(bT);
                                                       %Normalized Planck
36
      Radiation Distribution
37
  % Here we must apply the window
38
39
  %
40
  %Normalize everything
41
^{42}
   height_normalized = height./waves; %Normalize
^{43}
   cl_normalized = cl_vaves;
                                         %Normalize
44
   waves_normalized = waves./waves;
                                         %Normalize
45
46
   k_number=2*pi./waves_normalized;
                                                %Wavenumber!
47
^{48}
  %
49
  %Scattered field initialization
50
51
   spm_final = zeros(1, N_f);
52
  dan = pi/(N_f+1);
53
  degrees = -(pi/2-dan):dan:(pi/2-dan);
54
   degs = degrees.*180/pi;
55
   half = (N_{-}f/2) : N_{-}f;
56
57
```

```
58 %
```

59 %Image printing settings

```
60
  format = '-djpeg';
61
   resolution = '-r300';
62
   directory_output = uigetdir('', 'Choose directory with output data
63
      files:');
64
  %
65
66
   for i = 1:N_w
67
68
  spm_new = 4*k_number(i)^3*(cos(degrees).^2).*cos(theta_rad)*(
69
      height_normalized (i) 2* cl_normalized (i) /(2* sqrt (pi))).* exp(-(((sin
      (degrees)-sin(theta_rad)).^2)*k_number(i)^2*cl_normalized(i)
       2*0.25));
   spm_final = ((i-1)/i) . * spm_final + spm_new./i;
70
71
  end
72
73
   spm_final = 10 * log 10 (spm_final); \% Convert to dB
74
75
   figure('name', 'Small Perturbation Method');
76
   plot(degs, spm_final);
77
   xlabel('Angle [degrees]')
78
   ylabel('Bistatic Scattering Coefficient [dB]');
79
   title('Small Perturbation Method for Diffuse Scattering');
80
   grid on
81
   hold on;
82
   print (format, resolution, streat (directory_output, '\Small Perturbation
83
       Method for Diffuse Scattering'))
84
  %
85
  %Plot the Planck distribution
86
87
   figure('name', 'Planck Radiation Distribution');
88
   plot (linspace (lambda_min, lambda_max, N_w), bT, 'r*');
89
   xlabel('Wavelength [m]')
90
   ylabel('Normalized Spectral Radiance (originally [W m^{-2}] sr^{-1} nm
91
       (-1]);
```

```
92 title ('Planck Radiation Distribution');
```

```
93 grid on
```

```
94 hold on;
```

```
{}_{95} \ legend(strcat(num2str(T), 'K')); \ \% See \ if \ this \ works
```

```
96 print(format, resolution, strcat(directory_output, '\Planck Radiation
Distribution'))
```

```
97
```

```
%The simulated surface in 2D spherical coordinates is saved to an
99
       Excel
   %document. Could be changed to anything else
100
101
   degrees = num2cell((degs)');
102
   degrees = ['Degrees'; degrees];
103
104
   res = num2cell(spm_final');
105
   res = ['SPM'; res];
106
107
   data = [degrees, res];
108
   xlswrite(strcat(directory_output, '\Scattered_field_SPM.xlsx'),data
109
       ,1, 'A1');
110
   %
111
```

112

```
headers = ['Number of Wavelenghts: %1.0f\n', 'Number of Field Points:
     %1.0f\n', 'Ratio Scale: %1.0f \n', 'Surface Standard Deviation:
113
        1.5e [m] n', 'Surface Correlation Lenght: %1.5e [m] n', 'Hot
        Source Temperature: %1.1f [K]\n', 'Starting Wavelenght: %1.5e [m]\n
        ', 'Final Wavelenght: %1.5e \mbox{[m]}\n ', 'Incident Angle: %1.3f [Degrees
        ]\n'];
   vals = [N_w, N_f, scale, height, cl, T, lambda_min, lambda_max,
114
        theta_deg];
115
   fid = fopen(streat(directory_output, '\Simulation Parameters.txt'),'
116
        wt ');
    fprintf(fid , headers , vals);
117
   fclose(fid);
118
```

#### Appendix F

### Neural Network Code

```
<sup>1</sup> %This simple code creates a feedforward neural network with 4 input
  %neurons, 3 hidden layer neurons, and 2 output layer neurons. It
2
      accepts an
  %excel file with all the necessary data and then estimates the
3
      network,
  % it 's performance, and runs a quick test with unseen data. The size
4
      and
  %shape of the network can change
5
6
   clear all
7
   close all
8
9
  \%~F\_8 = 2460; %Mean size in meters. Just to be known
10
  \% F_{-12} = 1765;
11
  \% F_{-16} = 1230;
12
13
   directory_input = uigetdir('', 'Choose directory with input data
14
       files:');
                              % Select input directory
   Data = xlsread(strcat(directory_input, '\Taguchi NN.xlsx'), 'Neural
15
      Network', 'B3:G18'); %Choose your own files
   inputs = Data(:, 1:4)';
16
   targets = Data(:,5:6)';
17
18
19
  % % Designing the network
20
  % %=
21
   net = fitnet([7]);
22
23
   net.trainFcn = 'traincgp';
^{24}
25
   net.divideParam.trainRatio = 11/16;
26
   net.divideParam.testRatio = 5/16;
27
   net.divideParam.valRatio = 5/16;
^{28}
^{29}
```

```
[net,tr] = train(net,inputs,targets);
30
^{31}
   outputs = net(inputs);
32
   errors = gsubtract(targets, outputs);
33
   performance = perform(net, targets, outputs);
^{34}
35
   test_data = xlsread(strcat(directory_input, '\Taguchi NN.xlsx'), 'NN
36
      Test', 'B3:G7'); %Choose your own files
37
  Z = test_data(:, 1:4)';
38
   real_data = test_data(:,5:6)'.*1e5
39
40
   predicted = sim(net, Z) . *1e5
^{41}
```

## Bibliography

- Abd-Elhady, M. S., Rindt, C. C. M., Wijers, J. G., and van Steenhoven, A. A. (2006). Modelling the impaction of a micron particle with a powdery layer. *Powder Technology*, 168:111–124.
- Abrasives, F. (2013). F-grit sizes bonded. Accessed on 10/04/2014 from http://www.fepa-abrasives.org/Abrasiveproducts/Grains/Fgritsizesbonded.aspx.
- Aguiar, P. R., Cruz, C. E. D., Paula, W. C. F., and Bianchi, E. C. (2008). Predicting Surface Roughness in Grinding using Neural Networks, chapter 3, pages 33–44. Advances in Robotics, Automation and Control. InTech.
- Ali, S. M. and Dhar, N. R. (2010). Tool wear and surface roughness prediction using an artificial neural network (ann) in turning steel under minimum quantity lubrication (mql). World Academy of Science, Engineering and Technology, 38:830–839.
- Arokiadass, R., Palaniradja, K., and Alagumoorthi, N. (2011). Predictive modeling of surface roughness in end milling of al/sicp metal matrix composite. *Scholars Research Library*, 3(2):228–236.
- Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159.
- Bell, T. (1996). Basic source separataion code. Accessed on 10/05/2014 via http://cnl.salk.edu/~tewon/ica\_cnl.html.
- Benardos, P. G. and Vosniakos, G. C. (2003). Predicting surface roughness in machining: a review. *International Journal of Machine Tools & Manufacture*, 43:833–844.
- Bergstrom, D. (2008). The Absorption of Laser Light by Rough Metal Surfaces. PhD thesis, Lulea University of Technology, Department of Applied Physics and Mechanical Engineering.
- Burghignolin, P., Nallo, C. D., Frezza, F., and Galli, A. (2002). A simple nystrom approach for the analysis of 3d arbitrarily shaped conducting and dielectric bodies. *INTERNATIONAL JOURNAL OF NUMERICAL MODELLING: ELECTRONIC NETWORKS, DEVICES AND FIELDS*, 16:179–194.
- C., A. S., Lopez, M. V., Sabre, M., and Gomes, L. (1998). Implications of a sandblasting model for dust production by wind erosion in arid areas. Technical report, Centre National de la Recherche Scientifique (CNRS).
- Choi, N. and Harvey, J. E. (2013). Numerical validation of the generalized harveyshack surface scatter theory. *Optical Engineering*, 52(11).

- Ciarlet, P. and Zou, J. (1999). Fully discrete ?nite element approaches for time-dependent maxwells equations. *Numerische Mathematik Electronic Edition*, 82:193219.
- Colak, O., Kurbanoglu, C., and Kayacan, M. C. (2007). Milling surface roughness prediction using evolutionary programming methods. *Materials & Design*, 28(2):657–666.
- Crowther, B. G. (1996). Computer modeling of integrating spheres. *Applied Optics*, 35(30):5880–5886.
- Dang, V. H., Perret-Liaudet, J., Scheibert, J., and Bot, A. L. (2013). Direct numerical simulation of the dynamics of sliding rough surfaces. *Comput Mech*, pages 1170–1183.
- Demuth, H. and Beale, M. (2002). Neural Network Toolbox. MathWorks, 4 edition.
- Doja, B. and Singh, D. D. K. (2012). Analysis and effect of process parameters on surface roughness and tool flank wear in facing operation. *International Journal of Engineering Research & Technology*, 1(4):1–7.
- Du, Y. and Liu, B. (2009). A numerical method for electromagnetic scattering from dielectric rough surfaces based on the stochastic second degree method. *Progress In Electromagnetics Research*, 97:327–342.
- Dubey, A. and John, F. (2013). Finite element modelling of surface stresses in coatings under single particle impact. International Journal of Engineering Research & Technology, 2(6):3421– 3426.
- Durbin, J. and Koopman, S. J. (2012). Time Series Analysis by State Space Models. Oxford Statistical Science Series. Oxford University Press, 2 edition.
- ESA (2014). Spica a space infrared telescope for cosmology and astrophysics. Accessed on 15/04/2014 from http://sci.esa.int/cosmic-vision/53635-spica.
- Esme, U., Sagbas, A., and Kahraman, F. (2009). Prediction of surface roughness in wire electrical discharge machining using design of experiments and neural networks. *Iranian Journal of Science and Technology*, 33(3):231–240.
- Evans, P. L., Schwartz, L. W., and Roy, R. V. (2000). A mathematical model for crater defect formation in a drying paint layer. *Journal of Colloid and Interface Science*, 227:191–205.
- Ferlet, M., Geis, N., Goicoechea, J. R., Griffin, D. K., Heras, A. M., Isaak, K., Nakagawa, T., Onaka, T., Rando, N., Swinyard, B., Takahashi, N., and Vives, S. (2009). Spica revealing the origins of planets and galaxies. Technical report, ESA.
- Ferrari, L. and Panman, J. (2013). Al sample sandblasting results. Internal report.
- Finger, R. and Kerr, A. R. (2008). Microwave loss reduction in cryogenically cooled conductors. International Journal of Infrared and Millimeter Waves, 29(10):924–932.
- Foldyna, J., Klich, J., Hlavacek, P., Zelenak, M., and Scucka, J. (2013). Erosion of metals by pulsating water jet. *Technicki vjesnik 19*, pages 381–386.
- Fraley, S., Oom, M., Terrien, B., and Zalewski, J. (2007). Design of experiments via taguchi methods: orthogonal arrays. Accessed on 12/04/2014.
- Freedman, D., Pisani, R., and Purves, R. (2007). *Statistics.* W. W. Norton & Company, 4 edition.

- Garg, R. (2008). Analytical and Computational Methods in Electromagnetics. Artech House Electromagnetic Analysis Series. Artech House.
- Gascon, F. and Salazar, F. (2011). Simulation of Rough Surfaces and Analysis of Roughness by MATLAB, chapter 19, pages 391–421. InTech.
- Gaspard, P. (2002). The correlation time of mesoscopic chemical clocks. Journal of Chemical Physics, 117:8905–8916.
- Germer, T. A. (2001). Polarized light scattering by microroughness and small defects in dielectric layers. Journal of the Optical Society of America A, 18(6):1279–1288.
- Goicoechea, J. R., Isaak, K., and Swinyard, B. (2009). Exoplanet research with safari: A far-ir imaging spectrometer for spica. *The Astronomical Society of the Pacific*.
- Gordon, G. and Heyman, E. (2004). Rough surface scattering using the gaussian beam summation method. Technical report, School of Electrical Engineering, Tel Aviv University.
- Hamilton, L. R., Ottusch, J. J., Stalzer, M. A., Turley, R. S., Visher, J. L., and Wandzura, S. M. (1999). Numerical solution of 2-d scattering problems using high-order methods. *IEEE Transactions on Antennas and Propagation*, 47(4):683–691.
- Han, L., Okamoto, A., Fukushima, M., and Okiji, T. (2008). Evaluation of flowable resin composite ssurface eroded by aciding and alcoholic drinks. *Dental Materials Journal*, 27:455–465.
- Hangartner, B. (2002). A 2d electromagnetic scattering solver for matlab using method of moments and green's tensor technique. Technical report, ETH Zurich.
- Harvey, J. E. and Shack, R. V. (1978). Aberrations of diffracted wave fields. Applied Optics, 17(18):3003–3009.
- Instruments, O. (2014). Infrared Elements. Newport.
- Jandhyala, V., Michielssen, E., Balasubramaniam, S., and Chew, W. C. (1998). A combined steepest descent-fast multipole algorithm for the fast analysis of three-dimensional scattering by rough surfaces. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, 36(3):738–748.
- Japan Association of Remote Sensing (1996). 3.4 surface scattering. Accessed on 29/04/2014 via http://wtlab.iis.u-tokyo.ac.jp/~wataru/lecture/rsgis/rsnote/cp3/cpe-4.htm.
- JAXA (2003). Japan aerospace exploration agency. Accessed on via http://global.jaxa.jp.
- Johnson, K. L. (1987). Contact Mechanics. Cambridge University Press.
- Kamely, M. A., Kamil, S. M., and Chong, C. W. (2011). Mathematical modeling of surface roughness in surface grinding operation. *International Journal of Engineering and Natural Sciences*, pages 146–149.
- Kay, S. (1999). Modern Spectral Estimation: Theory and Application. Prentice Hall, 1 edition.
- Khorasanizadeh, S. (2010). The effects of shot and grit blasting process parameters on steel pipes coating adhesion. World Academy of Science, Engineering and Technology, 42:1290–1298.

- Klandermans, P. (2013). Beam pattern measurements of baffe and integrating sphere for the development of the safari aiv calibration source. Master's thesis, University of Groninge, Faculty of Mathematics and Natural Sciences.
- Kleinedlerova, I., Junac, A., and Kleinedler, P. (2013). Analysis evaluation parameters of ra, rz surface roughness on short sectors.
- Klinkov, K. V. (2005). Unsteady gas flows and particle dynamics in the shock layer formed by the impingement of a supersonic two-phase jet onto a plate. PhD thesis, Georg-August-Universitat.
- Knockaert, L. (1991). A general gauss theorem for evaluating singular integrals over polyhedral domains. *Electromagnetics*, 11:269–280.
- Koopman, S. J., Shephard, N., and Doornik, J. A. (1999). Statistical aalgorithm for models in state space using ssfpack 2.2. *Econometrics Journal*, 2(1):113–166.
- Krywonos, A., Harvey, J. E., and Choi, N. (2006). Linear systems formulation of scattering theory for rough surfaces with arbitrary incident and scattering angles. PhD thesis, University of Central Florida, College of Optics and Photoics.

Laboratories, I. (2013).

- Langlois, D., Chartier, S., and Gosselin, D. (2010). An introduction to independent component analysis: Infomax and fastica algorithms. *Tutorials in Quantitative Methods in Psychology*, 6(1):31–38.
- Lee, S.-W., Boersma, J., Law, C.-L., and Deschamps, G. A. (1980). Singularity in green's function and its numerical evaluation. *IEEE Transactions on Antennas and Propagation*, 28(3):311–317.
- Letnes, P. A., Maradudin, A. A., Nordam, T., and Simonsen, I. (2012). Calculation of all elements of the mueller matrix for scattering of light from a rough two-dimensional randomly rough metal surface. *Physical Review A*, 86(3):31803–31808.
- Levkin, B. F., Vecchio, K. S., DuPont, J. N., and Marder, A. R. (1999). Modelling solid-particle erosion of ductile alloys. *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science*, 30A:1763–1774.
- Maradudin, A. A. (2007). Light Scattering and Nanoscale Surface Roughness. 1571-5744. Springer US, 1 edition.
- Mehrotra, K., Mohan, C., and Ranka, S. (1996). Elements of Artificial Neural Networks (Complex Adaptive Systems). A Bradford Book.
- Mie, G. (1908). Beitrage zur optik truber medien, speziell kolloidaler metallosungen. Annalen der Physik, 330(3):377–445.
- Mischenko, M. L., Hovenier, J. W., and Travis, L. D. (1999). Light Scattering by Nonspherical Particles: Theory, Measurements, and Applications. Academic Press, 1 edition.
- Miyoshi, K., Sutter, J. K., Horan, R. A., Naik, S. K., and Cupp, R. J. (2004). Assessment of erosion resistance of coated polymer matrix composites for propulsion applications. NASA/TM-2004-212911.

- Nachamkin, J. (1990). Integrating the dyadic green's function near sources. *IEEE Transactions on Antennas and Propagation*, 38(6):919–921.
- Naik, G. R. and Kumar, D. K. (2011). An overview of independent component analysis and its applications. *Informatica*, 35:63–81.
- Nakajima, T. Y., Nakajima, T., Yoshimori, K., Mishra, S. K., and Tripathi, S. N. (2009). Development of a light scattering solver applicable to particles of arbitrary shape on the basis of the surface-integral equations method of muller type. i. methodology, accuracy of calculation, and electromagnetic current on the particle surface. APPLIED OPTICS, 48(19):3526–3536.
- Nanotech (2009). Dektak profilometer. Accessed on 04/04/2014 via http://www.nanotech. ucsb.edu/index.php/nanofabrication-facility/equipment-list?id=116.
- Nasser, M. M. (2013). Fast solution of boundary integral equations with the generalized neumann kernel. Technical report, Cornell University Library.
- Negrea, A. and Predoi, M. V. (2012). The elastic contact of a sphere with an elastic halfspace, a comparison between analytical and finite elements solutions. University Politehnica of Bucharest Scientific Bulletin, 74(4):69–78.
- Nordam, T., Letnes, P. A., and Simonsen, I. (2013). Numerical simulations of scattering of light from two-dimensional surfaces using the reduced rayleigh equation. *Frontiers in Physics*, 1(8).
- Normanshire, C. (2012). ZEMAX 12 SRON training course. Radiant Zemax LLC.
- Online, P. (2014). Biorad: Ft-irs for spectroscopic research. Accessed on 20/04/2014 via http: //www.pharmaceuticalonline.com/doc/ft-irs-for-spectroscopic-research-0001.
- Oren, M. and Nayar, S. K. (1996). Generalization of the lambertian model and implications for machine vision. *International Journal of Computer Vision*, 14(3).
- Ottusch, J. J., Valley, G. C., and Wandzura, S. (1998). Integral equations and discretizations for waveguide apertures. *IEEE Transactions on Antennas and Propagation*, 46(11):1725–1738.
- Ozel, T. and Karpat, Y. (2004). Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks. *International Journal of Machine Tools & Manufacture*, 45:467–479.
- Patel, S. P. (2011). Effects of alumina slurry jet on polishing and erosion of aluminum substrate: Experimentation and cdf modelling. Master's thesis, Northeastern University Department of Mechanical and Industrial Engineering.
- Pinel, N., Bourlier, C., and Saillard, J. (2010). Degree of roughness of rough layers: Extensions of the rayleigh roughness criterion and some applications. *Progress in Electromagnetics Research* B, 19:41–63.
- Rao, P. and Buckley, D. H. (1984). Empirical relations for cavitation and liquid impingement erosion processes. Technical Report 1984023279, NASA.
- Rashid, M. A. and Lani, M. A. (2010). Surface roughness prediction for cnc milling process using artificial neural network. *World Congress on Engineering*, 3.
- Rice, S. O. (1951). Reflection of electromagnetic waves from slightly routh surfaces. Communications on Pure and Applied Mathematics, 4(3):351–378.

Ripken, J. F. (1969). Comparative studies of drop impingement erosion and cavitation erosion. Technical report, University of Minesota St. Anthony Falls Hydraulic Laboratory.

Sadiku, M. N. O. (2009). Monte Carlo Methods for Electromagnetics. CRC Press, 1 edition.

- Sanchez-Avila, C. and Sanchez-Reillo, R. (2002). A wavelet-based method for solving discrete first-kind fredholm equations with piecewise constant solutions. *INTERNATIONAL JOUR-NAL FOR NUMERICAL METHODS IN ENGINEERING*, 57:577–598.
- Schroder, S., Duparre, A., Coriand, L., Tunnermann, A., Penalver, D. H., and Harvey, J. E. (2011). Modelling of light scattering in different regimes of surface roughness. *Optics Express*, 19(10):9820–9835.
- Schuerman, D. W. (1980). Light Scattering by Irregularly Shaped Particles. Springer.
- Sidorchuk, A., Smith, A., and Nikora, V. (2004). Probability distribution function approach in stochastic modelling of soil erosion. Sediment Transfer through the Fluival System, 288:345– 353.
- Slatineanu, L., Potarniche, S., Coteata, M., Grigoras, I., Gherman, L., and Negoescu, F. (2011). Surface roughness at aluminium parts sand blasting. *Proceedings in Manufacturing Systems*, 6(2):70–74.
- Soto-Crespo, J. M. and Nieto-Vesperinas, M. (1988). Electromagnetic scattering from very rough random surfaces and deep reflection gratings. *Optical Society of America*, pages 367–384.
- Sun, Y. (2006). Analytic framework for calculating brdfs of randomly rough surfaces. Computer Science Technical Reports 1653, Purdue University.
- Suresh, P., Rao, P. V., and Deshmukh, S. G. (2002). A genetic algorithmic approach for optimization of surface roughness prediction model. *International Journal of Machine Tools & Manufacture*, 42:675–680.
- Tavares, S. M. O. (2005). Analysis of surface roughness and models of mechanical contacts. Master's thesis, Faculdade de Engenharia da Universidade do Porto.
- Tian, Z. F., Inthavong, K., Tu, J. Y., and Yeoh, G. (2007). Numerical simulation of gas-particle flows over an in-line tube bank. ANZIAM Journal, 48:509–526.
- Ticconi, F., Pulvirenti, L., and Pierdicca, N. (2011). Models for Scattering from Rough Surfaces, chapter 10, pages 204–226. InTech.
- Torrance, K. E. and Sparrow, E. M. (1967). Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114.
- Tsang, L., Kong, J. A., Ding, K.-H., and Ao, C. O. (2001). Numerical Simulations, volume 2 of Scattering of Electromagnetic Waves. John Wiley & Sons, Inc.
- Vigolo, D., Grif?ths, M., Radl, S., and Stone, H. A. (2013). An experimental and theoretical investigation of particlewall impacts in a t-junction. *Journal of Fluid Mechanics*, 727:236 – 255.
- Wauer, J. and Rother, T. (2009). Considerations to rayleigh's hypothesis. Optics Communications, 282(3):339–350.

Wilde, I. (2010). Neural networks. Loose leaf, Kings College London Mathematics Department.

- Young, C. W. and Laboratories, S. (1967). The Development of Empirical Equations for Predicting Depth of an Earth-penetrating Projectile. Development report. Sandia Laboratories.
- Zhenrong, Z., Jing, Z., and Peifu, G. (2010). Roughness characterization of well-polished surfaces by measurements of light scattering distribution. *Optica Applicata*, 40(4):812–818.
- Zhou, J., Ai, N., Wang, L., Zheng, H., Luo, C., Jiang, Z., Yu, S., Cao, Y., and Wang, J. (2011). Roughening the white oled substrates surface through sandblasting to improve the external quantum efficiency. *Organic Electronics*, 12:68–653.