Optical Extreme Learning Machines: a new trend in optical computing

Duarte Silva
Mestrado em Engenharia Física
Departamento de Física e Astronomia
2022

Orientador
Prof. Dr. Ariel Guerreiro, Faculdade de Ciências da Universidade do Porto

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____

UNIVERSIDADE DO PORTO

MASTERS THESIS

Optical Extreme Learning Machines: a new trend in optical computing

Author:
Duarte SILVA

Supervisor:
Ariel GUERREIRO

A thesis submitted in fulfilment of the requirements
for the degree of MSc. Engineering Physics

at the

Faculdade de Ciências da Universidade do Porto
Departamento de Física e Astronomia

November 2, 2022

" It's better to try and fail, than to not try at all. "

Hélio Lucas, canoeing coach. Said to me over 6 years ago at the end of a race.

Declaração de Honra

Eu, Duarte José Fernandes da Silva, inscrito no Mestrado em Engenharia Física da Faculdade de Ciências da Universidade do Porto declaro, nos termos do disposto na alínea a) do artigo 14.º do Código Ético de Conduta Académica da U.Porto, que o conteúdo da presente dissertação reflete as perspetivas, o trabalho de investigação e as minhas interpretações no momento da sua entrega.

Ao entregar esta dissertação, declaro, ainda, que a mesma é resultado do meu próprio trabalho de investigação e contém contributos que não foram utilizados previamente noutros trabalhos apresentados a esta ou outra instituição.

Mais declaro que todas as referências a outros autores respeitam escrupulosamente as regras da atribuição, encontrando-se devidamente citadas no corpo do texto e identificadas na secção de referências bibliográficas. Não são divulgados na presente dissertação quaisquer conteúdos cuja reprodução esteja vedada por direitos de autor.

Tenho consciência de que a prática de plágio e auto-plágio constitui um ilícito académico.

Duarte José Fernandes da Silva

Porto, 30 de Setembro de 2022

Acknowledgements

As with most achievements in life, this work could not have been accomplished alone. There are many people to thank for, and I apologize upfront if I forget to mention anyone, but know that every experience I've had thus far has contributed to who I am, and consequently to this work.

First and foremost, I want to thank my supervisor, Ariel Guerreiro, as well as Nuno Azevedo Silva and Tiago Ferreira, for giving me the opportunity to develop this work in a fantastic environment, full of fruitful discussions and ambition to do and be better. I have learned many lessons that I will carry with me onwards.

I also want to thank all my friends: be it those I've met in Porto, those back home from the "good old days", or those I've met abroad. We've shared many adventures and laughs, and I've always been able to count on you.

I want to thank the Center of Applied Photonics at INESCTEC for the opportunity to carry out this work in a great environment.

Finally, a big thank you to my family, without whom none of this would have been possible.

UNIVERSIDADE DO PORTO

Abstract

Faculdade de Ciências da Universidade do Porto
Departamento de Física e Astronomia

MSc. Engineering Physics

Optical Extreme Learning Machines: a new trend in optical computing

by Duarte SILVA

As we begin to drift away from the Von Neuman computing paradigm, new disruptive technologies are needed to accomodate the ever-increasing hunger for greater computing capacity. Such need for innovation leaves an open playground for the ressurgence of analog computing. Optics, particularly, has seen an opportunity for the rejuvenation of the race for the long-sought optical computer.

An *Extreme Learning Machine* (ELM) is a single layer feed-forward neural network, which consists of a non-linear projection of an input to a high dimensional output space, where the training then takes place. Its simplicity makes it highly attractive for hardware implementations. Our goal is to study and implement an ELM within the optical domain.

We start by developing a theoretical framework based on the transmission matrix formalism that allows us to model the information flow of our ELM. In particular, we aim to examine the dimensionality of the output space and discuss its learning capabilities, with respect to the input fields. We then perform numerical simulations which validate the theoretical model, and benchmark the machine on standard machine learning (ML) tasks. For its physical implementation, we artificially encode information on an input electric field's phase and amplitude profiles, in order to study our model. We have given experimental proof that validate the theoretical framework, and we've benchmarked the system as in the simulations, having achieved consistent results throughout the experiments. We also discuss real-world applications of our machine and alternative platforms.

Finally, we have performed a first proof-of-principle experiment towards an analog optoelectronic computer, based on the ELM architecture. We discuss current challenges and propose a future experiment to overcome them.

UNIVERSIDADE DO PORTO

Resumo

Faculdade de Ciências da Universidade do Porto
Departamento de Física e Astronomia

Mestrado Integrado em Engenharia Física

***Extreme Learning Machines* óticas: uma nova tendência em computação ótica**

por Duarte SILVA

À medida que nos afastamos do paradigma de computação de Von Neuman, surge a necessidade de novas tecnologias capazes de acomodar a nossa crescente sede por uma maior capacidade de computação. Tal demanda por inovação abre portas para a ressurgência de computação analógica. Ótica em particular, viu uma oportunidade para a rejuvenescência da corrida para o tão aguardado computador ótico.

Uma máquina de *Extreme Learning Machine* (ELM) é uma rede neuronal *feed-forward* com apenas uma camada interna, que consiste numa projeção não linear de uma entrada para um espaço de saída de elevada dimensionalidade, onde o treino acontece. A sua simplicidade torna-a altamente atrativa para aplicações em hardware. O nosso objetivo é estudar e implementar uma ELM no domínio ótico.

Começamos por desenvolver um modelo teórico baseado no formalismo da matriz de transmissão, que nos permite modelar o fluxo de informação da nossa implementação da ELM. Em particular, pretendemos estudar a dimensionalidade do espaço de saída e discutir as suas capacidades de aprendizagem. Realizamos simulações numéricas que validam o modelo teórico, e avaliamos a sua *performance* em tarefas tradicionais de *machine learning* (ML). Para a sua implementação física, codificamos artificialmente informação nos perfis de fase e intensidade do campo elétrico de entrada, de forma a estudar o nosso modelo. Com isto, recolhemos provas experimentais que validam o modelo teórico, e avaliamos a *performance* do sistema tal como nas simulações, tendo obtido resultados consistentes. Discutimos ainda aplicações e plataformas alternativas.

Finalmente, desenvolvemos uma experiência como prova de princípio para um computador optoeletrónico analógico, baseado na arquitetura de uma ELM. Discutimos os desafios atuais e propomos uma experiência futura que os permite ultrapassar.

Contents

List of Figures

Chapter 1

Introduction

One of the most impressive characteristics of modern society is the ever increasing rate by which it evolves. Such advance can be largely attributed to our ability and ambition to process, store and transmit ever-increasing amounts of data. This has created a fertile environment for every field of science and technology to grow, stimulating unforeseen progress. Despite this, we are now at a point in history where our needs have largely surpassed our computing abilities. Indeed, the amount of data generated globally is (and has been) increasing at an exponential pace. This is largely motivated by the easiness of collecting such data volumes allied with the high efficiency upon transmission and storage. Within this context, it was created a race for the next-generation computing platform, allowing new technologies to emerge, namely, new computing architectures, new materials and devices and even new computing paradigms altogether.

At the same time, machine learning (ML), which is the field that specializes in finding actionable information hidden within some data, has seen an outstanding progress over the years, and can now be found in numerous aspects of our lives. However, its success and efficacy has been linked to the capability of processing large amounts of information, which has been unable to keep up with current demands. Nonetheless, the specificity of the calculations required by ML algorithms has rekindled the interest in a once-abandoned computing platform: analog computing.

This platform is based on the principle of "letting nature do the computations for us". In short, a physical system is set-up such that the equations that govern its behaviour are similar to the problem we want to solve. Under certain conditions, these systems can offer staggering energy efficiency and speed, but historically they were largely abandoned in mid 20th century, mostly due to high development and maintenance costs, and

calculation inaccuracies. Nowadays, however, due to our technological progress, costs associated with such systems can be greatly reduced, and certain applications, particularly ML algorithms, are able to withstand a higher degree of inaccuracy in calculations. For these reasons, analog systems have become relevant once more. While there are various suitable physical platforms that can be used, optics stands as a natural candidate as opposed to conventional electronics in terms of energy efficiency, speed and parallelism.

In this thesis, we will begin by looking at the history of the general purpose computer to understand the computational plateau that we have reached and why we need novel computing paradigms. We will draw special focus towards ML and how it relates to specialized hardware. Our goal is to study and implement a particular ML architecture analogically, with optics as its backbone. Ultimately, we wish to extend our system to a purely optical domain.

1.1 An opportunity for specialized hardware: a series of fortunate events

The first programmable general purpose computer was completed in 1945, the ENIAC [1], and it was a massive engineering endeavour: it had over 17 000 vacuum tubes and consumed over 150 kilowatts of power in operations*. However, reprogramming this device meant to physically change it by hand, retaining no memory of past programs, being a rather slow and inefficient process. In that same year, John Von Neumann introduced an innovative computer architecture [2] which allowed an efficient operation of a general-purpose computer. Two years later, in 1947, John Bardeen, Walter Brattain, and William Shockley managed to make the first working transistor [3], and two years after that, the first patent was filed for a device that resembled that of an integrated circuit [4]. With these inventions, the world now had not only a good computer architecture to rely on, but also a compact, energy efficient and scalable technological platform, which allowed the field of modern electronics to be established and grow. So much so that in 1965, Gordon Moore predicted that the number of components in an integrated chip would double every two years [5], which served as a guide for long term planning for research centres and companies. Finally, 9 years after that, in 1974, Robert Dennard, established what is known as the *Dennard scaling* law [6]. To put it simply, this law states that for a given

*In comparison, today's laptops consume only dozens of watts, with a hardware reliability within the yearly range.

area of silicon, if we make the transistors smaller, but put more of them in the same area, the power we need to use for operation remains constant, thus being a major motivation for component miniaturization. With all of these innovations, it was created a fertile environment for modern electronics.

While this revolution was taking place, other computing technologies tried to keep up, namely analog and optical computing. Despite offering fundamental advantages in terms of speed, capacity and energy efficiency, they were unable to follow suit. The reason being that their electronics counterpart were providing computing capabilities that were growing at an exponential rate. Fast forward a few decades, it was found that the Dennard scaling law overlooked the leakage current and threshold voltage of the devices, which created a *power wall* that establishes a fundamental limit on the energy required to operate them. As we've gotten to build electronic components at the nanometer scale [7, 8], we have hit this power wall, and further miniaturisation no longer translates in a performance increase. For this reason, at around 2004 [9], the industry started looking into multi-core processing rather than single core, as can be seen in figure 1.1. Alas, the general purpose computing economic cycle has slowed down, and has led to researchers and industry leaders to discuss possible paths towards the future of computing [10, 11]. After a long quiescence, an opportunity for optical and analog platforms is among us.

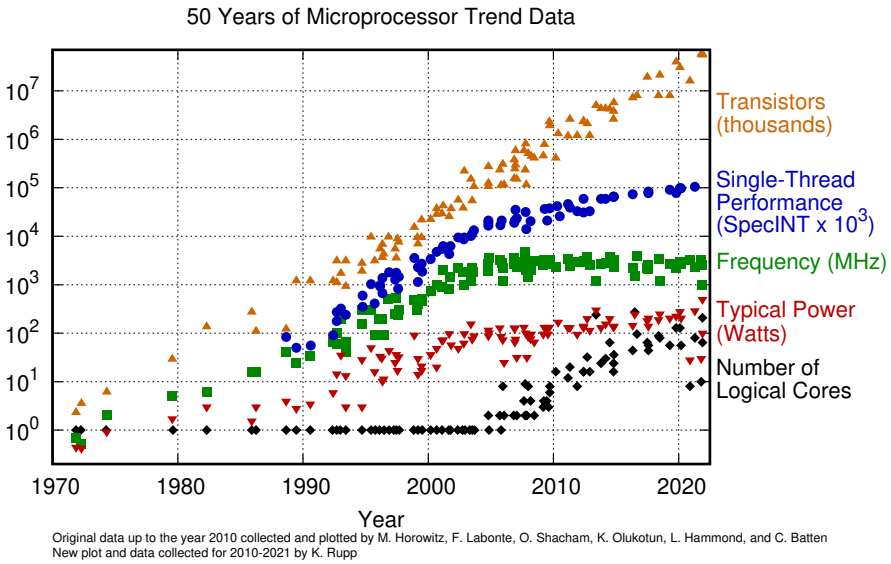

FIGURE 1.1: 50 years of microprocessor trend data. Figure from Kunle Olukotun, Lance Hammond, Herb Sutter and Mark Horowitz.[12]

Along with these alternate technologies, another field of research saw a period of disinterest and reduced funding amidst the electronics implosion: artificial intelligence. In 1958, the world witnessed what is recognized as the "first ever neural network": the percetron, developed by Frank Rosenblat [13]. However, as the field grew, its algorithms required computing resources that simply were not available, and it was not until the last couple of decades that this field has truly flourished [14–17]. The year 2012 was particularly interesting for the field of Deep Learning*, as a team of researchers introduced the neural network called *AlexNet* [18] and decided to participate in an image classification contest called *ImageNet* [19]. This team achieved record-breaking performance, and what set them appart was the size and depth of the network, which had 8 layers and a total of 500 000 thousand neurons, which leads to millions of parameters to optimise. To do it, they pioneered the use of GPU's for artificial intelligence, which made heavy use of parallel computations. This feat revealed two things: i) the success of a neural network is intimately linked to its scale, and ii) the use of specialized hardware for artificial intelligence can be very rewarding. In the years that followed, there was a large appetite in the industry for specialised hardware, particularly for Deep Learning: Google with it's Tensor Processing Unit (TPU) [20], Nervana's AI architecture [21] and Meta's Big Sur [22]. Thus, if specialised hardware allowed for Deep Learning to rise, now it is Deep Learning driving the innovation within specialised hardware. From an economic and technological point of view, we are at a golden age to explore new models of computation for ML applications.

1.2 Machine Learning and Extreme Learning Machines

Machine Learning is the field that specializes in finding concrete information within unstructured data. With today's technology, it is very easy to collect, store and transmit data. In fact, it is estimated that by 2025 the volume of data created, stored and transmitted may surpass the 180 zettabytes [23]. The challenge is to process and infer on this data, which is where ML algorithms truly shine. While these algorithms can be quite flexible, the types of things they can actually do can be described in very broad terms. We can classify information (for example, to know if an image contains a dog or not), we can make predictions based on inputs and find trends (for example, to predict the weather based on a set of meteorological data like yesterday's temperature, humidity, wind, etc.),

*a sub-field of artificial intelligence, to be explored later.

we can find where the effective information of some data is (for example, not all the pixels in an image may contain relevant information), among others. These are some examples of what they can do, although there are many more applications [24]. Within the available algorithms, *artificial neural networks* (ANNs) have been proven to achieve remarkable performance [14–17]. These algorithms are inspired in the human brain, and constitute a network of artificial neurons.

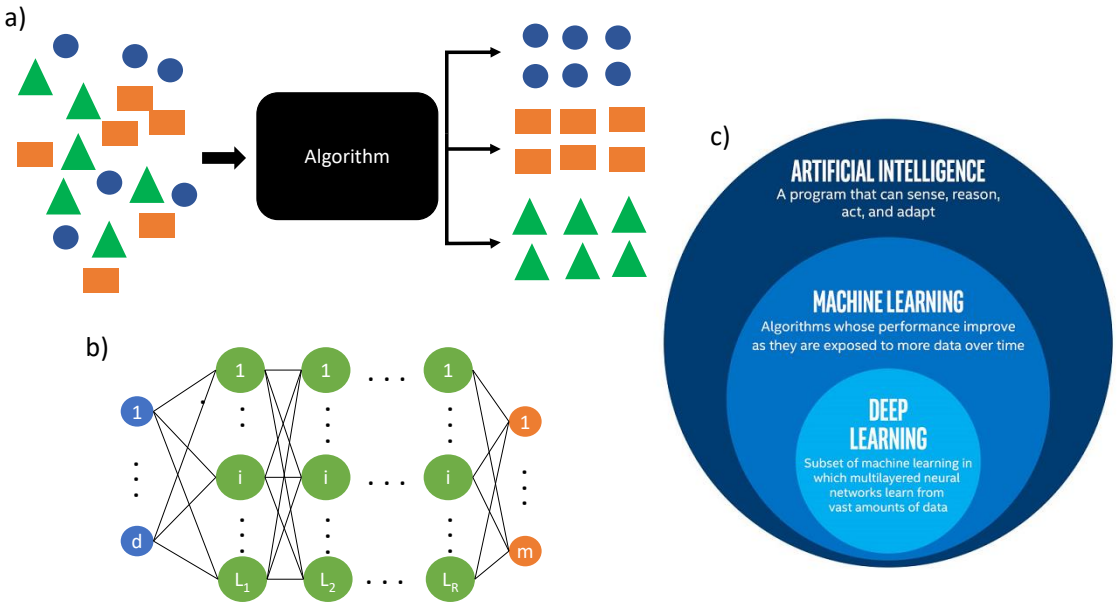

FIGURE 1.2: Illustration of Machine Learning algorithms. In a) you can find an example of a classification or clustering problem, a typical ML problem, while in b) is a depiction of an artificial neural network. In c) is outlined how the different terms within artificial intelligence relate between themselves. Image taken from [25]

As in the brain, each neuron receives and transmits information to other neurons. In ANNs, they are arranged in layers, and the field of Deep Learning is founded on ANNs with a great number of layers with each having a large number of neurons. As each neuron has to be finely tuned, the optimisation problem to be solved can quickly achieve a high number of dimensions, making it hard to solve. Luckily, we have at our disposal an efficient algorithm to train these networks called *backpropagation* [26]. Despite its effectiveness, the amount of data needed to train some networks can be quite overwhelming, for example, DeepMinds's AlphaGo [14], the first ever AI to overcome human performance, used 38 million positions to train the algorithm, and more recently DALL-E 2 [17] relied on a dataset of 250 million images. Using the backpropagation algorithm with such volumes of data can consume a great amount of time and energy. For this reason, there has been an effort in bypassing this algorithm to allow for a more efficient training. A recent

approach lies within Extreme Learning Machines (ELMs), developed a decade ago by Huang et al. [27–30]. Simply put, it is a neural network with a single hidden layer, called a *reservoir*, whose neurons are not going to be optimised. The optimisation takes place only at the output layer, which can be done via a *linear regression*, a very cheap and fast training algorithm. Despite its simplicity, it can be shown that it can achieve remarkable performance [28]. Furthermore, this same simplicity makes it highly attractive for hardware implementations. For this reason, we are interested in studying and implementing an ELM within the optical domain.

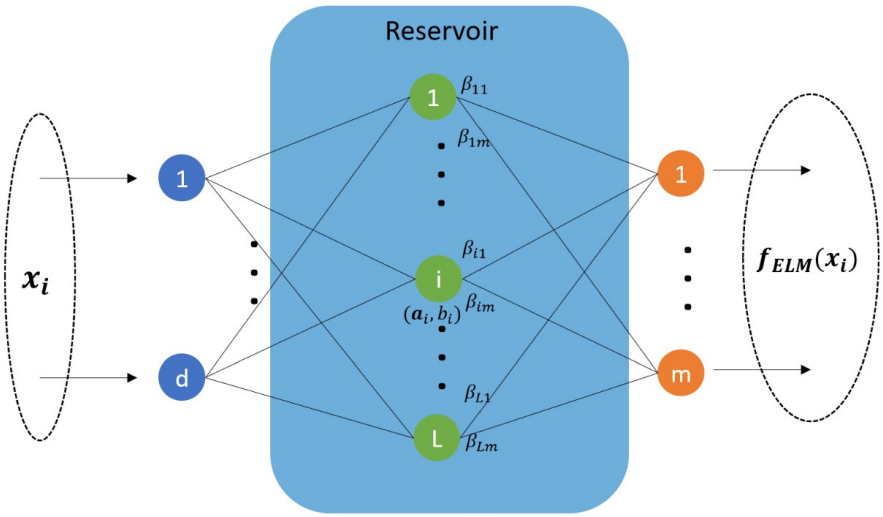

FIGURE 1.3: Illustration of an ELM architecture.

1.3 Optical neural networks

Due to the bosonic nature of photons, they do not mutually interact, and allying this with the large bandwidth at our disposal, optical information processing can be massively parallellizable and energy efficient. In addition, optical devices have much faster response times in comparison with electronics, thus making such processing very fast. For these reasons, realizing ANNs in optical platforms has gained much attention, particularly deep networks.

The types of networks usually employed with optics are Hopfield Neural Networks (HNNs) [32, 33], a type of NN where all the neurons are linked with each other, and each neuron is both an input and output; Multilayered Perceptrons (MNNs) [26], where informations flows only from left to right in multiple layers; Reservoir Computing (RC)

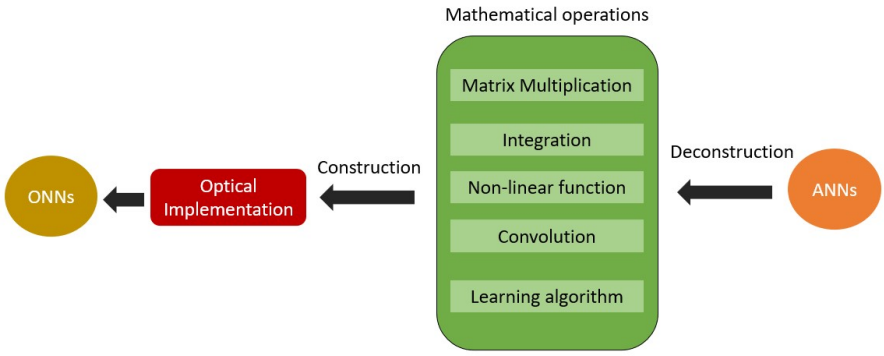

FIGURE 1.4: Illustration of the relationship between ANNs and ONNs. Recreated from [31].

[34, 35], which, similarly to ELMs, consists of a fixed and random reservoir of hidden neurons, the only difference being that it allows temporal dynamics and can retain memory of past inputs; and finally, ELMs, which we have already covered. In order to implement these machines, we first need to deconstruct the mathematical abstractions and be able to implement each operation individually, as per figure 1.4. Out of the five mathematical operations mentioned, three stand-out: i) matrix multiplication, ii) non-linear function implementation and iii) training algorithm. These challenges have been thoroughly studied over the last decades. For example, the matrix multiplication can be implemented by making use of a number of physical mechanisms, such as light transmission [36, 37], diffraction [38–40], interference [41] and even scattering [42]. These works focus on guaranteeing reliable connections between the neurons. As for the non-linear activation functions, the common approaches have been to either use optical non-linearities [43, 44], which is particularly attractive due to their ultrafast operation, or we resort to optoelectronic approaches [45] through light intensity measurements. Finally, the back-propagation algorithm still remains as the preferred training method [40, 46], as it is valid as long as we have a nice mathematical model for information propagation along our physical system.

Despite these advancements, real implementations of ONNs based on precise neuron connections (HNN and MNN) are quite difficult, often limited by materials and devices imperfections. Furthermore, they still rely on an intensive training algorithm, which can be ineffective if the neurons connections are not properly done. For these reasons, optical RC and ELMs have gained much attention.

The field of optical ELMs is still in its infancy, and the available literature is still quite limited. However, one of the earliest implementations can be traced back to Saade et al.

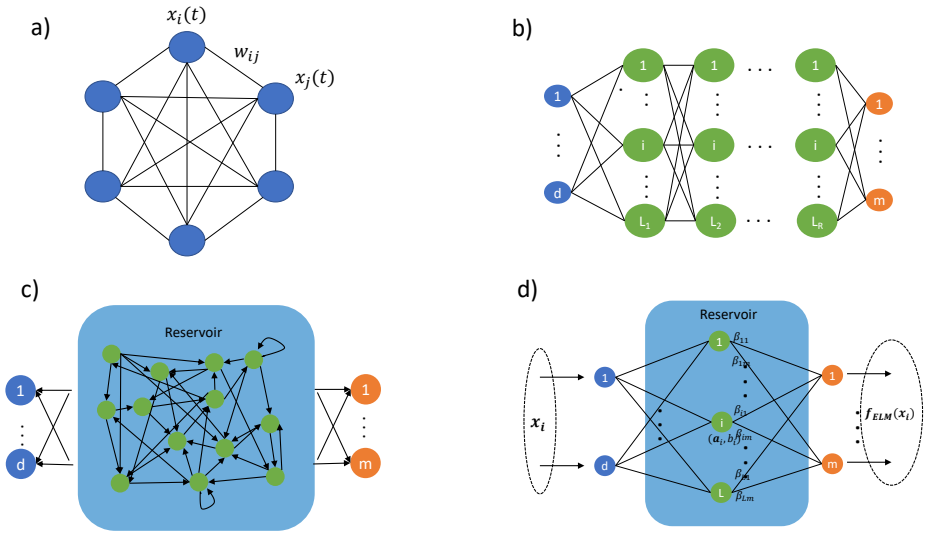

FIGURE 1.5: Common ANNs implemented within the optical domain. a) the Hopfield NN, b) a Multilayered Perceptron NN (MNN), c) a Reservoir Computing (RC) architecture and d) an Extreme Learning Machine (ELM) architecture.

[47], where they made use of an optical complex media as a reservoir, followed by intensity measurements. Years later, some works have also been done exploring such machines with a Kerr non-linearity as a reservoir [48, 49]. As for the platform used, it has been demonstrated to perform well in free-space [47, 50, 51], fiber optics [52] and even in integrated optical chips [53]. Nonetheless, the success of these approaches have been largely empirical, thereby lacking a fundamental description of the inner workings of these machines. To that end, we will take inspiration in the work of Saade et al. [47], and perform a thorough study on the learning abilities of a particular optical ELM implementation.

1.4 Outline of this thesis

The goal of this thesis is to study and implement an optical set-up of a particular version of an extreme learning machine, towards the goal of building a fully analog optoelectronic computer. This thesis is divided as follows: in the current chapter we have looked at the historical events that reveal the technological relevance of our work; briefly introduced key ML aspects and reviewed the current state of the art on ONNs, with particular emphasis on optical ELMs. The outputs achieved over the course of this thesis are also given at the end of this chapter.

In chapter 2, we introduce the reader to the machine learning landscape, and follow to the mathematical foundations of ELMs. After, we focus on optical ELMs and discuss the

current understanding within the literature of the learning capabilities of ELMs. Then, we develop a mathematical framework based on the transmission matrix formalism to model the information flow in an optical set-up based on optical complex media. This model encompasses phase, amplitude and phase and their combined information encoding, and permits us to directly infer on the output space dimensionality of the ELM input projection.

In chapter 3, we present a set of numerical simulations that have allowed us to validate our theoretical model in the different encoding regimes. We have also used these simulations to benchmark the ELM, and analysed its performance with the presence of a strong physical non-linearity.

In chapter 4, we outline the different experimental methods and equipments used throughout the various experiments. Particularly, we introduce the spatial light modulator we've used, a digital micromirror device, and give some experimental tips and tricks to help in a set-up which may come in handy for future works. We then introduce our phase and amplitude wavefront modulation techniques. Finally, we describe the detector array, and outline the main software used.

In chapter 5, we present our experimental results of an optical extreme learning machine as modeled in chapter 2. We discuss the validity of our theoretical framework upon different scenarios. Simultaneously, we also benchmark the system in standard machine learning tasks, having achieved remarkable performance in some cases. We also discuss the implementation of our technology in real-world applications, and make a quick technological assessment for such implementations.

In chapter 6, we propose to extend the results from chapter 5 further into the analog domain, and allow for calculations previously done in the digital domain, to now be performed analogically. We present our experimental results and outline current challenges to overcome. Finally, we propose a future experiment which should surpass such difficulties and provide a higher performance, while assessing compatibility with other hardware platforms for a potential commercial deployment.

In chapter 7 we present our conclusions, and future work perspectives.

Finally, along the development of this thesis, there were opportunities to explore other topics that, despite not being directly related to this thesis' main research, have contributed to our working group research directions, and also constitute an important step

towards our work. The work developed within them is included here for completeness. Such topics include:

1. A real-time phase retrieval software based on off-axis digital holography technique (see Appendix C);
2. A study of liquid crystal on silicon (LCoS) spatial light modulators and calibration procedures (see Appendix B);
3. A study on the theory of optical complex media, as well as a study on the theory of the transmission matrix formalism and experimental procedures measuring it (see Appendix D and Appendix E);
4. A study on wavefront optimisation algorithms applied to focusing through optical complex media (see Appendix F).

1.5 Thesis outputs

In the development of this thesis, the author has contributed with 1 conference paper and 4 oral presentations as first author, one of which has been distinguished for *Best student oral presentation* at an international conference. As co-author, it has resulted in 1 scientific paper (in submission), 1 conference paper, 2 oral presentations and 1 poster presentation.

As first author:

Articles and conference proceedings

1. *"Unravelling an optical extreme learning machine"* - Duarte Silva, Nuno A. Silva, Tiago D. Ferreira, Carla C. Rosa, Ariel Guerreiro. Preceedings of EOSAM - European Optical Society Annual Meeting. [54] (2022)

Oral presentations

1. *"Unravelling an optical extreme learning machine"* - Duarte Silva, Nuno A. Silva, Tiago D. Ferreira, Carla C. Rosa, Ariel Guerreiro. EOSAM - European Optical Society Annual Meeting (2022). Distinguished for *Best Student Oral Presentation*.

2. *"Taming light for novel computing machines"* - Duarte Silva, Nuno A. Silva, Tiago D. Ferreira, Carla C. Rosa, Ariel Guerreiro. FÍSICA 2022 - 23ª Conferência Nacional de Física e 32º Encontro Ibérico para o Ensino da Física. (2022)

3. *"Shedding light on the inner workings of an optical extreme learning machine"* - Duarte Silva, Nuno A. Silva, Tiago D. Ferreira, Carla C. Rosa, Ariel Guerreiro. IMOS2022 – Iberian Meeting of Optics Students [55]. (2022)

4. *"Harnessing speckle patterns for an optical extreme learning machine"* - Duarte Silva, Nuno A. Silva, Tiago D. Ferreira, Carla C. Rosa, Ariel Guerreiro. IJUP2022 – 15th Young Researcher Meeting of University of Porto. (2022)

As co-author:

Articles and conference proceedings

1. *"Towards the experimental observation of turbulent regimes and the associated energy cascades with paraxial fluids of light"* - Tiago D. Ferreira, Vicente Rocha, Duarte Silva, Ariel Guerreiro, Nuno A. Silva. Article submitted to the New Journal of Physics.

2. *"Reservoir computing with nonlinear optical media"* - Tiago D. Ferreira, Nuno A. Silva, Duarte Silva, Carla C. Rosa, and Ariel Guerreiro. Proceedings of the V International Conference on Applications of Optics and Photonics (2022). To be published.

Oral presentations

1. *"Using fluids of light in photorefractive media to create turbulent states"* - Tiago D. Ferreira, Nuno A. Silva, Duarte Silva, Vicente Rocha, Carla C. Rosa, and Ariel Guerreiro. FÍSICA 2022 – 23ª Conferência Nacional de Física e 32º Encontro Ibérico para o Ensino da Física. (2022)

2. *"Experimental turbulent states with paraxial fluids of light in photorefractive media"* - Tiago D. Ferreira, Nuno A. Silva, Duarte Silva, Vicente Rocha, Carla C. Rosa, and Ariel Guerreiro. V International Conference on Applications of Optics and Photonics. (2022)

Poster presentations

1. *"Reservoir computing with nonlinear optical media"* - Tiago D. Ferreira, Nuno A. Silva, Duarte Silva, Carla C. Rosa, and Ariel Guerreiro. V International Conference on Applications of Optics and Photonics. (2022)

Chapter 2

Optical Extreme Learning Machines

As mentioned, the success of neural networks currently lies in its scale, be it in width or depth, but large scale networks are accompanied by an overwhelming number of parameters, thus the training of such network not only carries a large energy cost but can also be time-consuming, which can be counterproductive. In recent years there has been a rising interest in finding ways to tackle this problem and one solution lies in removing the training of the inner layers altogether. Such techniques are within the domain of reservoir computing and extreme learning machines. Both approaches consist of a network of hidden neurons with random fixed weights and biases, thus generating numerous possible complex behaviours in response to a certain input. The output data is then read by a single output layer that is optimised (i.e. trained) to solve a particular computational task. By doing so, the energetic cost and time spent during training are largely reduced, without compromising computing capabilities since such networks have been demonstrated to achieve comparable performances to standard AI methods [34].

The difference between reservoir computing and extreme learning machines is simple: the former is inspired in recurrent neural networks where information is allowed to flow backwards between the nodes, thus allowing to retain some memory of past inputs, while the latter consists on projecting an input into an output space of high-dimensionality. Contrary to reservoir computing, ELMs don't have any recurrence between neurons nor dynamical memory. Both are particularly attractive for hardware implementations due to the richness of non-linear dynamics and high number of degrees of freedom present in physical systems. At the same time, optics stands as a particularly attractive choice for such machines and there has been experimental realisations across several platforms, be

it at chip scale [53], simple free space propagation [50] or through speckle patterns either in free space or multi-mode fibers [47, 51, 52].

In this chapter we aim to introduce the reader to the world of machine learning by giving a birds-eye view of the field. Then, we move on to the mathematical aspects of an extreme learning machine, as outlined by Huang et al. [29] and review the state of the art regarding optical implementations of this framework. Finally, we propose a theoretical model that will shed some light on the mathematical intricacies and learning capabilities of a particular optical implementation.

2.1 Machine learning landscape

Machine learning (ML) can be defined as [56]

> *[The] field of study that gives computers the ability to learn without being explicitly programmed.*
>
> ARTHUR SAMUEL, 1959

The missing definition here is what we mean by "learning". Though it can be open to debate, we will take it to mean the ability to make predictions on a set of data, having previously been given a part of it. Indeed, that is what we as humans do: as children, we have been able to identify what a dog is, purely by having seen many and being told they were dogs, and when we saw a new living creature, we were able to draw conclusions on whether it was a dog or not, or even if it merely resembled one or not. Nonetheless, this is just an illustrative example, and it doesn't say much about why should we use ML in the first place. Géron [24] has beautifully summarized the highpoints of ML:

- In problems for which existing solutions require a lot of fine-tuning or long lists of rules, a Machine Learning algorithm can often simplify code and perform better than the traditional approach.
- In complex problems for which using a traditional approach yields no good solution, the best Machine Learning techniques can perhaps find a solution.
- A Machine Learning system can adapt to new data, making it useful in fluctuating environments.

- Machine learning algorithms can get insights about complex problems and large amounts of data that would otherwise be overlooked.

Most applications that we see nowadays of this field often relate to one or more of the previous points. Examples include the detection of tumours in brain, detecting credit card fraud, recommending a product that a client may be interested in, based on past purchases, and many others.

While there are many different types of machine learning algorithms, they can be classified according to three broad criteria:

- If they are trained with or without human supervision (supervised, unsupervised, semisupervised, and Reinforcement Learning);
- If they can learn incrementally as new data comes in (online versus batch learning);
- If they work by simply comparing new data points to known data points, or by detecting patterns in the training data and building a predictive model (instance-based versus model-based learning).

Any algorithm may fulfil one or many of these criteria.

Another aspect to consider is what kind of things can an ML algorithm do. This is very general point, and a thorough answer is far too lengthy for this thesis, but we will give a general overview of its capabilities. Nonetheless, we invite the reader to read on Ref.[24] for a more careful review. As mentioned, a major characteristic that dictates the purpose of an ML algorithm is its human supervision. In *supervised learning* you feed the algorithm with a training set that includes inputs with respective labels. The goal with this is to train an algorithm to recognize an input and automatically infer on its label. For this learning method there are typically two intimately linked tasks: regression and classification. They largely differ on their output: in regression, we aim to predict a value given an input feature, whereas in classification we want to infer on the class an input belongs to. For example, we may want to classify if a new email is spam or not (classification), or may we want to know the next day's maximum temperature given a set of meteorological input values (regression). In *unsupervised learning*, however, the training data has no labels. The algorithms used often want to find hidden patterns in the data that will enable us to better analyse it. Take *clustering* as an example: in such algorithm we want to group bits of data together as if they belong to the same class. Other than that

you can also have anomaly detection and novelty detection, visualization and dimensionality reduction and association rule learning*. Though we have looked at supervised and unsupervised learning, there are other regimes worth looking at, including semisupervised learning, reinforcement learning and even batch and online learning. These are mentioned here for completion, and are only cited without further explanation. In figure 2.1 we provide a list of the most relevant ML algorithms for supervised and unsupervised learning.

Supervised Learning		**Unsupervised Learning**			
Regression	**Classification**	**Clustering**	**Anomaly and novelty detection**	**Visualization and dimensionality reduction**	**Association rule learning**
Linear Regression	Logistic Regression	K-Means	One-class SVM	Principal Component Analysis (PCA)	Apriori
	Naïve Bayes	DBSCAN	Isolation Forest	Kernel PCA	Eclat
Support vector machines (SVMs)		Hierarchical Cluster Analysis (HCA)		Locally Linear Embedding (LLE)	
Decision Tree				t-Distributed Stochastic Neighbor Embedding (t-SNE)	
K-nearest neighbours					
Neural Networks					

FIGURE 2.1: List of most of the most important supervised and unsupervised machine learning algorithms. List generated based on Ref.[24].

As for the training, ML problems are inherently optimization problems. Simply put, we usually have some model with a set of parameters, $\boldsymbol{\theta}$, that will generate an output to an input $\mathbf{x}_i$, $f(\boldsymbol{\theta}, \mathbf{x}_i)$, and our goal is minimise or maximise a certain function (usually called *cost function*), g. In the case of supervised learning, g is defined in relation to the data labels. A popular example is the *mean squared error* (MSE) defined as

$$MSE(\theta) = \frac{1}{m}\sum_{i=1}^{m}(f(\boldsymbol{\theta}, \mathbf{x}_i) - y_i)^2 \tag{2.1}$$

for the input $\mathbf{x}_i$ with label y_i. Their solutions can be found in two ways: either the problem is so nicely put that we have analytical solutions, whose prime example is the Linear Regression, or we resort to iterative algorithms with hopes to achieve a global solution, with

*These algorithms are well illustrated in Ref.[24], but we won't be diving deeper in them.

Gradient Descent and its variants being the most common choice in supervised learning tasks. Simply put, the Gradient Descent evaluates the derivative of the cost function locally and leads the algorithm through the path that has the biggest change towards a minimum.

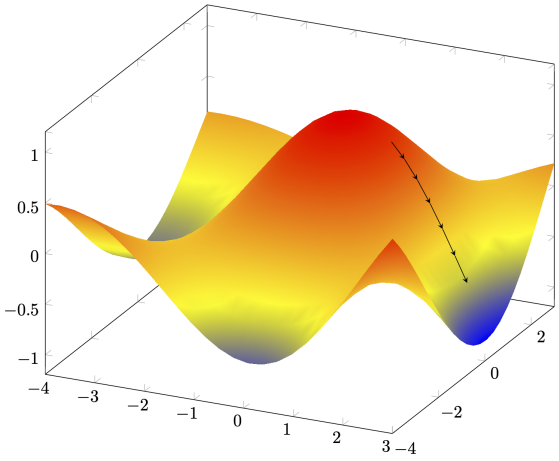

FIGURE 2.2: Illustration of the Gradient Descent algorithm on a 2D cost function. The arrows path illustrates the parameters of the model over the various iterations. Notice how the algorithm follows the steepest curve possible towards the minimum. Plot taken from Ref.[57].

Despite the wide plethora of ML algorithms available, there is one that has flourished far more than any other: Artificial Neural Networks (ANNs). Like many other technologies, ANNs take inspiration in nature, particularly in the brain's network of biological neurons. Referring to figure 2.3a), biological neurons produce short electrical impulses called *action potentials*, which travel along the axons to the telodendria, whose tips hold miniscule structures called *synaptic terminals* (or *synapses*), which are connected to the other neurons dendrites. Upon this potential, the synapses release chemical signals called *neurotransmitters*. When the next neuron receives enough of these neurotransmitters, it fires its own electrical signal, and the process continues along the neural network*. In essence, a single biological neuron is capable of collecting information from many other neurons (through the dendrites) and decide for itself if it sends a signal forward to the network or not. This is the inspiration to an artificial neuron. As per figure 2.3b), an artificial neuron takes in a vector of inputs $\mathbf{x} = [x_1, x_2, \ldots, x_N]$, and combines this information through a simple linear combination of the inputs, with respective weights $\{w_i\}_{i=1}^{N}$. Then it generates an output through a non-linear function, often called an *activation function*, as

*The real process is far more complex than what we describe, but such level of detail is unnecessary for our purposes.

$y = h\left(\sum_{i=1}^{N} w_i x_i\right)$. The goal is to connect many of these neurons together in an artificial neural network with many layers, as per figure 2.3c), so as to mimic a human brain.

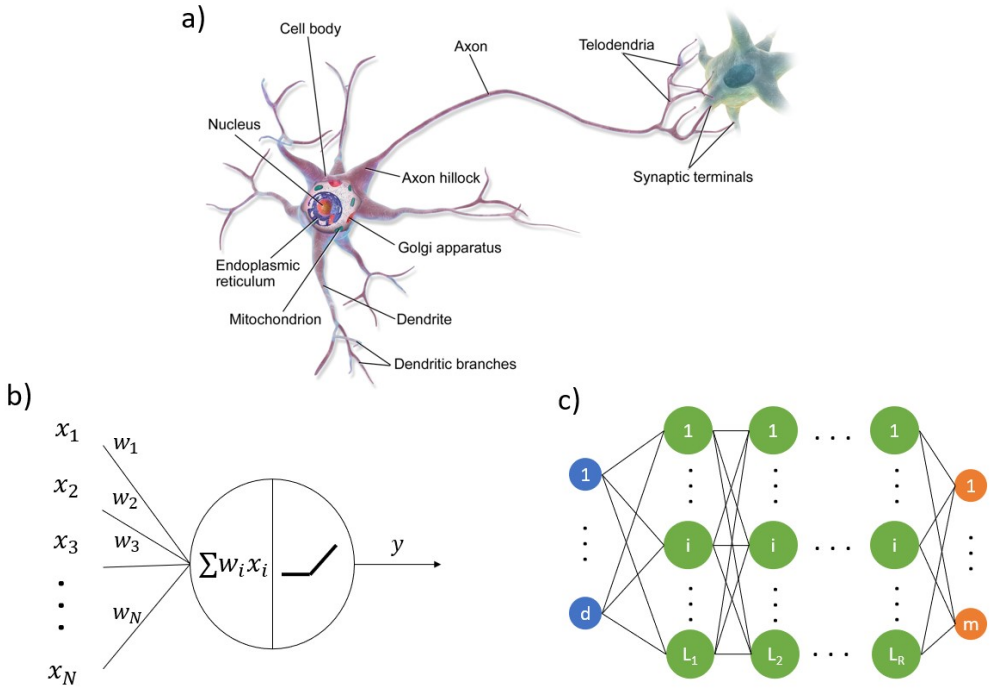

FIGURE 2.3: A comparison of a biological neuron to an ANN. a) human neuron, b) a single artificial neuron and c) an artificial neural network. Figure a) has been adapted from [58].

The model presented illustrates information flowing only forward, these are called *feed-forward artificial neural networks* (FFANN). However, over the years many architectures have been proposed, some of which allow information to flow backwards within the network, or even retain information from previous inputs. Nonetheless, we will restrict ourselves to the FFANNs. Regarding training, one can already see that even fairly simple networks can result in a very high dimensional optimization problem, as each neuron has a weight, w_i, that needs to be optimised. Furthermore, the non-linear activation functions make the problem even harder. Luckily, we have a training algorithm called *backpropagation algorithm* (BP) [26], introduced in 1986, which allows an efficient training. In short, it is a variant of the Gradient Descent which uses a clever technique for computing the gradients automatically, due to the differentiability of the activation functions. Despite its efficacy, we have come to a point where the scale of a standard neural network has become very large, and allied with the fact that we often need massive amounts of data to train our network, makes the use of the BP algorithm a a highly inneficient task.

In this section we have given a general overview of the field of ML: we have seen why we should use it and when; what kind of algorithms are there; what kind of things ML algorithms can do for us; how we can train them, and we've introduced the basic concepts of ANNs. In the next section we will introduce the mathematical formalism and essence of an ELM, and we will see how it fits within the ML landscape.

2.2 ELM in a nutshell

As we've seen, an ELM consists on projecting an input space into an output space of high dimensionality through a hidden layer, and performing the intended computational task on this new data. Consider an input $\mathbf{x}$. The output function of an ELM can be written as

$$f_{ELM}(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \cdot \boldsymbol{\beta} \tag{2.2}$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_L]^T$ is the output weight vector between the hidden layer of L nodes to the $m \geq 1$ output nodes and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]^T$ is the ELM's nonlinear feature mapping. The output functions of the hidden nodes don't need to be unique, that is, different neurons may have different functions. This leads us to generalise $h_i(\mathbf{x})$ for a d-dimensional input,$\mathbf{x}$, as

$$h_i(\mathbf{x}) = G(\mathbf{a}_i, b_i, \mathbf{x}),\ \mathbf{a_i} \in \mathbb{R}^d, b_i \in \mathbb{R} \tag{2.3}$$

where $G(\mathbf{a}, b, \mathbf{x})$ is a nonlinear piecewise continuous function satisfying ELM universal approximation capability theorems [27, 59, 60]. As can be seen from figure 2.4, an ELM can be seen as a training of a single layer feedforward neural network in two stages: i) random feature mapping and ii) linear parameters solving. In the first step, we see that a d-dimensional input, $\mathbf{x}_i$, is mapped onto an L-dimensional space, $\mathbf{h}(\mathbf{x}_i)$, while in the second step we must find the adequate weight values $\boldsymbol{\beta}$ that will better approximate some target function.

In more detail, the values of $\boldsymbol{\beta}$ are found via a least squares minimisation problem

$$\min_{\beta \in \mathbb{R}^{L \times m}} ||\mathbf{H}\boldsymbol{\beta} - \mathbf{T}||^2 \tag{2.4}$$

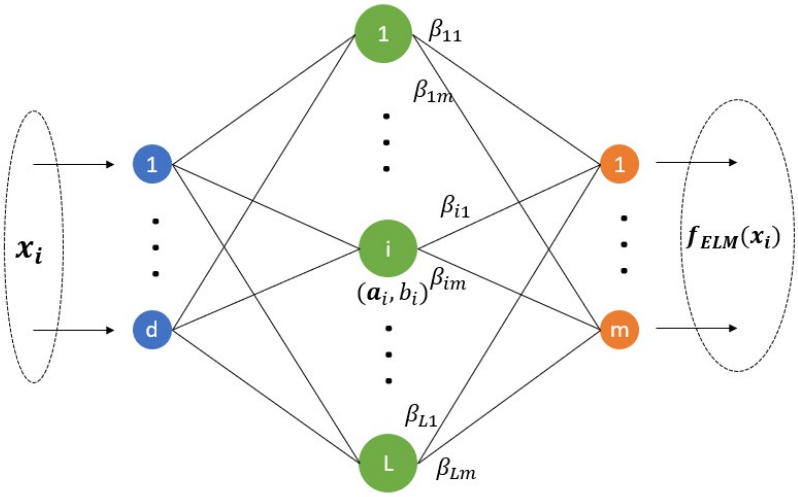

FIGURE 2.4: Illustration of an ELM architecture.

where $\mathbf{H}$ is the hidden layer output matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \dots & h_L(\mathbf{x}_1) \\ \vdots & & \vdots \\ h_1(\mathbf{x}_N) & \dots & h_L(\mathbf{x}_N) \end{bmatrix} \tag{2.5}$$

$\mathbf{T}$ is the training data target matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_N \end{bmatrix} = \begin{bmatrix} t_{11} & \dots & t_{1N} \\ \vdots & & \vdots \\ t_{N1} & \dots & t_{Nm} \end{bmatrix} \tag{2.6}$$

$\boldsymbol{\beta}$ is the output weight matrix

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_N \end{bmatrix} = \begin{bmatrix} \beta_{11} & \dots & \beta_{1N} \\ \vdots & & \vdots \\ \beta_{N1} & \dots & \beta_{Nm} \end{bmatrix} \tag{2.7}$$

and $||\cdot||$ denotes the Frobenious norm. This problem has an analytical solution given by $\boldsymbol{\beta} = \mathbf{H}^{+}\mathbf{T}$ where $\mathbf{H}^{+}$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{H}$. A particularly interesting aspect of the ELM architecture is that, unlike traditional learning algorithms, it aims to satisfy several targets simultaneously, as pointed out in Ref.[28], which we replicate here:

1. **Generalization performance**: Most algorithms proposed for feedforward neural networks do not consider the generalization performance when they are proposed for the first time. ELM aims to reach better generalization performance by reaching both the smallest training error and the smallest norm of output weights:

$$\min_{\beta} ||\boldsymbol{\beta}||_p^{\sigma_1} + C||\mathbf{H}\boldsymbol{\beta} - \mathbf{T}||_q^{\sigma_2} \tag{2.8}$$

where $\sigma_1 > 0$, $\sigma_2 > 0$, p, q = 0, 1 2 , 1, 2, . . . ,+∞. The first term in the objective function is a regularization term which controls the complexity of the learned model.

2. **Universal approximation capability**: Although feedforward neural network architectures themselves satisfy universal approximation capability, most popular learning algorithms designed to train feedforward neural networks do not satisfy the universal approximation capability. In most cases, network architectures and their corresponding learning algorithms are inconsistent in universal approximation capability. However, ELM learning algorithms satisfy universal approximation capability.

3. **Learning without "iteratively tuning" hidden nodes**: ELM theories believe that hidden nodes are important and critical to learning, however, hidden nodes need not be tuned and can be independent of training data. Learning can be done without iteratively tuning hidden nodes.

4. **Unified learning theory**: There should exist a unified learning algorithm for "generalised" networks [28], that is, it should be compatible with many kinds of neurons and its connections, hidden layers, as well as with different activation functions.

In this section we have introduced the formalism that allows us to understand an ELM and have given an intuitive picture as to why this should work. However, we haven't given any formal justification for it. To that end, we follow on to the next section.

2.3 Why should an ELM work?

A reasonable question to ask is: why should an ELM work? In all fairness, at first sight it seems that we let randomness take over and hope for the best! Which, coincidentally,

fits perfectly with some phenomena observed in nature, thus implying that nature could be a promising computing machine. Extreme learning machines have a strong theoretical support, based essentially on four theorems (see Ref.[28] and references therein):

Theorem 2.1. *Given any small positive value $\epsilon > 0$, any activation function which is infinitely differentiable in any interval, and N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i) \in \mathbb{R}^d \times \mathbb{R}^m$, there exists $L < N$ such that for any $\{\mathbf{a}_i, \mathbf{b}_i\}_{i=1}^{L}$ randomly generated from any interval of $\mathbb{R}^d \times \mathbb{R}$, according to any continuous probability distribution, with probability one, $||\mathbf{H}\boldsymbol{\beta} - \mathbf{T}|| < \epsilon$. Furthermore, if $L = N$, then with probability one, $||\mathbf{H}\boldsymbol{\beta} - \mathbf{T}|| = 0$.*

Theorem 2.2. *Given any nonconstant piecewise continuous function G:$\mathbb{R}^d \to \mathbb{R}$, if span$\{G(\mathbf{a}, b, \mathbf{x}) : (\mathbf{a}, b) \in \mathbb{R}^d \times \mathbb{R}\}$ is dense in L^2, for any continuous target function f and any function sequence $\{G(\mathbf{a}_i, b_i, \mathbf{x}_i)\}_{i=1}^{L}$ randomly generated according to any continuous sampling distribution, $\lim_{L\to\infty} ||f - f_{ELM}|| = 0$ holds with probability one if the output weights β_i are determined by ordinary least square to minimize $||f(\mathbf{x}) - \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x})||$*

Theorem 2.3. *Given any feature mapping $\mathbf{h}(\mathbf{x})$, if $\mathbf{h}(\mathbf{x})$ is dense in $C(\mathbb{R}^d)$ or in $C(\boldsymbol{\Omega})$, where $\boldsymbol{\Omega}$ is a compact set of $\mathbb{R}^d$, then a generalized single layer feedforward network with such a random hidden layer mapping $h(\mathbf{x})$ can separate arbitary disjoint regions of any shapes in $\mathbb{R}^d$ or $\boldsymbol{\Omega}$.*

Theorem 2.4. *The VC dimension* of ELM with L hidden nodes which are infinitely differentiable in any interval is equal to L with probability one.*

These theorems can be condensed in the following conclusions:

1. Theorem 2.1 tackles the interpolation capability of an ELM and tells us that an ELM can fit perfectly to any training set provided the number of hidden neurons is large enough;

2. Theorems 2.2 and 2.3 concern the universal aproximation capability of the model, and together tell us the necessary properties that the activation function G should have, and that ELM can approximate any complex decision boundary in classification provided the number of hidden nodes is large enough;

3. Finally, theorem 2.4 together with theorem 2.1 lead to the conclusion that ELM is an ideal classification model under the SRM (structure risk minimization) framework.

*The VC dimension is a measure of the capacity of a statistical classification algorithm, defined as the cardinality of the largest set of points that the algorithm can shatter [28]. Informally, the capacity of a model is related to how complicated it can be.

The theorems above give us proof that an ELM can be quite powerful. Admittedly, it requires some restrictions, particularly regarding the number of neurons and non-linearity, however, those turn out to not be limitations as they can be easily met either with digital or analog systems. However, a clever training is important, especially with respect to overfitting issues. Luckily, as the training is done via a linear regression, we can make use of regularization to overcome this problem.

2.4 Regularized ELM

A version of an extreme learning machine of particular interest is a constrained version of the problem statement in 2.4

$$\min_{\beta \in \mathbb{R}^{L \times m}} ||\mathbf{H}\boldsymbol{\beta} - \mathbf{T}||^2 + \lambda ||\boldsymbol{\beta}||^2 \tag{2.9}$$

which is known as Ridge regression and the resultant solution is known to be stabler and tends to have better generalisation performance [29]. Depending on your training set [49], you may have more training samples than hidden nodes ($N > L$), which yields the closed form solution for $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = \left(\mathbf{H}^T\mathbf{H} + \lambda \mathbf{I}\right)^{-1} \mathbf{H}^T\mathbf{T} \tag{2.10}$$

where I is the identity matrix of size L. Conversely, for the case where you have more output channels than training samples ($N < L$), the solution reads

$$\beta = \mathbf{H}^T \left(\mathbf{H}\mathbf{H}^T + \lambda \mathbf{I}\right)^{-1} \mathbf{T} \tag{2.11}$$

The precise derivations can be found in Appendix A.

Here we have framed the optimisation problem as a regularized linear regression, with a closed-form solution. Nonetheless, the ELM framework is general enough to allow compatibility with classification tasks.

2.5 ELM for classification

An extreme learning machine can also be applied to binary or multiclass classification. In our case we are interested in studying the binary classification task. To this end, we'll make use of the well-known logistic regression. As opposed to a linear regression where

we fit data to a hyperplane, in this model we fit the data to a logistic function given by:

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{2.12}$$

This function squeezes the output of a linear equation between 0 and 1, which is suitable for interpreting its output as a probability. In our case the input is to be regarded as the non-linear projection $\mathbf{x}_i \rightarrow \mathbf{h}(\mathbf{x}_i)$. Thus, for a target $y_i \in \{0, 1\}$, we predict the probability of an input $\mathbf{h}(\mathbf{x}_i)$ belonging to class 1 as

$$P(y_i = 1|\mathbf{h}(\mathbf{x}_i)) = \frac{1}{1 + \exp(-\mathbf{h}(\mathbf{x}_i)\beta)} \tag{2.13}$$

Then, we assume a threshold of 0.5 to attribute a prediction of the class. As an optimisation problem we aim to minimise the following [61]

$$\min_{\beta} C \sum_{i=1}^{N} \Big(-y_i \log\big(P(y_i = 1|\mathbf{h}(\mathbf{x}_i))\big) - (1 - y_i) \log\big(1 - P(y_i = 1|\mathbf{h}(\mathbf{x}_i))\big) \Big) + \frac{1}{2}\beta^T\beta \tag{2.14}$$

In equation 2.14 we introduce an l_2 regularisation parameter, similar to the ridge model, and we allow for a tunable hyperparameter C to avoid overfitting phenomena.

Up to this point we have introduced the ELM framework as a two step learning process: i) non-linear feature mapping to a high dimensional output space, and ii) a linear regression training algorithm for the output layer. The first step, however, is not a particularly new idea. There are ML algorithms that already implement this concept in a rather ingenious way (the so-called *kernel trick*). This begs the question: how does ELM fare with those algorithms?

2.6 ELM vs digital kernel methods

So far, we've mentioned that an extreme learning machine can be seen as a two-step learning process including a random non-linear projection of the input data onto a high-dimensional output. This idea of projecting an input was first implemented in machine learning through kernel machines. Simply put, we apply a non-linear function to our input data, $k(\mathbf{x_i}, \mathbf{x_j})$, called a *kernel function* defined as $k : \mathcal{X} \rightarrow \mathcal{X}$, and then we perform the computation on this new transformed data. In the case of linear regression with $N > L$, whose solution lies in equation 2.10, we simply need to replace $\mathbf{H}$ by the *Gram matrix*,

$\mathbf{K} \in \mathbb{R}^{n\times n}$, whose entries are $K_{ij} = k(\mathbf{x_i}, \mathbf{x}_j)$. The solution then reads

$$\boldsymbol{\beta} = \left(\mathbf{K}^T\mathbf{K} + \lambda\mathbf{I}\right)^{-1}\mathbf{K}^T\mathbf{T} \tag{2.15}$$

The reason why this may work in many datasets is best seen when we write the kernel function as an inner product in a high dimensional space $\mathcal{V}$ with a defined inner product, through a feature map $\phi(\mathbf{x}) : \mathcal{X} \to \mathcal{V}$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \right\rangle_{\mathcal{V}} \tag{2.16}$$

Thus we see that the action of a kernel function k is to project the data onto a high dimensional space through $\phi(\mathbf{x})$ and compute a "similarity measure" between different samples through an inner product. Notice that this projection is implicit, meaning that we needn't to specify the feature map as long as the inner product is explicit. In fact, for a given function $k(\mathbf{x_i}, \mathbf{x_j})$, there isn't a unique feature map $\phi(\mathbf{x})$ satisfying the same inner product. The main motivation behind this *kernel trick* is that while a dataset may not be linearly separable in $\mathcal{X}$, it may be so in $\mathcal{V}$, information which is revealed through an inner product in $\mathcal{V}$. The same principle is applied in an ELM through the hidden layer but with a key difference on the feature mapping which is now done *explicitly* onto the output channels. Looking at the solution 2.15, we note that with a kernel trick we'd be tasked with inverting a matrix of $N \times N$ while in an ELM we only need to invert an $L \times L$ matrix. If $N < L$, the scenario is reversed and computationally we'd be better off with a kernel method, however, in most real-world applications, this is hardly the case [50].

Thus far, we have looked into the general theory of ELMs. We have seen how it fits within the ML landscape, we have introduced the mathematical formalism and discussed the theorems that support this architecture. We have already highlighted the attractiveness of this framework for hardware implementations, but we haven't discussed it any further. In the next section, we review the state of the art on ELM implementations in an optical platform, and we will focus on a particular set-up based on optical complex media.

2.7 Optical Extreme Learning Machines

As stated in chapter 1, one of the earliest physical implementations of an ELM based on an optical set-up goes back to Saade et al. [47]. In their experiment, they make use of complex optical media to generate speckle patterns upon incidence of a modulated wavefront which carries encoded information. Such patterns are known to have gaussian circular statistics* in the complex field [62], which ensures the randomness needed for the hidden layer of an ELM. This output is converted to an intensity pattern by electronic measurement on a digital camera, whose pixels act as output channels. As for the non-linearity, it was guaranteed upon detection with intensity measurements, and electronic saturation. In their work, they introduce the set-up as a kernel machine and compare experimental data with an ansatz kernel inspired by infinite neural networks theories. This approach, while accurate, did not provide a deeper understanding of the architecture so as to infer on the nature of the explicit projection and the learning capability. Fast forward a few years, Marcucci et al. [49] explored the theory of neuromorphic computing using the non-linear Schrodinger equation as an effective reservoir

$$i\frac{\partial \psi}{\partial \zeta} + \frac{\partial^2 \psi}{\partial \xi^2} + \chi|\psi|^2\psi = 0 \tag{2.17}$$

In their work, they were able to find quantitative parameters that would allow to infer on the learning capability of the machine. It's important to note that, even though this was done in the context of reservoir computing, the conclusions are transferable to an ELM architecture. More specifically, they observed a *learning transition*, i.e. the machine would have the conditions for near-zero error, when rank $(\mathbf{H}) = N$ and $N = L$, with N being the number of training samples, L being the number of output channels and $\mathbf{H}$ being an $N \times L$ matrix containing all the outputs. In order to observe such behaviour, they were able to conclude that the rank $(\mathbf{H})$ would increase with increasing χ, that is, with increasing strength of the physical non-linearity, as per equation 2.17. The rank of a matrix can be regarded as measure of the "amount of information" of the matrix, that is, it is the greatest dimension of the vector space spanned by it's columns or rows. Therefore, when looking into the rank $(\mathbf{H})$ we are analysing the dimension of the output space. With this in mind, stating that having rank $(\mathbf{H}) = N = L$ gives rise to a learning transition, may be evidence of overfitting. This idea was explored by Silva et al. [48]. Indeed, within a

*A more detailed study of speckle statistics can be found in appendix D.

similar framework as Marcucci et al. [49], they were able to replicate the observed learning transition, but explored further the machine by testing its performance on noisy data. By doing so, they have observed a decrease in performance as N approaches L, which is explained by overfitting phenomena, commonly associated with the pseudo-inverse technique. Besides these works, there have been other implementations of optics-based ELM's [50–52] with remarkable success, however we are still lacking a deeper understanding of the inner workings of such a machine, as well as tools that would enable us to better infer on the learning capabilities of the machine.

Having said this, we take inspiration on the work carried out by Saade et al. [47] and, through an *ab initio* approach, we develop a theoretical framework based on the transmission matrix formalism [63]. This will allow us to better understand the mathematical structure of the input projection, and infer on the type of problems best suited for the machine. Furthermore, we establish metrics that will allow us to experimentally verify our model, and we'll also draw conclusions on the effects of strong physical non-linearities.

2.7.1 Mathematical model

Following figure 2.5, we will let an input optical field E_{in} evolve across some linear media. Since it is linear, we can borrow the transmission matrix formalism to write $E_{out} = ME_{in}$, with M being the so-called *transmission matrix*[*], and finally study the intensity[†] pattern defined as $I = |E_{out}|^2$. Let us define a set of K orthonormal vectors $\{\mathbf{e}_j^{in}\}_{i=1}^{K}$, such that

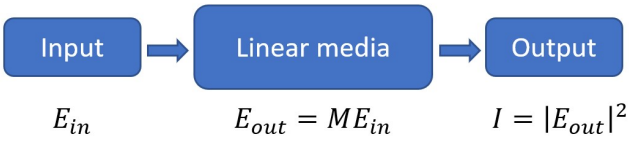

FIGURE 2.5: Diagram of the physical blocks of a set-up for an optical implementation of an ELM.

$\mathbf{e}_j^{in} = [\underbrace{0,\dots}_{\text{j-1 times}},1,\dots,0]^T$, and each $\mathbf{e}_j^{in}$ represents an independent input electric field. In order to encode the information in our input field, we'll allow for phase and amplitude

[*]The transmission matrix formalism is explored in detail in Appendix E.

[†]Not to be confused with the magnitude of the Poynting vector $|\mathbf{S}| = \frac{1}{\mu_0}|\mathbf{E}\times\mathbf{B}|$, though our quantity is proportional to the latter.

modulation through $f(\mathbf{x}_i) = a(\mathbf{x}_i)e^{ib(\mathbf{x}_i)}$, where $a_i \in [0,1]$ and $b_i \in [-\pi, \pi]$, then E_{in} reads

$$E_{in}(\mathbf{x}_i) = \sum_{j=1}^{K} f_j(\mathbf{x}_i)\mathbf{e}_j^{in} \tag{2.18}$$

The transmission matrix formalism is defined for a spatially sampled field, thus we will assume that our output field is sampled on an array of pixels. So as to represent the field in a single dimension, assume a row-wise ordering of a 2D display. With this in mind, we can make use of the transmission matrix $\mathbf{M}$ to write the output field, E_{out}, at the pixel l

$$E_{out}^{l}(\mathbf{x}_i) = \sum_{j=1}^{K}\sum_{k=1}^{K} f_j(\mathbf{x}_i)M_{lk}(\mathbf{e}_j^{in})_k \tag{2.19}$$

In the following, we will omit the functions arguments $(\mathbf{x}_i)$, and replace it by a superscript i. After some algebra*, the intensity can be written as

$$\begin{aligned} I_{out}^{il} = &\sum_{j=p=1}^{K} |a_j^i|^2|M_{lj}|^2 \\ &+ \sum_{j=1}^{K}\sum_{p>j}^{K} C_{ljp}a_j^i a_p^i \left\{ \begin{array}{c} \xi_{ljp}^{e}\left[\cos(b_j^i)\cos(b_p^i) + \sin(b_j^i)\sin(b_p^i)\right] \\ -\xi_{ljp}^{o}\left[\sin(b_j^i)\cos(b_p^i) - \cos(b_j^i)\cos(b_p^i)\right] \end{array} \right\} \end{aligned} \tag{2.20}$$

where C_{ljp}, ξ_{ljp}^{e} and ξ_{ljp}^{o} are real constants. Examining equation 2.20, we see that upon intensity measurements with amplitude and phase modulation, the output channels consist of polynomial and trigonometric functions of the encoding variables a_i and b_i, which is equivalent to applying an activation function of polynomial and/or trigonometric nature to our input data. Furthermore, intensity measurements alone clearly allow for a non-linear transformation of our data, which may be sufficient to solve some classes of problems.

At this point one should note that the matrix $\mathbf{I_{out}}$ is the equivalent $\mathbf{H}$ matrix defined in 2.5. Having said this, it is worthwhile to study this matrix, particularly in terms of its rank. Indeed, the rank of a matrix will essentially tell us the dimensionality of the output space, which is of prime importance when evaluating the learning capabilities of an ELM, as it allows us to understand the data projection better. To this end, in the following subsections we will try to answer the question *What is the rank of the matrix* $\mathbf{I_{out}}$ *for a certain input encoding?*. We will answer this question quite pedagogically and start with a particular encoding scheme for phase modulation, then prove the general case for an arbitrary encoding with phase modulation. After, we prove the general case

*The full derivation can be seen in Appendix A.

for amplitude modulation, and finally analyse the rank of $\mathbf{I_{out}}$ for phase and amplitude simultaneously.

2.7.2 Phase modulation

We start by rewriting equation 2.20 with $|a_j^i| = 1 \; \forall j, i$

$$I_{out}^{il} = \sum_{j=1}^{K} |M_{lj}|^2 + \sum_{j=1}^{K} \sum_{p>j}^{K} C_{ljp} \left\{ \begin{array}{c} \xi_{ljp}^{e} \left[\cos(b_j^i)\cos(b_p^i) + \sin(b_j^i)\sin(b_p^i) \right] \\ -\xi_{ljp}^{o} \left[\sin(b_j^i)\cos(b_p^i) - \cos(b_j^i)\cos(b_p^i) \right] \end{array} \right\}$$

A particular case

Let $E_{in}^{il} = (\mathbf{e}_1^{in})_l + e^{ib_2^i}(\mathbf{e}_2^{in})_l + e^{ib_3^i}(\mathbf{e}_3^{in})_l$. The field $\mathbf{e}_1^{in}$ is intentionally not modulated, in order for a more intense interferometry within the complex media to take place. In that case, equation 2.21 unfolds to

$$\begin{aligned} I_{out}^{il} =& |M_{l1}|^2 + |M_{l2}|^2 + |M_{l3}|^2 \\ &+ C_{l12}\xi_{l12}^{e}\cos(b_2^i) + C_{l12}\xi_{l12}^{o}\sin(b_2^i) \\ &+ C_{l13}\xi_{l13}^{e}\cos(b_3^i) + C_{l13}\xi_{l13}^{o}\sin(b_3^i) \\ &+ C_{l23}\xi_{l23}^{e}\left[\cos(b_2^i)\cos(b_3^i) + \sin(b_2^i)\sin(b_3^i)\right] \\ &- C_{l23}\xi_{l23}^{o}\left[\sin(b_2^i)\cos(b_3^i) - \cos(b_2^i)\sin(b_3^i)\right] \end{aligned} \tag{2.21}$$

We now define the following quantities

$$\begin{aligned} z_0^l &= |M_{l1}|^2 + |M_{l2}|^2 + |M_{l3}|^2 & f_0^i &= 1 \\ z_1^l &= C_{l12}\xi_{l12}^{e} & f_1^i &= \cos(b_2^i) \\ z_2^l &= C_{l12}\xi_{l12}^{o} & f_2^i &= \sin(b_2^i) \\ z_3^l &= C_{l13}\xi_{l13}^{e} & f_3^i &= \cos(b_3^i) \\ z_4^l &= C_{l13}\xi_{l13}^{o} & f_4^i &= \sin(b_3^i) \\ z_5^l &= C_{l23}\xi_{l23}^{e} & f_5^i &= \begin{array}{c}\cos(b_2^i)\cos(b_3^i) \\ +\sin(b_2^i)\sin(b_3^i)\end{array} \\ z_6^l &= -C_{l23}\xi_{l23}^{o} & f_6^i &= \begin{array}{c}\sin(b_2^i)\cos(b_3^i) \\ -\cos(b_2^i)\sin(b_3^i)\end{array} \end{aligned} \tag{2.22}$$

With these definitions we can now write equation 2.21 as

$$I_{out}^{il} = z_0^l f_0^i + z_1^l f_1^i + z_2^l f_2^i + z_3^l f_3^i + z_4^l f_4^i + z_5^l f_5^i + z_6^l f_6^i \tag{2.23}$$

Consider the vectors $\mathbf{z}_i \in \mathbb{R}^{L\times 1}$ and $\mathbf{f}_i \in \mathbb{R}^{N\times 1}$. In that case, the matrix $\mathbf{I_{out}} \in \mathbb{R}^{N\times L}$ can be written as

$$\mathbf{I_{out}} = \sum_{n=0}^{6} \mathbf{f}_n \mathbf{z}_n^T \tag{2.24}$$

where $\mathbf{f}_n\mathbf{z}_n^T$ denotes the dyadic product between the two vectors. By writing $\mathbf{I_{out}}$ as in equation 2.24 we have expressed the matrix as a sum of matrices. By doing so, we can make use of the rank-sum inequality to affirm that [64]

$$\text{rank}\left(\mathbf{I_{out}}\right) \leq \sum_{n=0}^{6} \text{rank}\left(\mathbf{f}_n\mathbf{z}_n^T\right) \tag{2.25}$$

Nonetheless, we can say more than this since our matrices within the summation are dyadic products. Let us look closely at the line and column structures

$$\mathbf{f}_n\mathbf{z}_n^T = \begin{pmatrix} | & \dots & | & \dots & | \\ z_n^0\mathbf{f}_n & \dots & z_n^i\mathbf{f}_n & \dots & z_n^L\mathbf{f}_n \\ | & \dots & | & \dots & | \end{pmatrix} = \begin{pmatrix} — & f_n^0\mathbf{z}_n^T & — \\ \vdots & \vdots & \vdots \\ — & f_n^i\mathbf{z}_n^T & — \\ \vdots & \vdots & \vdots \\ — & f_n^N\mathbf{z}_n^T & — \end{pmatrix} \tag{2.26}$$

Note that when examining the lines, each line is the same vector $\mathbf{z}_n^T$ multiplied by a scalar. Likewise, in the columns, each column is $\mathbf{f}_n$ multiplied by a scalar. Thus, in either case there is only a line and column linearly independents. By definition, the rank of a matrix, A, is the dimension of the vector space generated by its columns, $\mathcal{C}(A)$. Coincidently, it can be shown that it is also the same dimension as the vector space generated by its rows, $\mathcal{R}(A)$. For a dyadic tensor, we can write

$$\mathcal{C}(\mathbf{f}_n\mathbf{z}_n^T) = \left\{\mathbf{v}_i \in \mathbb{R}^{N\times 1} : \mathbf{v}_i = c\mathbf{f}_n \forall c \in \mathbb{R}\right\} \tag{2.27}$$

$$\mathcal{R}(\mathbf{f}_n\mathbf{a}_n^T) = \left\{\mathbf{v}_i \in \mathbb{R}^{L\times 1} : \mathbf{v}_i = c\mathbf{z}_n \forall c \in \mathbb{R}\right\} \tag{2.28}$$

Which implies that $\dim(\mathcal{C}(\mathbf{f}_n\mathbf{z}_n^T)) = \dim(\mathcal{R}(\mathbf{f}_n\mathbf{z}_n^T)) = \text{rank}(\mathcal{C}(\mathbf{f}_n\mathbf{z}_n^T)) = 1$. Looking at 2.27 and 2.28, if $\mathbf{z}_n \neq \mathbf{z}_m \; \forall n \neq m$ and $\mathbf{f}_n \neq \mathbf{f}_m \; \forall n \neq m$, then $\mathcal{C}(\mathbf{f}_n\mathbf{z}_n^T) \cap \mathcal{C}(\mathbf{f}_m\mathbf{z}_m^T) = \{\mathbf{0}\}$ and $\mathcal{R}(\mathbf{f}_n\mathbf{z}_n^T) \cap \mathcal{R}(\mathbf{f}_m\mathbf{z}_m^T) = \{\mathbf{0}\} \; \forall n \neq m$. These conditions may seem highly restrictive at first glance, but it turns out to not be quite true. When examining the elements of the

vector $\mathbf{z}_n$, we see that they stem from the transmission matrix whose elements follow the statistics of a fully developed speckle, as studied in Appendix D. Thus the probability of $\mathbf{z}_n = \mathbf{z}_m$ is extremely small. As for $\mathbf{f}_n = \mathbf{f}_m$, it may be more easily met, but it can be overcome through clever encoding so as to not induce redundancy within the trigonometric functions. Assuming these conditions are met, equality on equation 2.25 holds* and we can write

$$\text{rank}(\mathbf{I_{out}}) = \sum_{n=0}^{6} \text{rank}(\mathbf{f}_n \mathbf{z}_n^T) = \sum_{n=0}^{6} 1 = 7 \tag{2.29}$$

With these tools, we are now ready to generalize to an arbitrary phase encoding scheme.

General case

For an easier reading, we rewrite equation 2.21

$$I_{out}^{il} = \sum_{j=1}^{K} |M_{lj}|^2 + \sum_{j=1}^{K} \sum_{p>j}^{K} C_{ljp} \left\{ \begin{array}{c} \xi_{ljp}^{e} \left[\cos(b_j^i)\cos(b_p^i) + \sin(b_j^i)\sin(b_p^i) \right] \\ -\xi_{ljp}^{o} \left[\sin(b_j^i)\cos(b_p^i) - \cos(b_j^i)\cos(b_p^i) \right] \end{array} \right\}$$

Let us now write

$$\gamma_0^l = \sum_{j=1}^{K} |M_{lj}|^2 \tag{2.30}$$

$$\beta_{jp}^l = C_{ljp}\xi_{ljp}^e \tag{2.31}$$

$$\alpha_{jp}^l = C_{ljp}\xi_{ljp}^o \tag{2.32}$$

$$f_0^i = 1 \tag{2.33}$$

$$g_{jp}^i = \cos(b_j^i)\cos(b_p^i) + \sin(b_j^i)\sin(b_p^i) \tag{2.34}$$

$$h_{jp}^i = \sin(b_j^i)\cos(b_p^i) - \cos(b_j^i)\cos(b_p^i) \tag{2.35}$$

Then, equation 2.21 reduces to

$$I_{out}^{il} = \gamma_0^l f_0^i + \sum_{j=1}^{K} \sum_{p>j}^{K} \beta_{jp}^l g_{jp}^i + \sum_{j=1}^{K} \sum_{p>j}^{K} \alpha_{jp}^l h_{jp}^i \tag{2.36}$$

Consider the index contraction outlined in table 2.1. With this new index, we can write:

$$I_{out}^{il} = \gamma_0^l f_0^i + \sum_{k=1}^{\frac{K(K-1)}{2}} \beta_k^l g_k^i + \sum_{k=1}^{\frac{K(K-1)}{2}} \alpha_k^l h_k^i \tag{2.37}$$

*See Appendix A for the mathematical proof of the equality on the rank-sum inequality

j	p	k
1	2	1
1	3	2
⋮	⋮	⋮
1	K	K-1
2	3	K
⋮	⋮	⋮
2	K	K-1+K-2
⋮	⋮	⋮
K-1	K	$\sum_{i=1}^{K-1} K-i = \frac{K(K-1)}{2}$

TABLE 2.1: Index contraction

In matrix form, it reads

$$I_{out} = \mathbf{f}_0 \boldsymbol{\gamma}_0^T + \sum_{k=1}^{\frac{K(K-1)}{2}} \mathbf{g}_k \boldsymbol{\beta}_k^T + \sum_{k=1}^{\frac{K(K-1)}{2}} \mathbf{h}_k \boldsymbol{\alpha}_k^T \tag{2.38}$$

The rank then is given by

$$\begin{aligned} \text{rank}(I_{out}) &= \text{rank}(\mathbf{f}_0 \boldsymbol{\gamma}_0^T) + \text{rank}\left(\sum_{k=1}^{\frac{K(K-1)}{2}} \mathbf{g}_k \boldsymbol{\beta}_k^T \right) + \text{rank}\left(\sum_{k=1}^{\frac{K(K-1)}{2}} \mathbf{h}_k \boldsymbol{\alpha}_k^T \right) \\ &= 1 + \frac{K(K-1)}{2} + \frac{K(K-1)}{2} \\ &= K^2 - K + 1 \end{aligned} \tag{2.39}$$

We have thus arrived at an expression which tells us the effective dimensionality of the output space of the data projection as a function of the number of input encoding fields (note that for K input fields, only $K-1$ were considered to carry information).

2.7.3 Amplitude modulation

We now carry on for the case of amplitude modulation and consider $f_j(\mathbf{x}_i) = a_i(\mathbf{x}_i)$ alone, and $a_1(\mathbf{x}_i) = 1$. In that case, equation 2.20 reduces to

$$I_{out}^{il} = \sum_{j=1}^{K} |a_j^i|^2 |M_{lj}|^2 + \sum_{j=1}^{K} \sum_{p>j}^{K} a_j^i a_p^i C_{ljp} \tilde{\varsigma}_{ljp}^e \tag{2.40}$$

Using the same kind of simplifications as before, we can write

$$I_{out}^{il} = \underbrace{|M_{l1}|^2}_{\Gamma^l} + \sum_{j=2}^{K} \underbrace{|a_j^i|^2}_{F_j^i} \underbrace{|M_{lj}|^2}_{\lambda_j^l} + \sum_{p=2}^{K} \underbrace{a_p^i}_{H_p^i} \underbrace{C_{l1p}\xi_{l1p}^e}_{\theta_p^l} + \sum_{j=2}^{K}\sum_{p>j}^{K} \underbrace{a_j^i a_p^i}_{G_{jp}^i} \underbrace{C_{ljp}\xi_{ljp}^e}_{\tau_{jp}^l}$$

$$\Rightarrow \mathbf{I_{out}} = 1\boldsymbol{\Gamma}^T + \sum_{j=2}^{K} \mathbf{F}_j \boldsymbol{\lambda}_j^T + \sum_{p=2}^{K} \mathbf{H}_p \boldsymbol{\theta}_p^T + \sum_{k=1}^{\frac{K(K-1)}{2}-K+1} \mathbf{G}_k \boldsymbol{\tau}_k^T \tag{2.41}$$

Within this notation, the rank can then be found as

$$\begin{aligned}\text{rank}(\mathbf{I_{out}}) &= \text{rank}(1\boldsymbol{\Gamma}^T) + \sum_{j=2}^{K} \text{rank}(\mathbf{F}_j \boldsymbol{\lambda}_j^T) + \sum_{p=2}^{K} \text{rank}(\mathbf{H}_p \boldsymbol{\theta}_p^T) + \sum_{k=1}^{\frac{K(K-1)}{2}-K+1} \text{rank}(\mathbf{G}_k \boldsymbol{\tau}_k^T) \\ &= 1 + K - 1 + K - 1 + \frac{K(K-1)}{2} - K + 1\end{aligned}$$

which can be simplified to

$$\text{rank}(\mathbf{I_{out}}) = \frac{K^2}{2} + \frac{K}{2} \tag{2.42}$$

2.7.4 Phase and amplitude modulation

For the sake of completeness, we'll go over the case where we allow for both amplitude and phase modulation simultaneously. We start again at equation 2.20

$$\begin{aligned}I_{out}^{il} &= \sum_{j=p=1}^{K} |a_j^i|^2 |M_{lj}|^2 \\ &+ \sum_{j=1}^{K}\sum_{p>j}^{K} \left\{ \begin{array}{l} \overbrace{\underbrace{C_{ljp}\xi_{ljp}^e}_{\gamma_{jp}^l} a_j^i a_p^i \left[\cos(b_j^i)\cos(b_p^i) + \sin(b_j^i)\sin(b_p^i)\right]}^{g_{jp}^i} \\ \underbrace{-C_{ljp}\xi_{ljp}^o}_{\beta_{jp}^l} \underbrace{a_j^i a_p^i \left[\sin(b_j^i)\cos(b_p^i) - \cos(b_j^i)\cos(b_p^i)\right]}_{h_{jp}^i} \end{array} \right\} \\ &= \underbrace{|M_{l1}|^2}_{\Gamma^l} + \sum_{j=2}^{K} \underbrace{|a_j|^2}_{h_j^i} \underbrace{|M_{lj}|^2}_{C_j^l} + \sum_{j=1}^{K}\sum_{p>j}^{K} g_{jp}^i \gamma_{jp}^l + \sum_{j=1}^{K}\sum_{p>j}^{K} h_{jp}^i \beta_{jp}^l \\ &= \Gamma_l + \sum_{j=2}^{K} f_j^i C_j^l + \sum_{k=1}^{\frac{K(K-1)}{2}} g_k^i \gamma_k^l + \sum_{k=1}^{\frac{K(K-1)}{2}} h_k^i \beta_k^l\end{aligned} \tag{2.43}$$

In matrix form, we have

$$\mathbf{I_{out}} = 1\boldsymbol{\Gamma}^T + \sum_{j=2}^{K} \mathbf{f}_j \mathbf{C}_j^T + \sum_{k=1}^{\frac{K(K-1)}{2}} \mathbf{g}_k \boldsymbol{\gamma}_k^T + \sum_{k=1}^{\frac{K(K-1)}{2}} \mathbf{h}_k \boldsymbol{\beta}_k^T \tag{2.44}$$

From which we can now write

$$\text{rank}(\mathbf{I_{out}}) = 1 + K - 1 + \frac{K(K-1)}{2} + \frac{K(K-1)}{2}$$

which simplifies to

$$\text{rank}(\mathbf{I_{out}}) = K^2 \tag{2.45}$$

We can now compare the different encoding mechanism, as in figure 2.6. We can see that the dimension of the output space scales quadratically with the number of encoding input fields. However, the fastest growing is when we allow for phase and amplitude modulation. Note also that the difference between phase and amplitude encodings combined and phase-only modulation is not as significant as the difference between phase-only and amplitude-only.

K	K-1 (Number of encoding input fields)	Rank(I_{out}) Phase encoding	Rank(I_{out}) Amplitude encoding	Rank(I_{out}) Phase and amplitude encoding
1	0	1	1	1
2	1	3	3	4
3	2	7	6	9
4	3	13	10	16
5	4	21	15	25
...	...	...	...	...

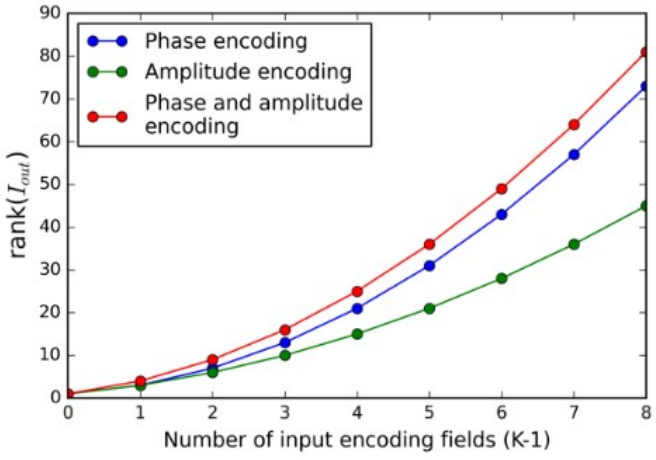

FIGURE 2.6: rank($\mathbf{I_{out}}$) as a function of the number of input encoding fields for the different encoding regimes.

In this chapter we have started by a general overview of the ML realm, and moved on

to the mathematical foundations of ELMs. We have seen that, despite their simplicity, they are anchored in strong theorems which guarantee their effectiveness. We have reviewed the state of the art regarding optical ELMs and we've seen that we are missing a more fundamental understanding of such machines. To that end, we have developed a theoretical model which allows us not only to understand the polynomial and sinusoidal nature of input projection, but we have also deduced simple laws which express how the output space dimensionality scales with the number of input encoding fields, demonstrating a quadratic dependence in each of the encoding schemes. In the next chapter, we are going to go over a set of numerical simulations which will allow us to corroborate our model, and after we will move on to an experimental implementation.

Chapter 3

A numerical simulation of an optical extreme learning machine

In this chapter, we aim to introduce some numerical simulations that will enable us to prove our theoretical framework in chapter 2. Particularly we wish to prove the rank scaling laws in equations 2.39, 2.42 and 2.45. We will study a method to simulate speckle patterns based on the works of Duncan and Kirkpatrick [65]. Then, we will extend this method to allow for phase and amplitude modulation of incoming light. Finally, we will study the output matrices for regression and classification tasks upon different encoding schemes, examine the machine's performance, and finally prove the rank scaling laws, as per equations 2.39, 2.42 and 2.45.

3.1 Speckle simulation

The approach follows closely the one employed by Goodman [62] when studying statistical properties of speckle patterns, as outlined in Appendix D. We start by assuming that the speckle pattern arises from a wavefront incident on a reflective rough surface. Due to the complex geometry of the surface, we can assume that the wavefront right after reflection, contains a fully randomised phase profile, as different parts of the wavefront will have travelled different lenghts. For this reason, let us assume that the field right after the reflection is given by

$$E_{in} = E(x,y)e^{i\varphi(x,y)} \tag{3.1}$$

where we will assume $E(x,y) = 1$ for simplicity and $\varphi(x,y)$ follows a uniform probability distribution in $[-\pi, \pi]$. Assume further that the incoming light is tightly constrained to

a circular aperture of diameter D, given by $D(x,y)$. The input field is then given by $E_{in} = D(x,y)e^{i\varphi(x,y)}$. We now calculate its fourier transform, E_{out}, and this output field is our speckle field. Performing a fourier tansform is equivalent to either placing the input field at the back focal plane of a convex lens and examine the front focal plane image, or letting the field propagate to very large distances and let diffraction take over (Fraunhofer regime). Numerically, we start with an $M \times M$ matrix, with a circular aperture at the center of diameter D, and within this aperture we include a uniformly sampled phase distribution. Then we let the Fast Fourier Transform algorithm perform the mapping. A typical result is shown in figure 3.1. The ratio M/D controls the speckle size, and for $M/D = 2$ the Nyquist criterion is met and the smallest speckle is two pixels.

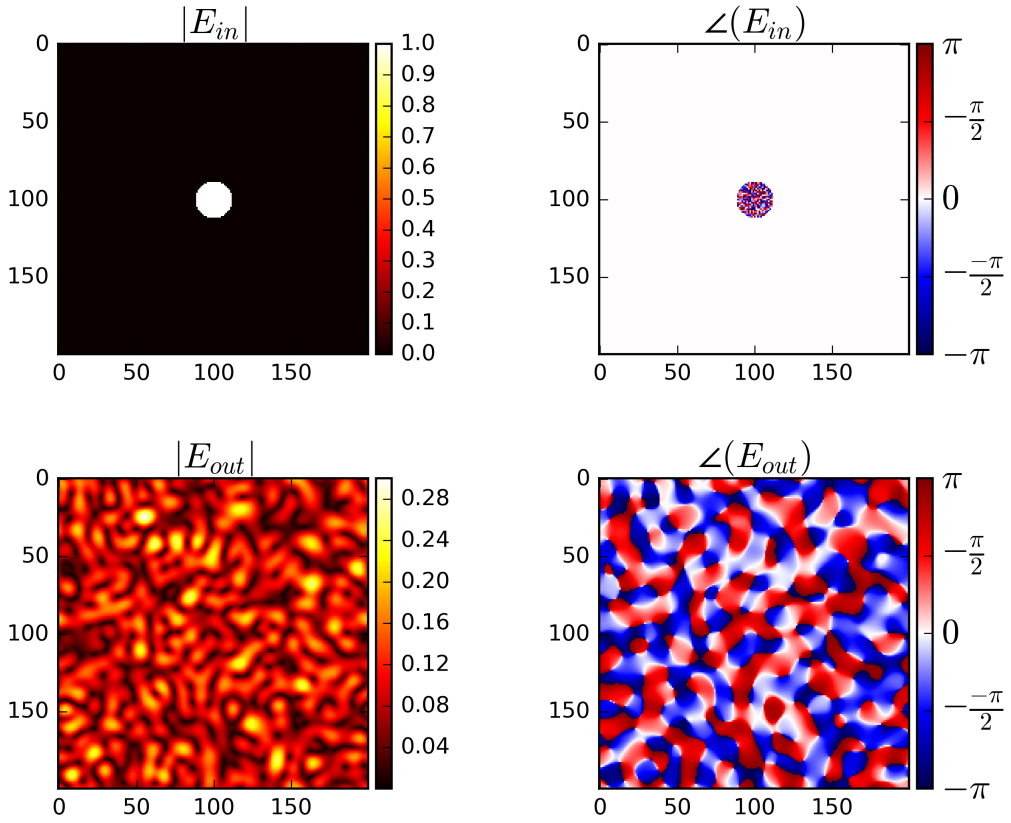

FIGURE 3.1: Typical relation between E_{in} and E_{out}, for $M = 200$ and $D = 25$.

This algorithm is capable of producing fully developed speckle patterns that follow the statistical properties outlined by Goodman [62], as can be seen in figure 3.2. Indeed, the electric field amplitude probability distribution function (PDF), $P(|E_{out}|)$, follows a Rayleigh distribution, whereas the phase PDF, $P(\angle E_{out})$, is essentially uniform and the intensity PDF, $P(I)$, is a decaying exponential. Furthermore, the circular gaussian statistics of the joint PDF of the imaginary and real parts of E_{out} also hold.

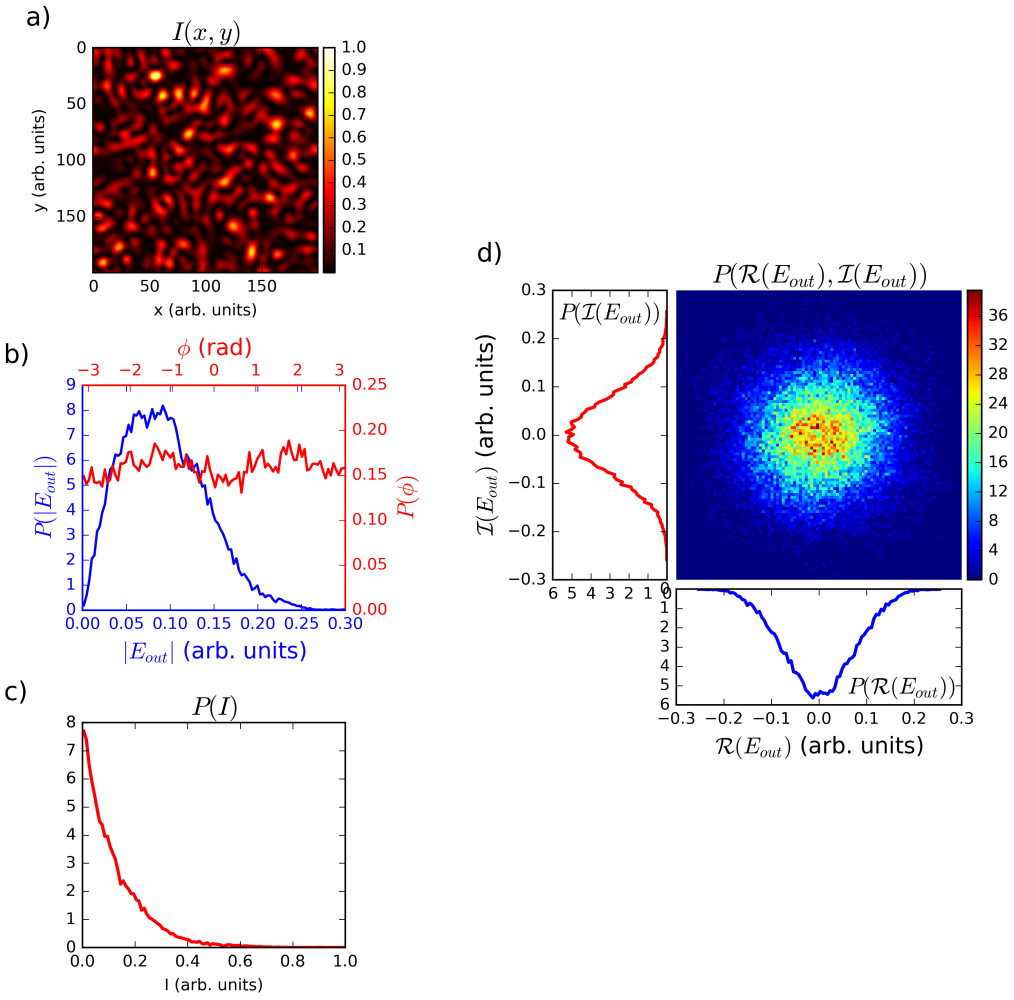

FIGURE 3.2: Speckle statistics for a speckle pattern with $M = 200$ and $D = 25$.

With this new tool, we aim now to extend this simulation to include information encoding within the input field. To this end, we introduce the functions $A(x,y)$ and $\phi(x,y)$ which are amplitude and phase masks, respectively, which will be responsible for the encoding. There is no alteration to the algorithm other than the fact that the input field is now given by $E_{in}(x,y) = A(x,y)D(x,y)\exp\left\{i\left[\varphi(x,y) - \phi(x,y)\right]\right\}$. This change does not alter the statistics of the speckle pattern, but does produce different speckle patterns upon changes on either $A(x,y)$ or $\phi(x,y)$. As can be seen in figure 3.3, we have chosen rectangular segments within the aperture to act as encoding regions.

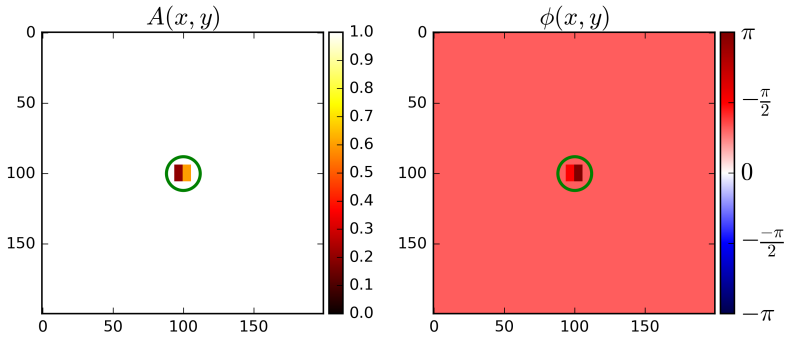

FIGURE 3.3: Encoding amplitude and phase masks. The green circle represents the aperture limits.

3.2 Rank of the outputs and learning capability

In this section, we aim to study the learning capability of our ELM with respect to the $rank(\mathbf{I_{out}})$. To do so, we will resort to the single value decomposition [66], and we will count the number of singular values above the noise threshold. We start with phase modulation, and employ the encoding scheme described previously, with detailed values in table 3.1. We test the machine in a regression and classification tasks. For the regression we use a Ridge model for training, while for classification we resort to a Logistic Regression model with an l_2 penalty.

	$K-1$	Saturation	ϕ_1	ϕ_2	ϕ_3	ϕ_4
Regression	1	No	$\frac{x-x_{min}}{x_{max}-x_{min}}2\pi$	0	0	0
	4	No	$\frac{x-x_{min}}{x_{max}-x_{min}}2\pi$	ϕ_1^2	$\phi_1^{0.5}$	$\phi_1^{0.25}$
	1	Yes	$\frac{x-x_{min}}{x_{max}-x_{min}}2\pi$	0	0	0
Classification	2	No	$\frac{x-x_{min}}{x_{max}-x_{min}}2\pi$	$\frac{y-y_{min}}{y_{max}-y_{min}}2\pi$	0	0
	4	No	$\frac{x-x_{min}}{x_{max}-x_{min}}2\pi$	$\frac{y-y_{min}}{y_{max}-y_{min}}2\pi$	ϕ_1^2	ϕ_2^2
	2	Yes	$\frac{x-x_{min}}{x_{max}-x_{min}}2\pi$	$\frac{y-y_{min}}{y_{max}-y_{min}}2\pi$	0	0

TABLE 3.1: Encoding schemes for phase modulation. The nomenclature $K-1$ follows the theoretical model developed in chapter 2, and it represents the number of encoding input fields.

The results are shown in figure 3.4. It is clear from the figure that the results improve as we examine the figure from top to bottom. We can, therefore, conclude that a higher rank of the outputs may indicate a greater performance, and vice-versa. This is intuitive due to the fact that the dimensionality of the output space is greater thus making the linear separability in higher dimensions more likely. Furthermore, the effect of the physical non-linearity is very noticeable as the machine correctly learned the spiral patterns in the

classification task, as well as the non-linear behaviour of the squared sinc function within the regression task.

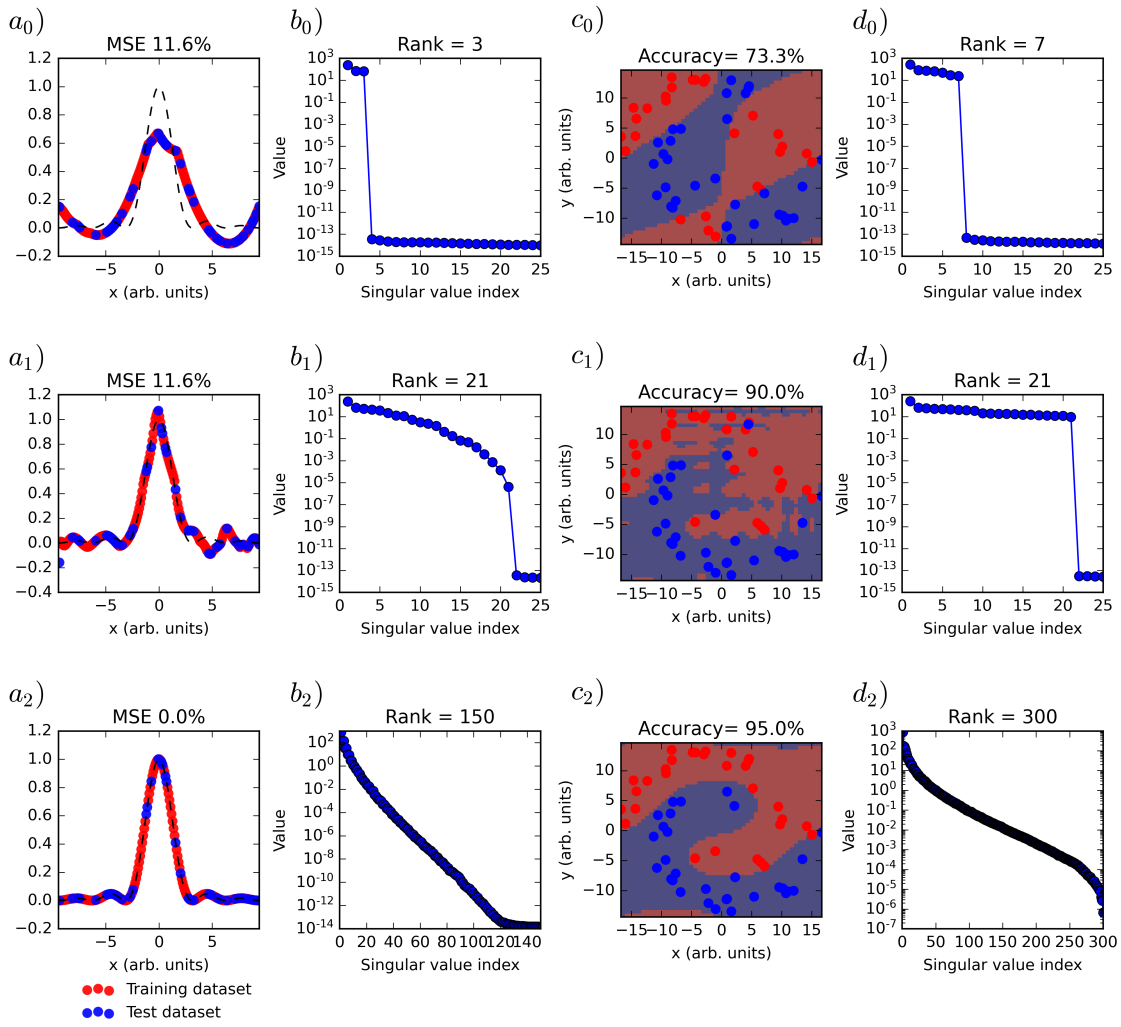

FIGURE 3.4: Numerical results of our proposed ELM in regression and classification tasks, with phase modulation. The columns $a)$ and $b)$ refer to the regression task on a nonlinear function, while columns $c)$ and $d)$ refer to the classification one on a spiral dataset. The lines correspond the different encoding schemes in regression and classification tasks, respectively, as outlined in table 3.1. In column $c)$ it is represented the classification performance in the test dataset, overlaid with a 50x50 grid to demonstrate the decision boundary.

We then follow for the amplitude modulation, and repeat the same study. The encoding schemes and results are shown in table 3.2 and figure 3.5. Our conclusions are the same, with the only relevant remark being the fact that the physical non-linearity has not produced such a significant improvement on the machine performance as we had seen with phase modulation.

	$K-1$	Saturation	ϕ_1	ϕ_2	ϕ_3	ϕ_4
Regression	1	No	$\frac{x-x_{min}}{x_{max}-x_{min}}$	1	1	1
	4	No	$\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_1^{2.17}$	$\phi_1^{0.52}$	$\phi_1^{0.27}$
	1	Yes	$\frac{x-x_{min}}{x_{max}-x_{min}}$	1	1	1
Classification	2	No	$\frac{x-x_{min}}{x_{max}-x_{min}}$	$\frac{y-y_{min}}{y_{max}-y_{min}}$	1	1
	4	No	$\frac{x-x_{min}}{x_{max}-x_{min}}$	$\frac{y-y_{min}}{y_{max}-y_{min}}$	$\phi_1^{2.27}$	$\phi_2^{2.27}$
	2	Yes	$\frac{x-x_{min}}{x_{max}-x_{min}}$	$\frac{y-y_{min}}{y_{max}-y_{min}}$	1	1

TABLE 3.2: Encoding schemes for amplitude modulation.

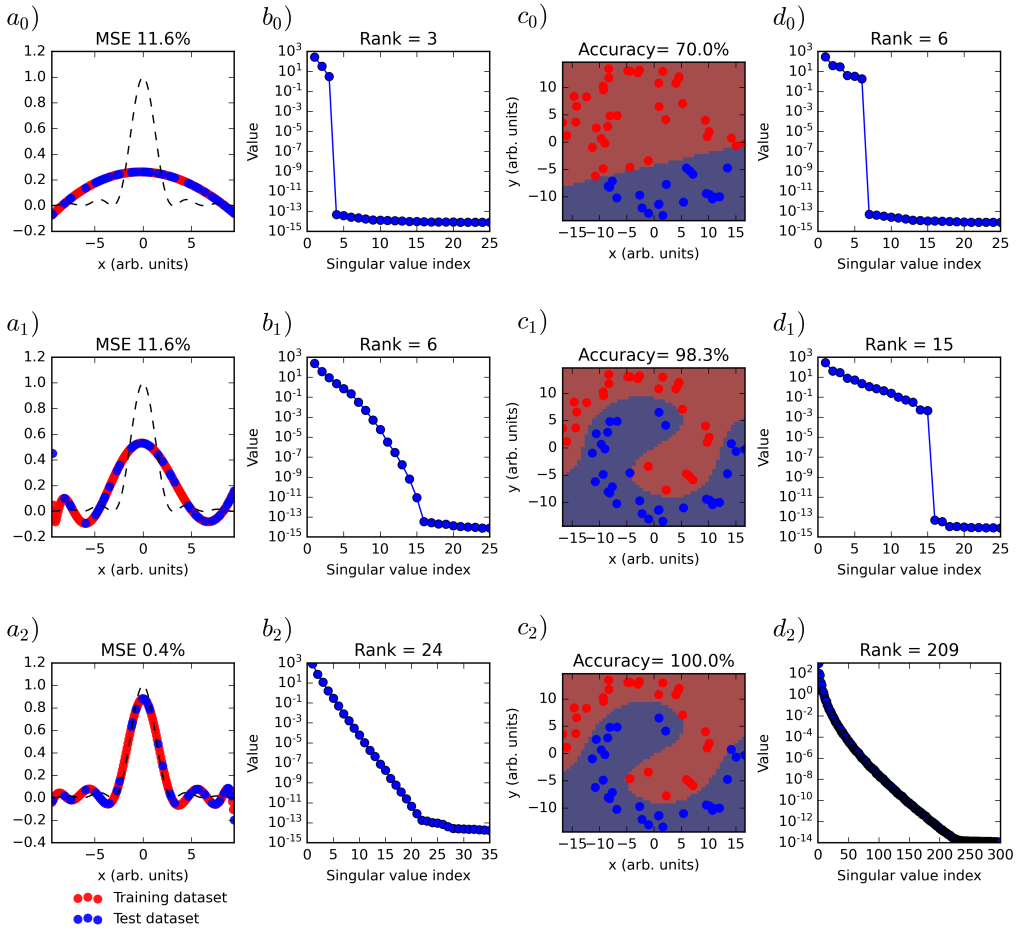

FIGURE 3.5: Numerical results of our proposed ELM in regression and classification tasks, with amplitude modulation. The columns a) and b) refer to the regression task on a nonlinear function, while columns c) and d) refer to the classification one on a spiral dataset. The lines correspond the different encoding schemes in regression and classification tasks, respectively, as outlined in table 3.2. In column c) it is represented the classification performance in the test dataset, overlaid with a 50x50 grid to demonstrate the decision boundary.

3.3 Rank scaling

We now wish to prove the rank scaling laws defined in equations 2.39, 2.42 and 2.45. To this end, we define rectangular regions in the aperture area, as outlined in the simulation algorithm above, and in each area we will:

1. allow phase modulation where each region will be modulated as ϕ_1^n where n will take the values from the $K-1$ possible values evenly sampled from -1 to +1, and $\phi_1 = \frac{x-x_{min}}{x_{max}-x_{min}} 2\pi$;
2. allow amplitude modulation where each region will be modulated as ϕ_1^n where n will take the values from the $K-1$ possible values evenly sampled from -1 to +1, and $\phi_1 = \frac{x-x_{min}}{x_{max}-x_{min}}$;
3. allow phase and amplitude modulation simultaneously where each region will be modulated as in the previous points.

The results are shown in figure 3.6, and we can see that the scaling laws hold for the three different scenarios. The last point seems to deviate from the theoretical curve, however, that is due to the difficulty in evaluating the rank from the singular value decomposition, as the distinction from the noise associated values is very difficult.

In this chapter we have developed a set of numerical simulations based on speckle generation, which allows us to emulate an experimental implementation of our ELM. We have analysed the performance of the machine upon different scenarios and we've concluded that it can perform quite well. Furthermore, we have given evidence of the rank scaling laws. The next step is to provide experimental proof of the same concept.

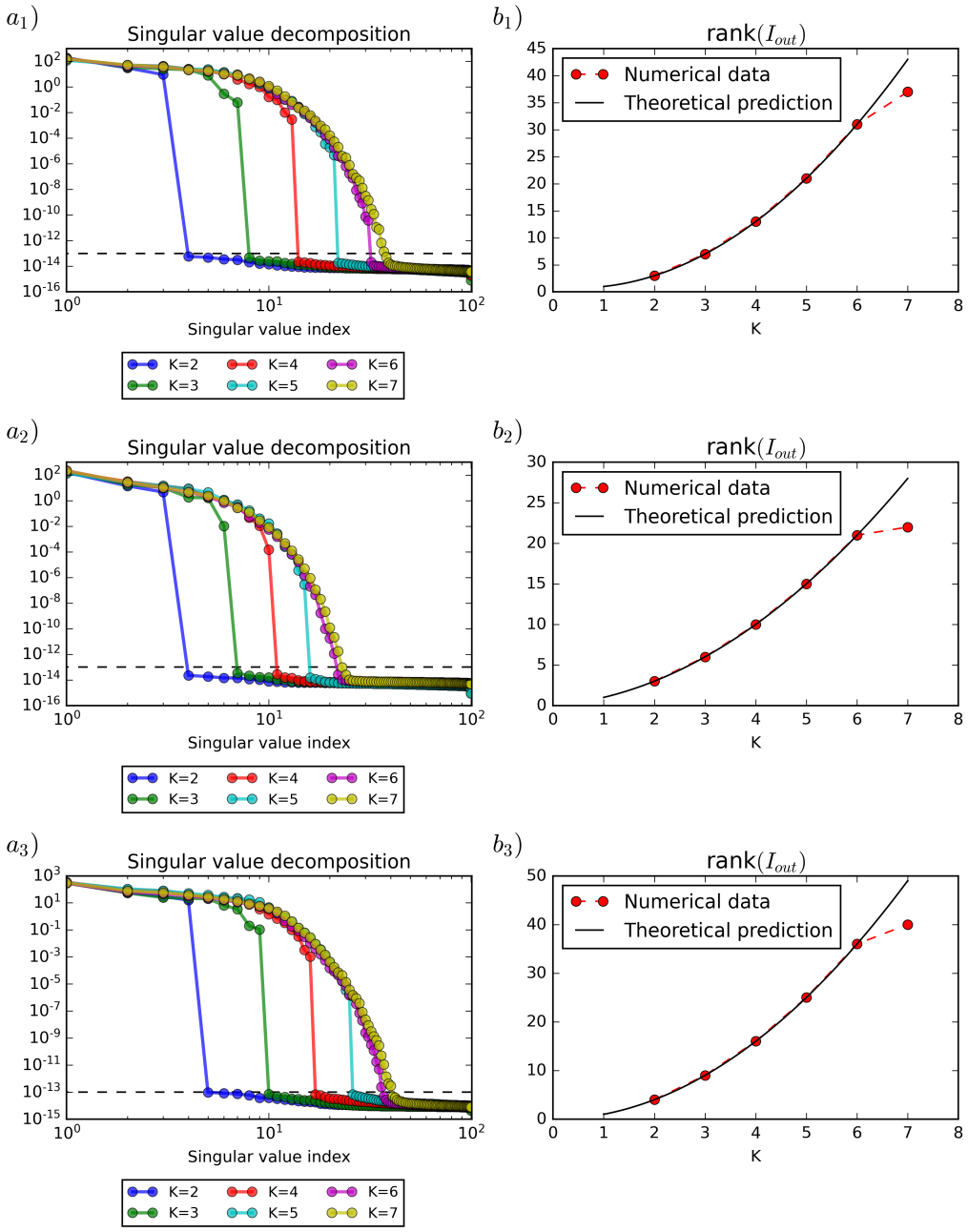

FIGURE 3.6: Numerical results on the rank scaling laws. The lines a,b and c correspond to the cases outlined in the main text of phase, amplitude and simultaneous phase and amplitude modulation schemes. In column a), the black dashed line represents the threshold value upon which the rank of the matrix was evaluated.

Chapter 4

Experimental methods and equipment

In this chapter, we aim to outline the experimental methods and equipment that were used throughout our work. We start by introducing the spatial light modulator used for the optical encoding. Then, we outline some experimental details for setting up the device, and finally we introduce techniques to achieve phase and amplitude modulation. Afterwards, we introduce our detector array as well as the major software tools used in our work.

4.1 Spatial light modulator: Digital micromirror device (DMD)

In our experiments we use a digital micromirror device (DMD) to encode information on the incident wavefront. A DMD screen consists of an array of micromirrors where each pixel can be either on the "on" state or the "off" state. Each state is characterised by the tilt angle of each individual pixel of +12° or -12° along the axis of rotation, typically set to a 45° angle with respect to the horizontal and vertical dimensions of the pixel. The model we've used is a Vialux V-7000 Hi-Speed module, featuring a Discovery 4100 DLP chipset [67]. The device is controlled via USB 2.0 through the proprietary ALP-4.2 firmware and software. Through the combination of FPGA logic and on-board RAM, the device can achieve up to 22 727 Hz array switching rates for binary images, while enabling storage of up to 43 690 binary images. This combination is ideal for our purposes as it can be necessary numerous training samples. The DLP chip also features an input and output connector used for external device synchronisation, including input triggering and 4 controllable

output channels for dynamic synchronisation. As for the display, it has a resolution of 1024x768 pixels, with a micromirror pitch of 13.7μm, and a large spectral range covering wavelengths from 363nm (UV) up to 2500nm (NIR).

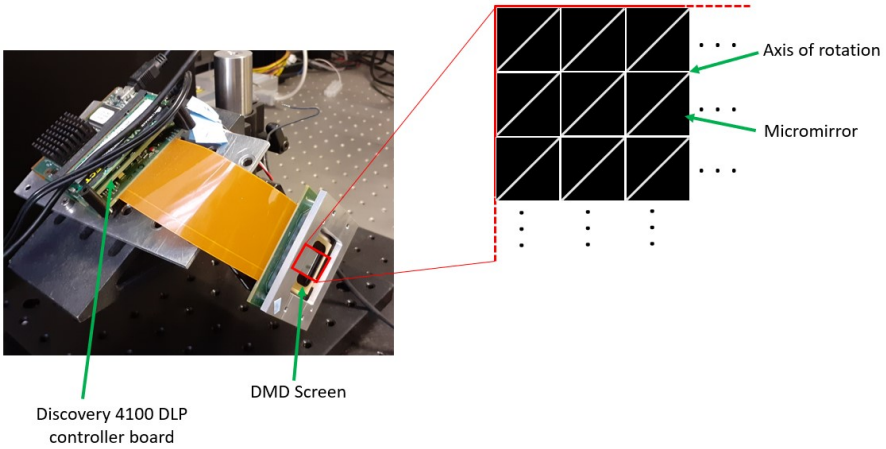

FIGURE 4.1: DMD Vialux V-7000 Hi-Speed module, and experimental set-up.

4.1.1 Setting up the DMD

When setting up a DMD one has to consider a few experimental details to achieve optimal wavefront control. The most obvious effect that will come up is diffraction. Due to the periodicity of the micromirrors, the device will act as a diffraction grating and numerous orders will crop up. Furthermore, the pixels have a tilt angle, thus the screen acts as a blazed grating at 12°. In order to minimise the diffraction losses, one should make use of the zeroth order of diffraction. For this reason, the input beam should either be normal to the DMD screen, as in figure 4.2, and the remainder of the optical set-up should follow the $\pm 12^o$ direction, or the incident beam can make a $\pm 12^o$ angle with the screen and the output can be collected normally to the screen. It's important to note that many diffraction orders will appear, and to avoid unwanted interference along the optical path, a spatial filtering stage should be employed after the DMD.

Another practical aspect is the orientation of the axis of rotation, as illustrated in figure 4.1. As the axis is at a 45° angle, the entire screen should be aligned such that the axis of rotation stays perpendicular to the optical table. Furthermore, if the device is to be used in precise wavefront shaping experiments, one should be careful with aberrations introduced by the micromirrors that may arise from fabrication defects. It is possible to compensate such effects through phase modulation, as outlined in section 4.1.2, and the use of zernike polynomials, however, in our case that proved itself an unnecessary effort

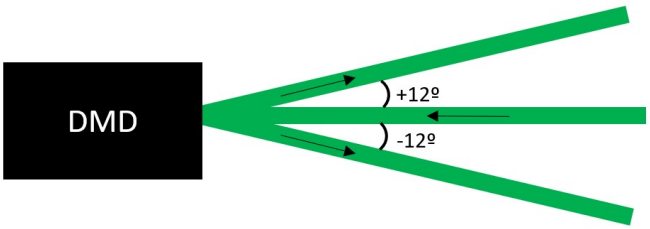

FIGURE 4.2: Alignment of a DMD.

and was not explored. Finally, it was verified experimentally that reflection on the DMD exhibited some effects commonly associated with birrefringence, that is, input linear polarisation resulted in elliptically polarised output light. The mirrors themselves should not exhibit such effect, to which we attribute the nature of such phenomena to the plastic encasing of the screen since unbalanced mechanical stress may induce birrefringence.

4.1.2 Phase modulation: Lee holography

A DMD is a binary spatial light modulator. Each pixel is a small mirror which can be in the "on" or "off" state, which is controlled by a tilt angle of $\pm 12^{o}$. While this device is an amplitude modulator, we can make use of light diffraction to achieve phase modulation. Let us suppose that we want to modulate an incoming beam travelling in the z direction with a transverse phase mask $\phi(x,y)$. Consider then the function $f(x,y)$

$$f(x,y) = \frac{1}{2}\left(1 + \cos(\mathbf{r}\cdot\boldsymbol{\nu} - \phi(x,y))\right) \tag{4.1}$$

where $\boldsymbol{\nu} = (\nu_x, \nu_y, 0)^T$. Now, let a plane wave be modulated by equation 4.1

$$\begin{aligned}
\mathbf{E}_{\text{out}} &= \mathbf{E}_0 f(x,y) e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t)} \\
&= \mathbf{E}_0 \frac{1}{4}\left(2 + e^{i(\mathbf{r}\cdot\boldsymbol{\nu}-\phi(x,y))} + e^{-i(\mathbf{r}\cdot\boldsymbol{\nu}-\phi(x,y))}\right) e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t)} \\
&= \frac{\mathbf{E}_0}{4}\left[\underbrace{2e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t)}}_{\text{Order 0}} + \underbrace{e^{i((\mathbf{k}+\boldsymbol{\nu})\cdot\mathbf{r}-\omega t-\phi(x,y))}}_{\text{Order +1}} + \underbrace{e^{i((\mathbf{k}-\boldsymbol{\nu})\cdot\mathbf{r}-\omega t+\phi(x,y))}}_{\text{Order -1}}\right]
\end{aligned} \tag{4.2}$$

From equation 4.2 we can see that modulation by $f(x,y)$ gives rise to two new waves that are modulated in phase by a certain mask $\phi(x,y)$. Now, all that is left to do is make this

suitable for a DMD through a binary quantisation process such as

$$g(x,y) = \begin{cases} 1 & , f(x,y) \geq \frac{1}{2} \\ 0 & , f(x,y) < \frac{1}{2} \end{cases} \tag{4.3}$$

This technique is called Lee holography [68]. It is important to note that the quantization process in equation 4.3 may give rise to ambiguities, thus there's a need to carefully choose ν_x and ν_y. We've found that by choosing $\nu_x \neq \nu_y$ and each a non-integer value, highly reduces the ambiguity for different phase values. Particularly, we've chosen $\nu_x = 2\pi \times \frac{1}{5.5}$ and $\nu_y = 2\pi \times \frac{1}{5.125}$.

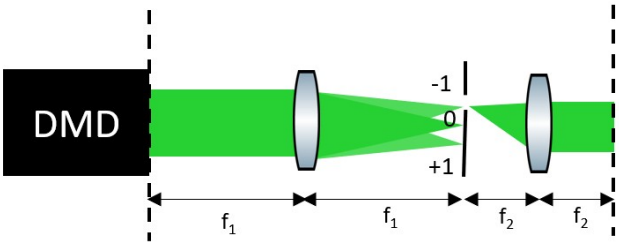

FIGURE 4.3: Example of spatial filtering stage within a 4f imaging system to achieve phase modulation through Lee holography in the zeroth order of diffraction.

4.1.3 Amplitude modulation

The DMD is a binary amplitude modulation device, therefore, to achieve more levels of amplitude modulation, one has to employ clever techniques. Our device has a built-in option of displaying 8-bit images through *Pulse Width Modulation*. This method achieves a grayscale pattern by modulating the duration of a sequence of impulses. However, for device synchronisation purposes, this was not suitable, and needed a way to achieve discrete amplitude modulation within a single frame. Our approach is depicted in figure 4.4. A single macropixel of the modulation area has several sub-macropixel which can either be on or off. These in turn may consist of agglomerates of individual DMD pixels. The sub-macropixels that are turned on are randomly chosen so as to avoid grating effects.

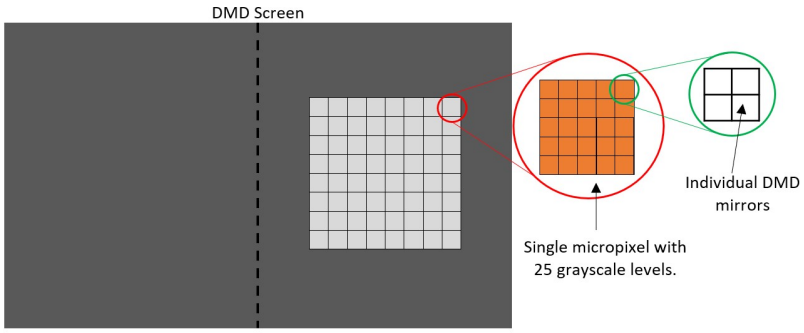

FIGURE 4.4: Illustration of the amplitude modulation.

4.2 Detector array: XIMEA MQ013MG

Our digital camera is a XIMEA MQ013MG-ON [69]. The device has a screen resolution of 1280x1024 pixels, with a pixel pitch of 4.8μm, and is capable of global shuttering, which is ideal in our case. It's capable of frame rates up to 500fps with a resolution of 640x512 pixels, with an 8 bit pixel depth. Nonetheless, it also supports 10 bit pixel depth. The machine interfaces with a computer via USB 3.0 which allows for high capacity data transfer, thus allowing for higher frame rates. It also features an I/O interface compatible with external triggering. The achievable frame rate is highly dependent on many factors, namely the API, data transfer time, USB connection, data processing time in the API, region of interest in the physical sensor and exposure time. Notice also that a high intensity in the laser may allow for a higher frame rate, as the camera needs a smaller integration time, however, the image is more susceptible to random noise.

FIGURE 4.5: XIMEA XiQ MQ013MG-ON. Image taken from [69]

4.3 Software

Throughout our work, we've made heavy use of the Python programming language [70] for various computational tasks. While there are several reasons to have chosen this tool, the main one lies with the extensive previous know-how within our research group which allied with Python's versatility for data manipulation and visualisation, as well as it's speed of computation, make it an obvious choice. Furthermore, for our experiments there were two hardware devices to be controlled: the detector array and the DMD. The first one has an official Python API offered by XIMEA [69]. For the former we've made use of an open-source python package developed by Popoff et al. [71]. On top of this, we intend to dive in machine learning algorithms, for which Python is highly popular, having many third-party libraries which allow the implementation of such algorithms to be done seamlessly. Finally, we make use of Jupyter Lab [72]: a web-based interactive development environment for notebooks, code, and data. With all these tools, we are able to create an ideal prototyping environment with the necessary tools for hardware control, data manipulation, analysis and visualisation.

Chapter 5

Experimental implementation of an optical ELM

In this chapter we demonstrate our implementation of an optical extreme learning machine. We give experimental proof of our theoretical framework developed in chapter 2, and benchmark our system in standard regression and classification tasks. We employ both phase and amplitude modulation schemes, as outlined in chapter 4. We also evaluate the machines' learning capability with respect to the effective dimensionality of the output space, as well as to the effect of a strong physical non-linearity (electronic saturation). We arrive at the same conclusions as our numerical simulations, thus validating our approach. We show that a higher rank may lead to greater performance, though it is highly dependent on the nature of the data projection functions, as well as the experimental noise.

5.1 Phase modulation

For our experimental implementation of an optical ELM with phase encoding, we've assembled the set-up illustrated in figure 5.1. It features a 50mW laser at 532nm, expanded by a simple converging lens. The beam then reflects off the DMD and follows to a spatial filtering stage, specifically mounted for the Lee holography technique (see chapter 4 for details). Together, they allow to encode information within the phase profile of the incident wavefront. Notice that the beam has not been collimated nor has it passed through a spatial filter (be it a pinhole or a single mode fibre) to achieve a gaussian mode. We've chosen to do so since for our purposes it is irrelevant the spatial "quality" of the beam,

since any irregularity within it can be captured by the transmission matrix formalism due to the linear dynamics of the set-up. Light is then coupled to a multimode fibre and, at its output, we've placed a 10x objective to do the imaging and finally use a cMOS camera for the detection. The fibre used features a $50\mu m$ core, which for a silica step-index fibre amounts to a V number of 50.4, with numerical aperture of 0.171 for a 532nm wavelength*. For such V number and for a step-index fibre, this amounts to a total number of modes of $M = \frac{4}{\pi^2}V^2 \approx 2540$. The output is a speckle pattern which is imaged through a 10x objective onto a cMOS camera.

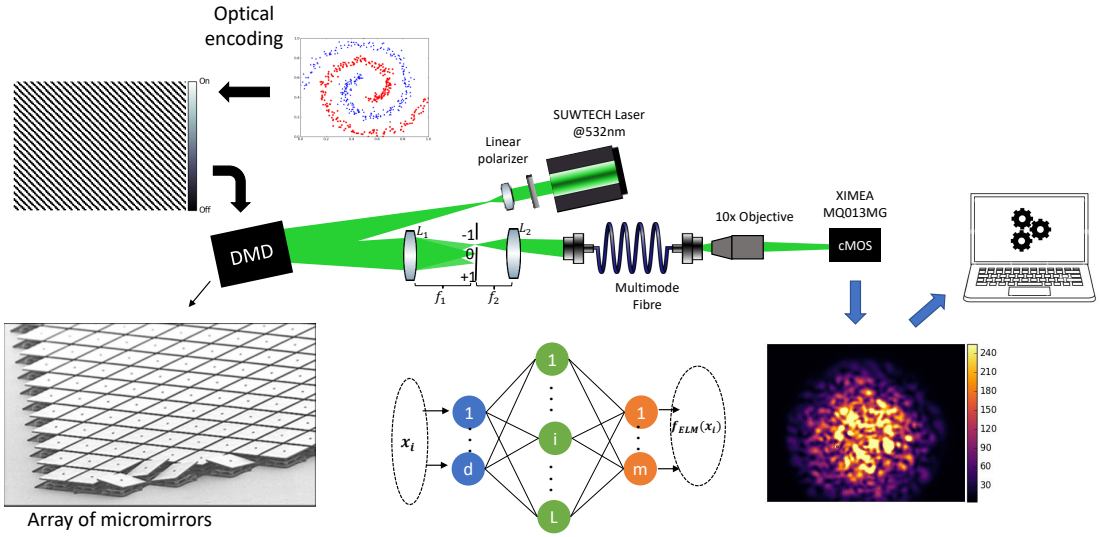

FIGURE 5.1: Illustration of the experimental set-up used for phase modulation with Lee holography.

Through holographic methods, we can imprint any phase mask $\phi(x, y)$. In order to test the theoretical analysis developed in chapter 2 we've chosen a one dimensional dataset as the non-linear function $f(x) = \left(\frac{\sin(x)}{x}\right)^2$ for a regression task, and the spiral dataset in figure 5.2b) for a classification task. In figure 5.2 we have inside the dashed red box the different phase masks $\phi(x, y)$ used for each task and respective encoding schemes. The different values of ϕ are expressed in tables 5.1 and 5.2.

According to our framework in chapter 2 we expect that the rank of the matrix $\mathbf{H}$, as in equation 2.9, to follow table 5.3†.

*Numbers obtained at `www.rp-photonics.com/v_number.html` . The fibre used has the inscription "optical cable 200807 50/125 LSZH 3660M ϕ3.0".

†Note that in encoding scheme a2), despite the fact that we have two input encoding fields, they generate redundancy within the trigonometric functions. However, after some algebra it can be shown that we predict rank 5 as in table 5.3

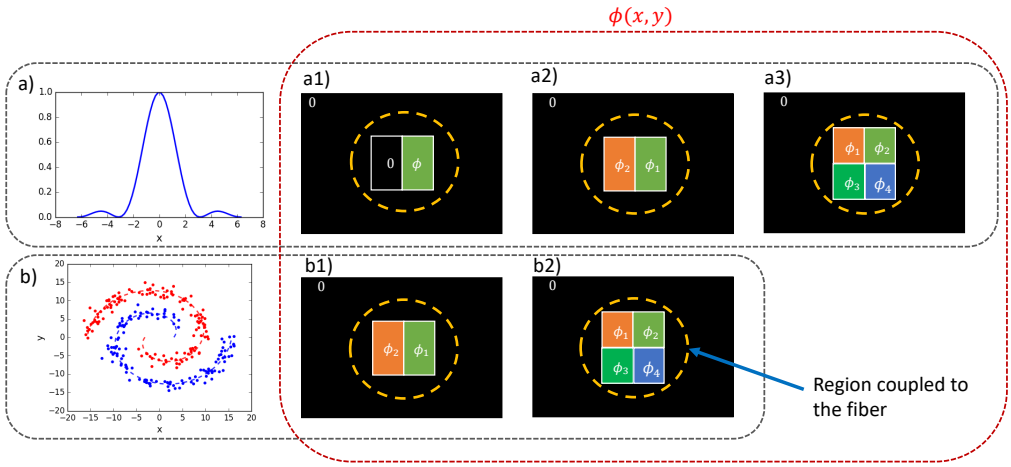

FIGURE 5.2: Illustration of the employed encoding schemes for phase modulation.

a1)	a2)	a3)	
$\phi = 2\pi\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_1 = 2\pi\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_1 = 2\pi\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_2 = \left(2\pi\frac{x-x_{min}}{x_{max}-x_{min}}\right)^2$
	$\phi_2 = 2\pi - 2\pi\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_3 = \left(2\pi\frac{x-x_{min}}{x_{max}-x_{min}}\right)^{0.5}$	$\phi_4 = \left(2\pi\frac{x-x_{min}}{x_{max}-x_{min}}\right)^4$

TABLE 5.1: Phase modulation encodings for the regression task according to figure 5.2.

b1)	b2)	
$\phi_1 = 2\pi\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_1 = 2\pi\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_3 = 2\pi\left(\frac{x-x_{min}}{x_{max}-x_{min}}\right)^2$
$\phi_2 = 2\pi\frac{y-y_{min}}{y_{max}-y_{min}}$	$\phi_2 = 2\pi\frac{y-y_{min}}{y_{max}-y_{min}}$	$\phi_2 = 2\pi\left(\frac{y-y_{min}}{y_{max}-y_{min}}\right)^2$

TABLE 5.2: Phase modulation encodings for the classification task according to figure 5.2.

	$\phi(x,y)$				
	a1)	a2)	a3)	b1)	b2)
rank(**H**)	3	5	21	7	21

TABLE 5.3: Prediction of the rank of the matrix **H** according to our model for phase modulation.

5.1.1 Results and discussion

In our experiment, we've used a region of 300x300 pixels on the DMD for modulation, and chose $\nu_x = 2\pi \times \frac{1}{5.5}$ and $\nu_y = 2\pi \times \frac{1}{5.125}$ for equation 4.1. For the regression task, we've used a total dataset of 64 samples in order to demonstrate both the rank of **H** as for the regression performance. For the classification task, we've used 32 samples per class to demonstrate the rank of **H**, and a dataset of 128 samples per class for training. In either

case, the training-test split followed a ratio 80%-20%. For the training we've used the popular python package *scikit-learn* [61], both for classification and regression. For regression tasks we've used a Ridge Regression model, as per equation 2.9, while for classification we've used a Logistic Regression model, as per equation 2.14. The reason for the choice of a small training set when demonstrating the predictions of rank($\mathbf{H}$), is experimental noise. Indeed, we noticed that when acquiring consecutive datasets, with the same samples, there was an average relative deviation to the first dataset of 5% +/- 4%. This noise is suspected to have origin in some or all of the following: the stability of the laser source, the DMD itself through thermal effects, stray light, shot noise in the digital camera and mechanical vibrations in the DMD and multimode fibre. If the number of samples were too high, the effect of two neighbouring samples on the output would be indistinguishable from random noise. Finally, each image collected on the camera has been downscaled through a local mean function such that the resolution went from 640x512 to 80x64. By doing so, we'll save time on the digital computations and the downscale method is compatible with a hardware implementation through large area photodetectors. For each encoding scheme, we've collected 10 datasets $\left\{\mathbf{H}_i^{exp}\right\}_{i=1}^{M=10}$. Finally, to avoid overfitting issues and better evaluate the generalisation capability of the model, all the benchmarks that follow have been tested on all 10 datasets.

To better evaluate the rank of the matrix $\mathbf{H}$, let us model each experimental $\mathbf{H}_i^{exp}$ as

$$\mathbf{H}_i^{exp} = \mathbf{H} + \mathbf{N}_i \tag{5.1}$$

Where $\mathbf{N}_i$ is some random matrix. Our best approximation of $\mathbf{H}$ is

$$\langle \mathbf{H}^{exp} \rangle = \frac{1}{M} \sum_{i=1}^{M=10} \mathbf{H}_i^{exp} \tag{5.2}$$

In that case, we can approximate the experimental noise as

$$\mathbf{N}_i^{exp} = |\mathbf{H}_i^{exp} - \langle \mathbf{H}^{exp} \rangle| \tag{5.3}$$

Since the rank of a matrix is reflected on the number of non-zero singular values, we can then approximate rank($\mathbf{H}$) $\approx$ rank($\langle \mathbf{H}^{exp} \rangle$). However, $\langle \mathbf{H}^{exp} \rangle$ is a mere approximation, thus we need a better criterion to count the number of relevant singular values. To do so, we resort to Weyl's inequality for singular values [73]

$$|\sigma_k(\mathbf{H} + \mathbf{N}_i) - \sigma_k(\mathbf{H})| \leq \sigma_1(\mathbf{N}_i) \tag{5.4}$$

where $\sigma_k(M)$ is the singular value k of a matrix M, and $\sigma_1(\mathbf{N}_i)$ is the highest singular value of the noise matrix. Looking at equation 5.4 we see that if $\text{rank}(\mathbf{H}) = r$ then there will be r singular values above the highest singular value of the noise matrix.

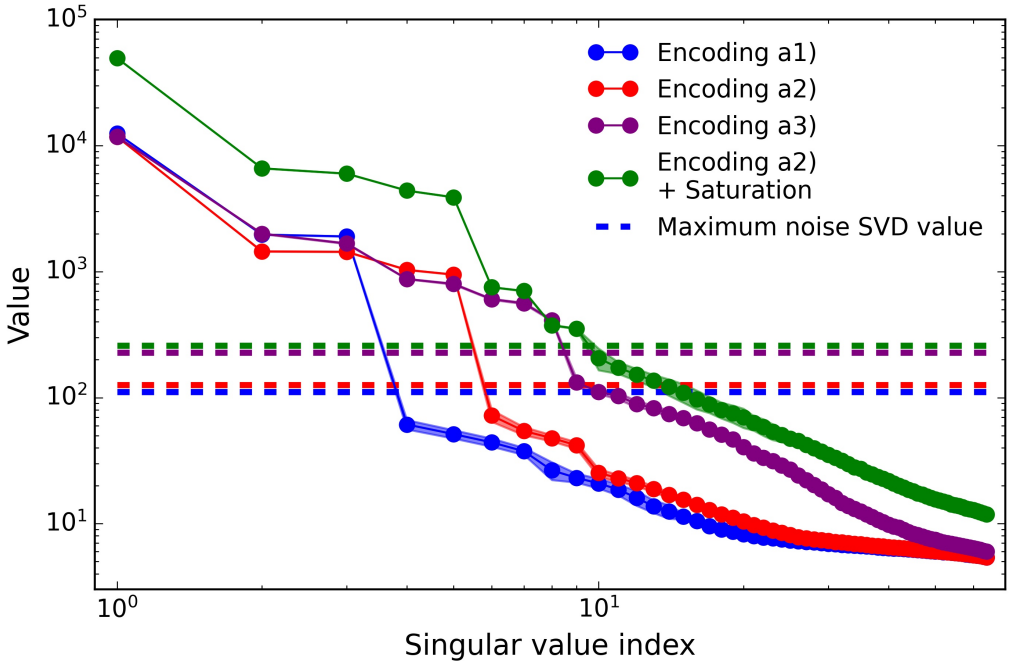

FIGURE 5.3: Singular value decomposition for the different encoding schemes within phase modulation. The dashed lines represent the highest singular value from the noise matrix for each encoding scheme. The solid and dashed lines are colour matched, as well as with figure 5.4

The results and performance benchmarks are shown in figure 5.3 and 5.4, and are summarised in table 5.4. From figure 5.3 we see that for encoding schemes a1) and a2) the prediction on the number of singular values above the noise level (i.e. the highest singular value of the noise matrix) matches exactly our prediction in table 5.3. The non-linear encoding, a3), on the other hand, does not. Here, we predict that the reason as to why this happens is that the corresponding modulation on equation 2.20 to such singular values is much smaller than the others, making the corresponding contributions indistinguishable from noise. In fact, this scheme is the one that shows the smallest contrast with the singular values regarded as noise. As predicted, the effect of the saturation is to increase the amount of information that is carried out to $\mathbf{H}$. It is interesting to note that even though the encoding scheme a3) and the a2) with saturation present a similar number of singular values above the noise threshold, their respective performance are the worst and best. However, there is a performance increase when going from scheme a1) to a2), which is also an increase in the rank of the matrix $\mathbf{H}$.

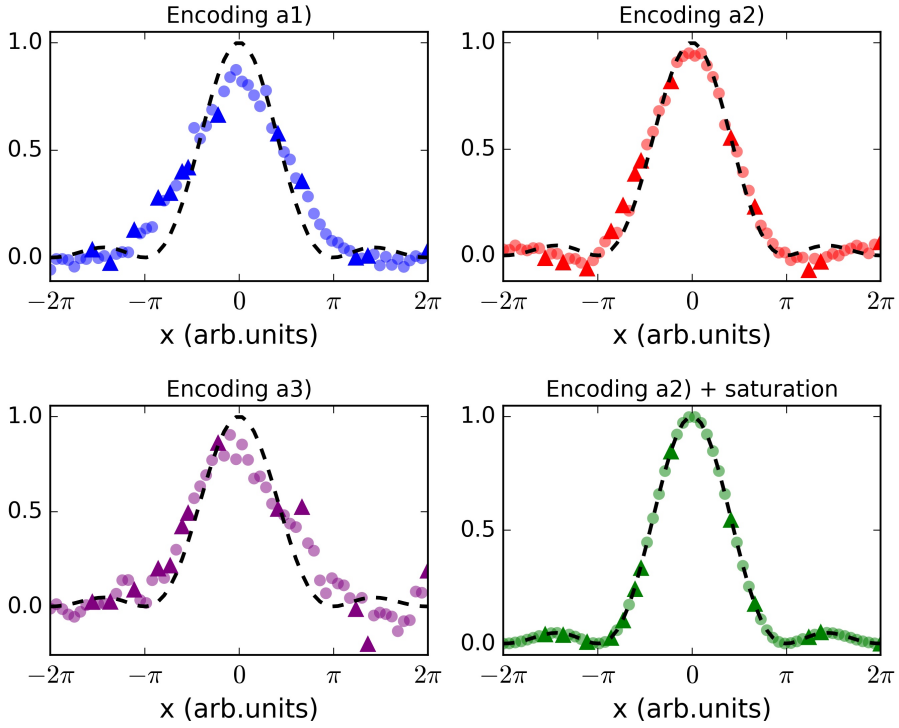

FIGURE 5.4: Regression performance of the machine on the training set (semi-transparent circles) and test set (solid triangles), for the different encoding schemes. The colours are matched with figure 5.3.

Encoding scheme	rank(H) (Prediction)	rank(H) (Experimental)	Number of input features	RMSE (%)
a1)	3	3	$\{b_1\}$	1.712
a2)	5	5	$\{b_1, 2\pi - b_1\}$	0.697
a3)	21	8	$\{b_1, b_1^{0.5}, b_1^2, b_1^4\}$	2.510
a2) + saturation	-	9	$\{b_1, 2\pi - b_1\}$	0.002

TABLE 5.4: Experimental results on the regression task.

Thus, we conclude that a higher rank(**H**) **may** indicate a better performance, however, it is not an ideal metric. The reason for that lies in the nature of the projection as evidenced by equation 2.20. Making a more complex encoding does in fact increase the complexity of the projection, however it will always have a sinusoidal and polynomial nature. Admittedly, such basis of functions can be quite powerful, but the outcome from a physical non-linearity, as is the case of saturation, will fundamentally modify the basis of the projection. With this in mind, the reason why the saturation has worked so well may lie in the alteration of the basis functions to perform the mapping from the input electric field to the intensity of the speckle pattern, in a way that such functions make the regression of $f(x)$ more tractable, all the while keeping the rank of the outputs the same. Nonetheless, within the saturation regime, the detection is also less prone to noise, which

has aided the performance further.

Having benchmarked the machine on a regression task we go on to a classification task. As a first challenge we choose the task of distinguishing between the classes represented by the blue and red dots in figure 5.5. We have chosen this dataset because it's circular symmetry should allow for an easy task for our machine due to the sinusoidal nature of the projection. Alas, our set-up has been able to solve this problem quite remarkably as can be seen from figure 5.7. In this case, even though the effect of saturation and a non-linear encoding have increased the rank of $\mathbf{H}$, the performance remains unaltered, which is explained by the symmetry of the dataset.

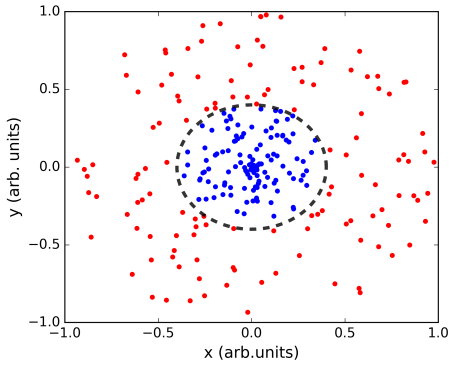

FIGURE 5.5: Circular dataset.

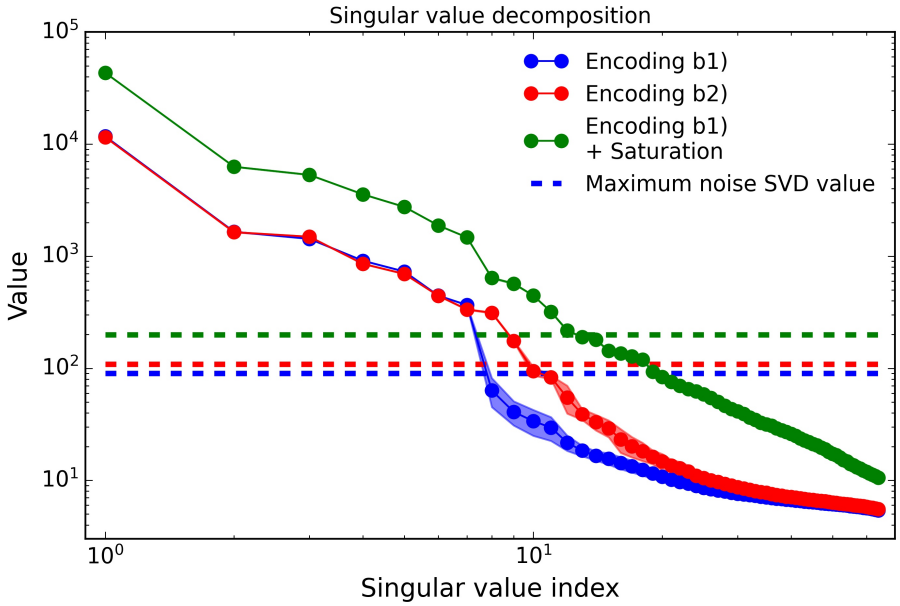

FIGURE 5.6: Singular value decomposition for the different encoding schemes within phase modulation. The dashed lines represent the highest singular value from the noise matrix for each encoding scheme. The solid and dashed lines are colour matched.

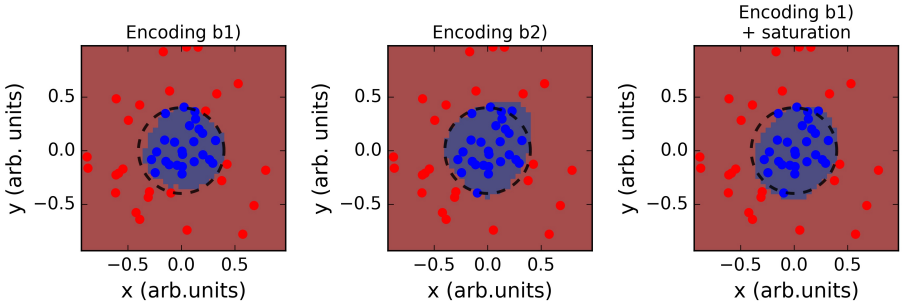

FIGURE 5.7: Classification performance of the machine on the test set overlaid on a rectangular grid of 40x40 points sampled across the respective domain, for the different encoding schemes.

Encoding scheme	**rank(H) (Prediction)**	**rank(H) (Experimental)**	**Number of input features**	**Accuracy (%)**
b1)	7	7	$\{b_1, b_2\}$	98.08
b2)	21	9	$\{b_1, b_1^2, b_2, b_2^2\}$	94.23
b1) + saturation	-	12	$\{b_1, b_2\}$ + saturation	94.23

TABLE 5.5: Experimental results on the classification task for the circular dataset.

To further benchmark the capabilities of our system, we have repeated the above on a more complicated dataset, as shown in figure 5.8. In this case, we can see that the effect of a non-linear encoding has diminished the accuracy (see table 5.6), and the effect of saturation is unnoticeable. Nonetheless, the machine exhibits great performance, offering an accuracy above 90%.

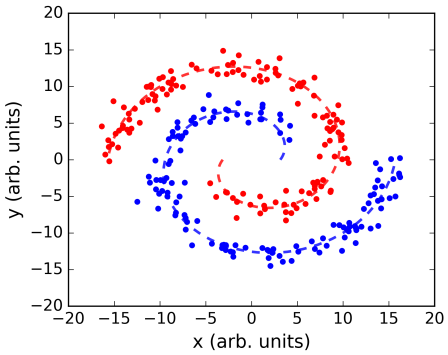

FIGURE 5.8: Spiral dataset.

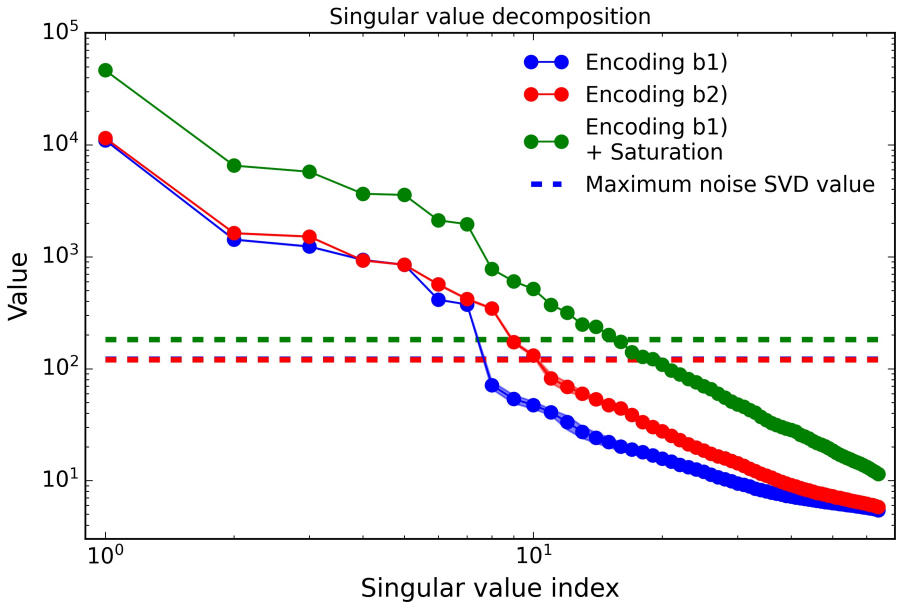

FIGURE 5.9: Singular value decomposition for the different encoding schemes within phase modulation. The dashed lines represent the highest singular value from the noise matrix for each encoding scheme. The solid and dashed lines are colour matched.

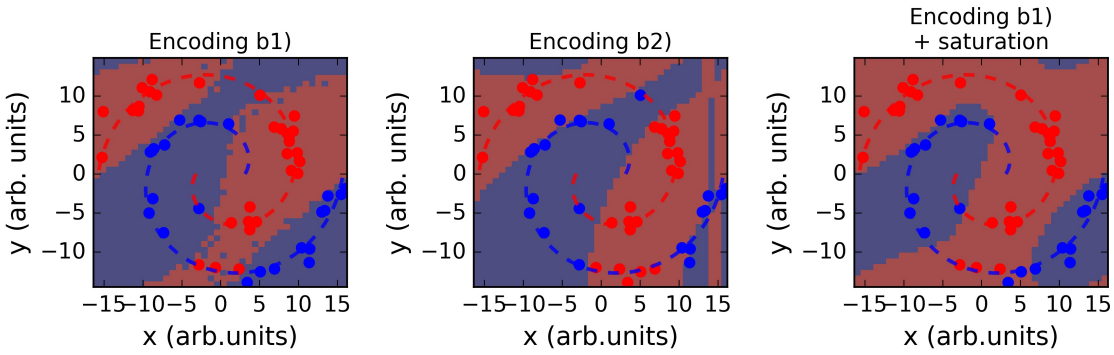

FIGURE 5.10: Classification performance of the machine on the test set overlaid on a rectangular grid of 40x40 points sampled across the respective domain, for the different encoding schemes.

Encoding scheme	**rank(H) (Prediction)**	**rank(H) (Experimental)**	**Number of input features**	**Accuracy (%)**
b1)	7	7	$\{b_1, b_2\}$	92.31
b2)	21	9	$\{b_1, b_1^2, b_2, b_2^2\}$	86.54
b1) + saturation	-	12	$\{b_1, b_2\}$ + saturation	92.31

TABLE 5.6: Experimental results on the classification task for the spiral dataset.

5.2 Amplitude modulation

For the amplitude modulation study, the experimental set-up is in every way similar to the one in figure 5.1, the only difference being the optical encoding and the spatial filtering section where now we only need to filter out the first order of diffraction of the blazed grating.

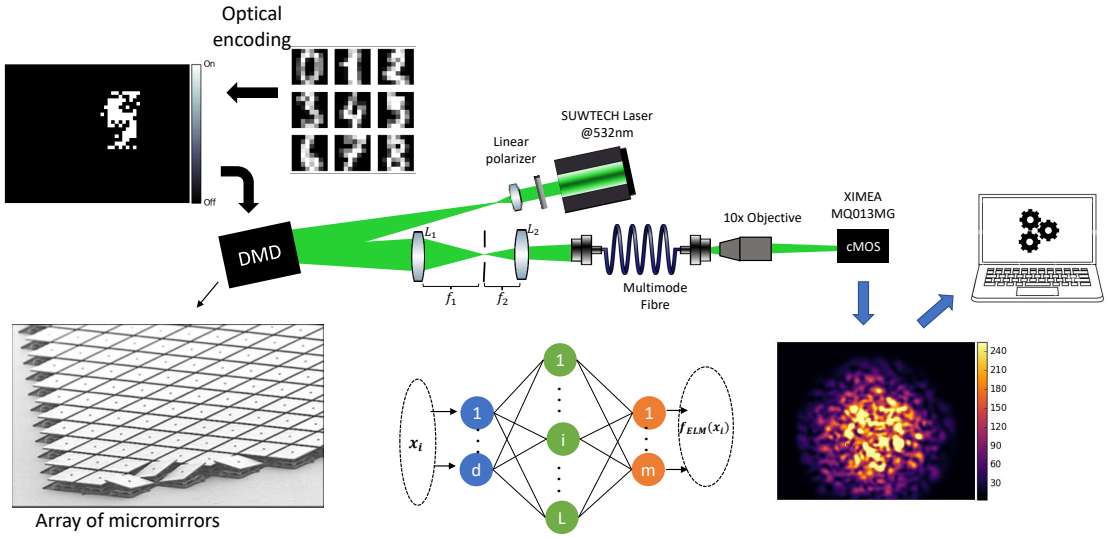

FIGURE 5.11: Illustration of the experimental set-up used for amplitude modulation.

The amplitude encoding schemes follow very closely the ones used for phase modulation as can be seen in figure 5.12, we only have to replace $\phi(x,y) \rightarrow A(x,y)$ and the background plane is now always on the "On" state. The amplitude modulation follows the procedure outlined in chapter 4. In tables 5.8 and 5.7 are outlined the encodings in detail and in table 5.9 are the predicted ranks of the outputs for the different encoding schemes.

a1)	a2)	a3)
$\phi = A_{max}\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_1 = A_{max}\frac{x-x_{min}}{x_{max}-x_{min}}$ $\phi_2 = A_{max} - A_{max}\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_1 = A_{max}\frac{x-x_{min}}{x_{max}-x_{min}}$ $\phi_2 = A_{max}\left(\frac{x-x_{min}}{x_{max}-x_{min}}\right)^{1.2}$ $\phi_3 = A_{max}\left(\frac{x-x_{min}}{x_{max}-x_{min}}\right)^{0.65}$ $\phi_4 = A_{max}\left(\frac{x-x_{min}}{x_{max}-x_{min}}\right)^{3}$

TABLE 5.7: Amplitude modulation encodings for the regression task according to figure 5.12.

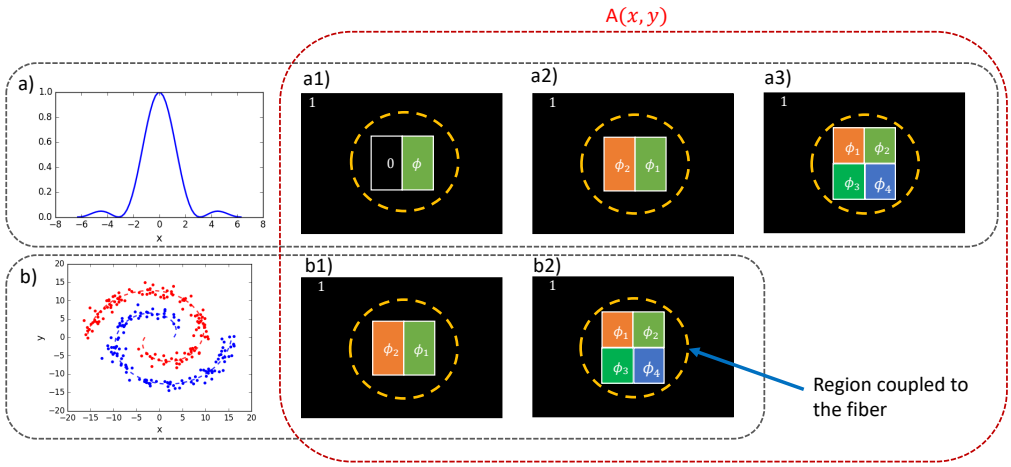

FIGURE 5.12: Illustration of the employed encoding schemes for amplitude modulation.

b1)	b2)	
$\phi_1 = A_{max}\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_1 = A_{max}\frac{x-x_{min}}{x_{max}-x_{min}}$	$\phi_3 = 2A_{max}\left(\frac{x-x_{min}}{x_{max}-x_{min}}\right)^3$
$\phi_2 = A_{max}\frac{y-y_{min}}{y_{max}-y_{min}}$	$\phi_2 = A_{max}\frac{y-y_{min}}{y_{max}-y_{min}}$	$\phi_2 = A_{max}\left(\frac{y-y_{min}}{y_{max}-y_{min}}\right)^3$

TABLE 5.8: Amplitude modulation encodings for the classification task according to figure 5.12.

	$A(x,y)$				
	a1)	a2)	a3)	b1)	b2)
rank(**H**)	3	3	15	6	15

TABLE 5.9: Prediction of the rank of the matrix **H** according to our model for amplitude modulation.

5.2.1 Results and discussion

For this part of the experiment, we've again used a region of 300x300 pixels of the DMD screen, however, it was noted that the amplitude modulation effect on the rank of **H** was far less pronounced regardless of the number of samples, either for regression or classification tasks. Thus, Weyl's inequality was no longer a useful metric since the noise didn't present itself as a perturbation to our system. In order to still be able to validate table 5.9, we measured 100 datasets of 300 samples each for regression and 120 datasets and 300 samples for classification. As we calculate $\langle \mathbf{H}^{exp} \rangle$ with an increasing number M of datasets, one should see a suppression of the noise contribution to the singular

value spectrum of **H**, while the ones from that effectively arise from information encoding should either be stable from the beginning or stabilise after an adequate averaging.

The results are shown in figure 5.13 and 5.14. Looking at panels a2) and b2) of each figure we see that the number of constant singular values matches exactly as those predicted in table 5.9. It is also interesting to note that in panels a1) and a2) the ELM prediction is very close to an inverted parabola, which makes sense given the quadratic polynomial nature of the effective activation function (I^{il}_{out}, in equation 2.20). As for the non-linear encodings (panels c) in each figure), we're only able to see 6 constant values, however, there seems to be some stabilization of the higher index singular values when 100 datasets were used, indicating that there is further information after full noise supression, however, to achieve such effect many more datasets would have to be taken. Furthermore, we see a performance increase with the use of a non-linear encoding, and is accompanied by the rank increase, which is in contrast with the behaviour in phase modulation. Also note that the same stabilization appears in some cases where it wasn't predicted. The reason for such effect is that while the modulation is mainly on the wave amplitude, there may be some unintentional phase modulation, whether by diffraction effects on the DMD or from wavefront aberrations, thus enabling a higher rank of **H** at the output than predicted, albeit being a much weaker effect. In both tasks, there's a clear improvement of the overall performance of the machine when including camera saturation.

Bearing in mind the previous results, we also benchmark the machine on the well-known MNIST dataset [74]. To do so we've used a total dataset of 1790 images of 28x28 pixels with 16 amplitude levels, with a train-test split of 80%-20%. For greater performance we've made use of the non-linear detection on the camera. The results are as shown in figure 5.15, where we were able to achieve an accuracy over 90%.

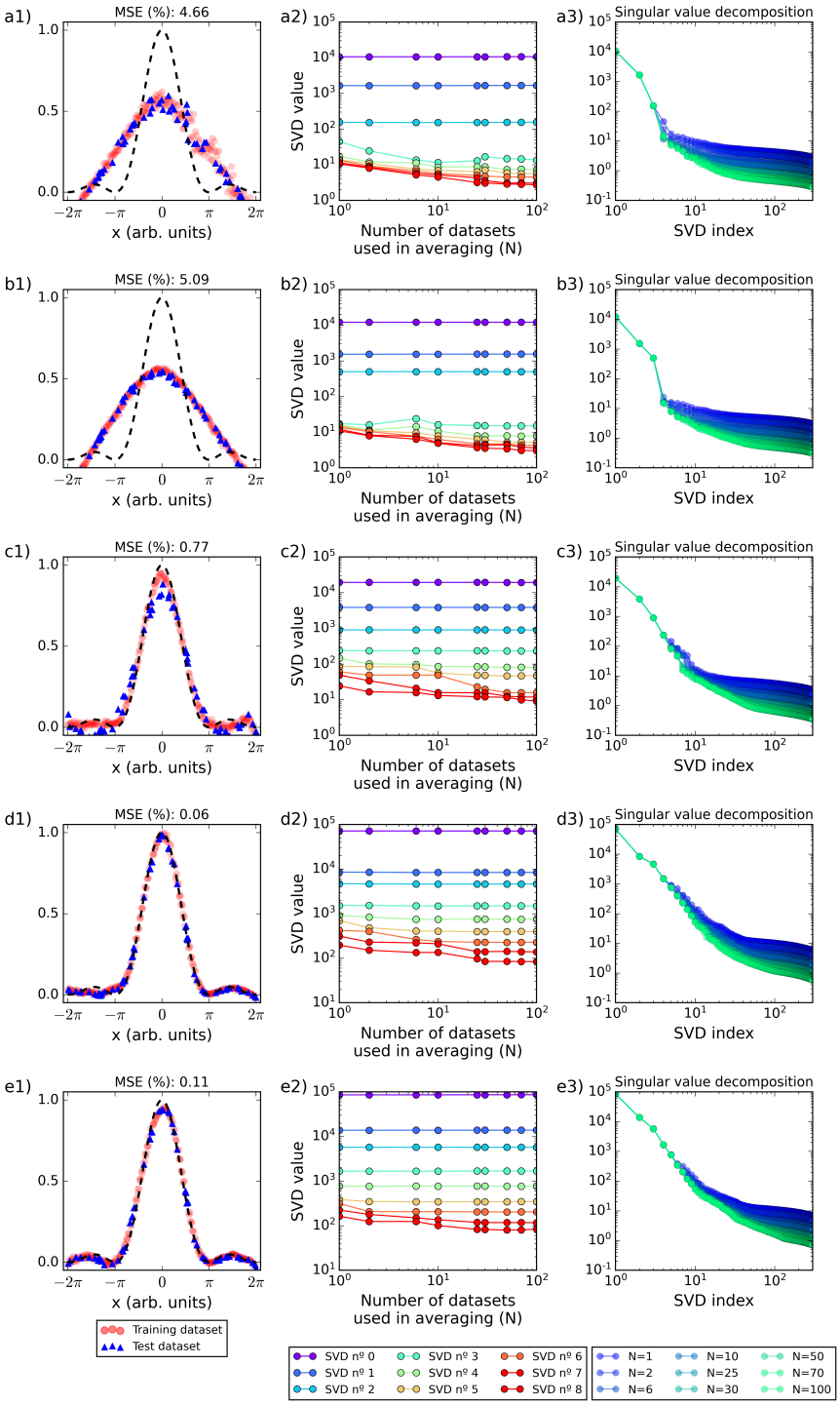

FIGURE 5.13: Experimental results with amplitude modulation on a regression task. Panels a), b), c),d) and e) correspond to the encoding schemes a1), a2), a3) and a1) and a2) with camera saturation, respectively. The first column in each panel demonstrates the performance of the model on the training set (semi-transparent red circles) and test set (blue triangles). In the second column it's a representation of the evolution of the first 9 singular values of $\langle \mathbf{H}^{exp} \rangle$ as a function of M, the number of datasets for averaging. In the third column, it's plotted the singular value spectrum of $\langle \mathbf{H}^{exp} \rangle$ and it's evolution for an increasing M.

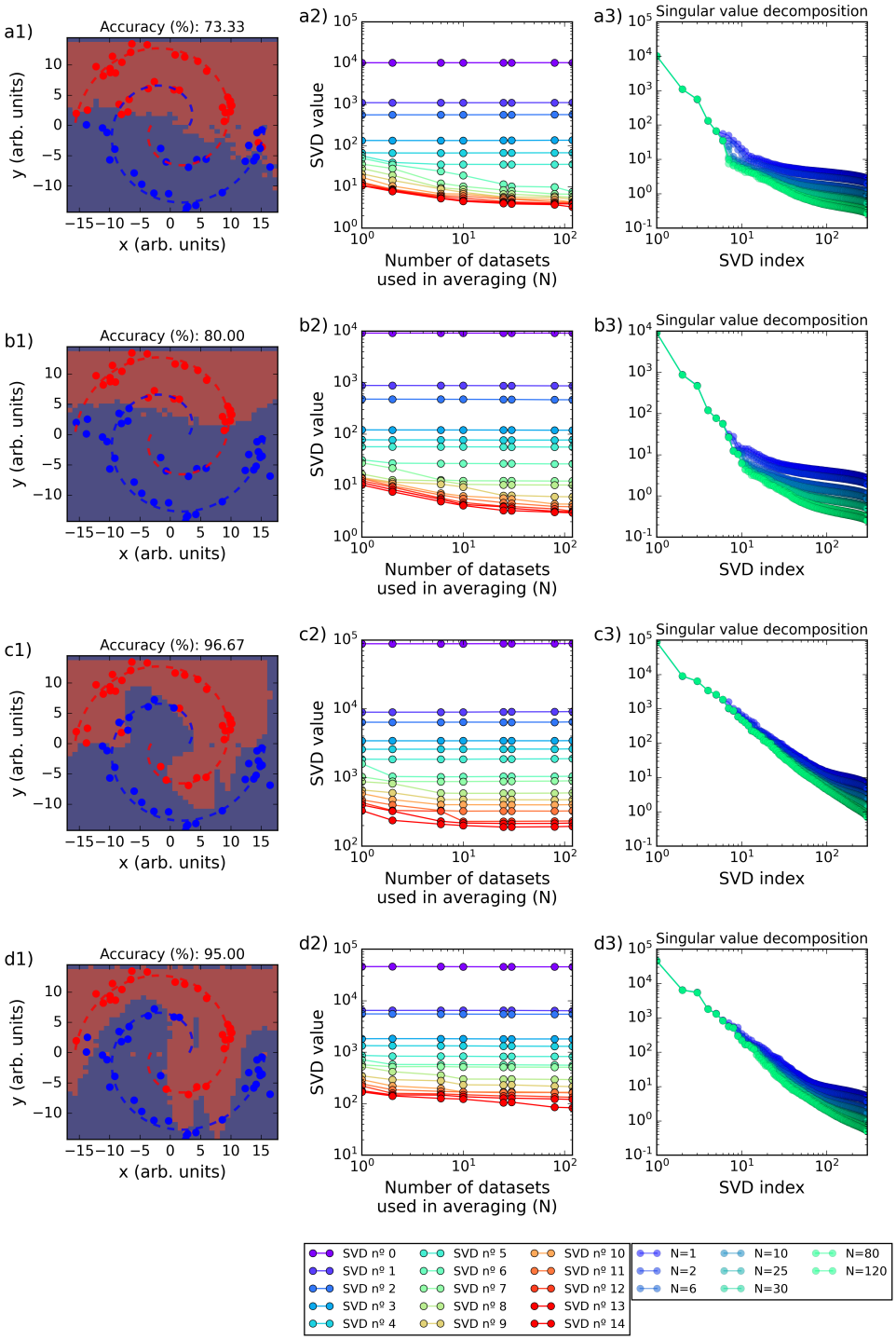

FIGURE 5.14: Experimental results with amplitude modulation on a classification task. Panels a), b), c) and d) correspond to the encoding schemes b1), b2) and b1) and b2) with camera saturation, respectively. The first column in each panel demonstrates the binary classification performance on the test set and on a rectangular grid of 40x40 points sampled across the respective domain. In the second column it's a representation of the evolution of the first 9 singular values of $\langle \mathbf{H}^{exp} \rangle$ as a function of M, the number of datasets for averaging. In the third column, it's plotted the singular value spectrum of $\langle \mathbf{H}^{exp} \rangle$ and it's evolution for an increasing M.

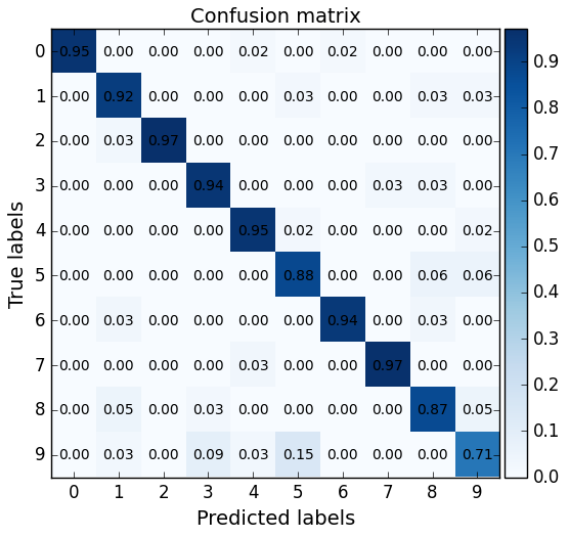

FIGURE 5.15: Confusion matrix with amplitude modulation on the MNIST dataset.

5.3 Final remarks and future work

In this chapter we have been able to experimentally validate our theoretical framework developed in chapter 2, thus proving that our set-up constitutes an optical implementation of an extreme learning machine. We have verified the rank scaling laws, as per equations 2.39 and 2.42, for small numbers of input encoding fields. Furthermore, we've benchmarked our machine on standard machine learning tasks having achieved remarkable performance. With our results we were able to draw conclusions on the learning capabilities of an ELM from a more fundamental point of view, greatly complementing the current literature on extreme learning machines, particularly its optical implementations.

From a technological standpoint, our results, allied with the simplicity of our set-up can be quite stimulating. On accounts of energy costs, our set-up is highly efficient since it does not dissipate significant amounts of energy anywhere besides the transduction process, and since there aren't any optical non-linearities, the system supports wavelength multiplexing, allowing a much higher capacity computation to take place, given that we have the necessary detection system. Note as well that the rank of the outputs in any experiment never got near the full dimensionality of the output space. This indicates that the detection could be greatly simplified, allowing for, not only a much simpler detector, but also faster and cheaper to be employed. Furthermore, all of the remarks thus far indicate that the system is compatible with a chip-scale implementation, which would allow for a more deployable and compact technology to reach the market. As an example, a simple

MMI (multi mode interferometer) could harness the same fundamental principles as our set-up. Finally, it's important to recognise that we have studied the computing capabilities of our machine with an artificial encoding of the information on the wavefront. Indeed, as it stands, the information goes from the electronic domain, to optical and then electrical again. This forces the transduction process to take place twice, and induces a great energy cost. At the same time, numerous technologies nowadays rely on the detection of optical signals for further processing, usually being fed to a machine learning algorithm. Two particularly interesting examples include: cell classification through backscattered light on optical tweezers and LIBS (Laser Induced Breakdown Spectroscopy) spectrum, both of which are active fields of research within our group. With our set-up, we can skip a transduction step and allow for most of the computation to take place within the analog domain, thus saving a significant amount of energy, as well as time due to the decreased latency associated with the optical to electrical conversion.

As we've seen, employing our set-up to an optical signal for analog processing can be quite advantageous. However, at the moment, our machine still needs to convert an optical signal to an electrical signal and perform some calculations in the digital domain. This induces a limiting latency and avoidable energy consumption, whose culprit are the advanced electronics. However, the mathematical operation performed for prediction, $\mathbf{H}_i\beta$, is a simple dot product which can be performed in the analog domain. By bypassing the digital computations, we will be able to employ much cheaper, energy-efficient and faster components that will allow to perform computations at a high speed. This will be the topic of discussion in the next chapter.

Chapter 6

Experimental implementation of an analog ELM

From previous work, we were able to show that it is possible to construct an extreme learning machine within an optical platform by leveraging the complex dynamics of a multimode fibre, however, such a machine still required the use of advanced electronics to perform the final digital computations for either a classification or regression task. Our first approach to achieve a fully analog computation has been inspired in the recent success of diffractive neural networks [40], however this has proven to be ineffective. Nonetheless, the results of this attempt can be seen in Appendix F where we explore wavefront optimisation algorithms, and in Appendix G where we apply such algorithms to train our network. In this chapter, we demonstrate our second approach, where we intend to simplify the computation further and build an analog computer, based on an ELM architecture, whose inner workings rely on a harmonious interplay between optics and electronics.

6.1 Problem statement

As can be seen in chapter 5 we had remarkable performance when testing the machine on a regression task with a non-linear function, namely $f(x) = \left|\frac{sin(x)}{x}\right|^2$. For that case, the problem at hand was the following

$$\min_{\beta \in \mathcal{R}^{L \times m}} ||\mathbf{H}\beta - \mathbf{T}||^2 + \lambda ||\beta||^2 \tag{6.1}$$

where H_i is the intensity pattern recorded on the camera for the sample $\mathbf{x}_i$ with respective target value T_i such that $m = 1$. Notice that having found the best solution for β, the operation that allows computing the desired target function is $y_i = \sum_j H_{ij}\beta_j$ which is a simple weighted sum of the intensity pattern. Analogically, the sum of an intensity pattern can be done through an integral within a photodetector, the only thing left is a clever amplitude modulation, which we have already explored with the DMD in chapter 4, to apply the respective weights β_j to the output channels.

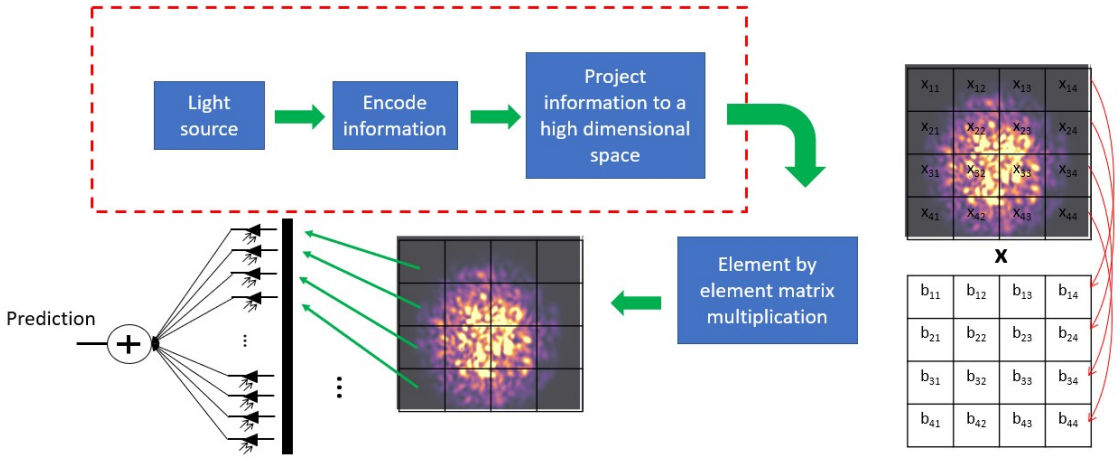

FIGURE 6.1: Illustration of the information flow through our new set-up for a fully analog ELM. The elements within the red dashed box consist of the set-up studied in chapter 5.

6.2 Experimental set-up

The experimental set-up is shown in figure 6.2. We use a 50mW laser with central wavelength at 532nm. The $\lambda/4$ plate along with the linear polarizer are used to control the light intensity throughout the optical set-up. The beam is expanded and collimated through a telescope system, and goes into a 50:50 beamsplitter, where the perpendicular reflection goes to a beam dump (not shown) and the parallel goes into the DMD screen, where it illuminates only half of the screen as shown. Notice that the mirrors of the DMD can be only be in an "on" or "off" state, thus there will be two possible paths, each at 12° with respect to the incident beam. We shall call henceforth the bottom path the "on" path, and the top path the "off" path (green and blue illustrations, respectively). For the incident beam, we will generate a Lee hologram on the corresponding side of the DMD in order to modulate the phase of the input field. Furthermore, we employ a 4f imaging system on

the "on" path which allows for the appropriate spatial filtering needed for phase modulation and allows for a seamless modification of the modulation principle from phase to amplitude of the input light at the multimode fiber since the 3 spots will coincide at the focal spot of the last lens. From an experimental point of view, it's also worth pointing out that some light of the input beam will go into the "off" path due to the pixels in the "off" state, which will contaminate the measurements on the digital camera. In order to prevent this, we place a linear polarizer perpendicular to the one at the input. The output of the MMF is then collimated with a 10x objective. Such output is a speckle pattern, whose optical path is represented in blue in the schematics, and is directed to the unused part of the DMD screen as shown. The output of the DMD is then imaged with a 4f system, allowing for a spatial filtering and beam reduction to take place.

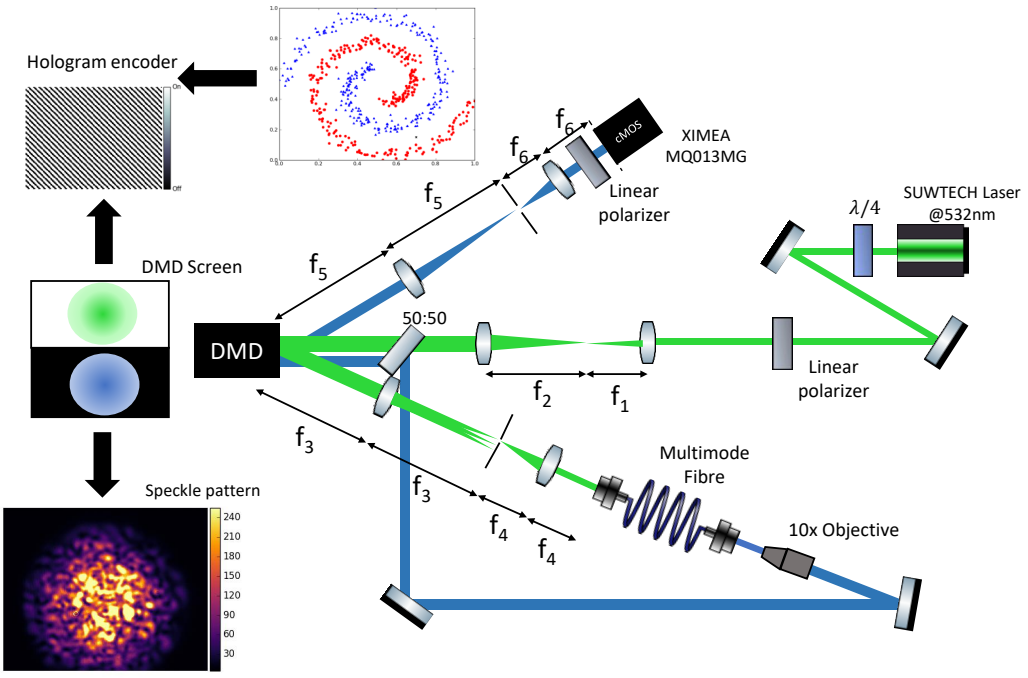

FIGURE 6.2: Experimental set-up for the implementation of analog extreme learning machine.

When performing the experiment, there were two major challenges that we faced, and to better understand them we'll go through each individually.

6.2.1 Output downsampling

The first challenge we need to tackle is to choose the number of output channels we want to consider. In chapter 5, this was done by downscaling an image detected on the full screen of our digital camera. As stated, we wish that on light's second pass through the DMD, an element-by-element matrix multiplication to take place, but first we must find such matrix β. To this end, a one to one correspondence must be established between macropixel region on the DMD and output channel on the camera. While there are different ways one can achieve this, we have opted for an approach that best emulates an array of large photodiodes. The method is illustrated in figure 6.3. Prior to the data acquisition, we run N_c binary amplitude masks on the DMD, one per each output channel. For each frame, we collect a high exposure image on the camera, and then apply a threshold operation on the image. Such threshold is chosen to be above the shot noise level on the camera. By doing so, we end up with N_c digital images that select the region of space corresponding to the output channel. Then, to generate a downsampled image, we collect an image from the DMD and then multiply this image by the respective phase masks and the integral over all the array is the value to consider for the output channel, as illustrated on the bottom part of figure 6.3.

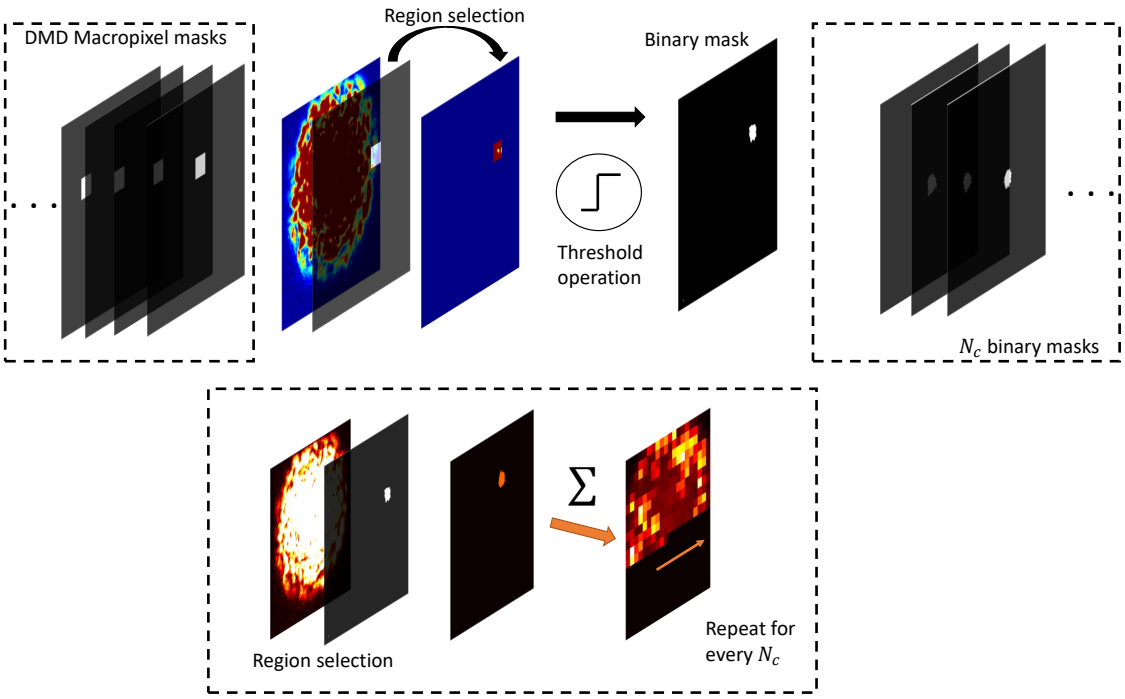

FIGURE 6.3: Illustration of the downsampling method. In the upper part are exemplified the methods for generation of digital binary masks and on the bottom is represented the digital downsampling method.

While this method is, in principle, quite accurate, it has some drawbacks. First of, the threshold function isn't accurate enough to select the correct region, due to poor alignment of the optical elements which results in unwanted diffraction to take place. Secondly, it was observed a correlation between neighbouring pixels, which allied with unwanted diffraction creates a big deviation from the ideal scenario. Thus, a trade-off situation emerges: if we choose many output channels, we get more resolution and may get more performance, but we get greater diffraction, thus reducing the quality of each channel.

6.2.2 Calculation of the weight matrix

As we've said previously, the problem at hand is to find the solution to equation 6.1, but oftentimes that solution has negative weights, which we're not able to implement in the DMD. There are two solutions for this. On the one hand we can define β' such that

$$\beta' = \frac{\beta - \beta_{min}}{\beta_{max} - \beta_{min}} n_{\text{levels}} \tag{6.2}$$

The new targets are

$$\begin{aligned} \mathbf{H}\beta' &= \mathbf{H}\left(\frac{\beta - \beta_{min}}{\beta_{max} - \beta_{min}} n_{\text{levels}}\right) \\ &= \mathbf{H}\beta \frac{n_{\text{levels}}}{\beta_{max} - \beta_{min}} - \mathbf{H}\frac{\beta_{min}}{\beta_{max} - \beta_{min}} n_{\text{levels}} \\ &= \mathbf{T}\frac{n_{\text{levels}}}{\beta_{max} - \beta_{min}} - \mathbf{H}\frac{\beta_{min}}{\beta_{max} - \beta_{min}} n_{\text{levels}} \end{aligned} \tag{6.3}$$

from equation 6.3 we can see that the new predictions are simply a scaled version of the original ones, $\mathbf{T}$, plus a constant bias. Thus, in principle, it would be possible to compensate for the negative weights while applying a purely positive β. However, this attempt has not been successful experimentally. The other solution is to apply constraints to the optimisation algorithms and find the best solution within the positive domain. As for the discrimination, we've employed the following

$$\beta' = \text{int}\left(\frac{\beta - \beta_{min}}{\beta_{max} - \beta_{min}} n_{\text{levels}} + 0.5\right) \tag{6.4}$$

where the $\text{int}(\cdot)$ operation returns the lowest integer value a decimal value may encompass. It's worth noting that we've found this discrimination to be the least limiting step, whereas the constraints on the optimisation were quite significant, often resulting in an order of magnitude of difference in the mean squared error when benchmarking the machine.

6.3 Results

Having considered the previous remarks, we have chosen the number of output channels as 400 (20 in each direction) and 9 modulation levels. To test for generalisation capability against noise, we've evaluated the solutions not only on train and test datasets, but on four extra datasets. The mean squared error as a function of the regularisation parameter, α, as can be seen in figure 6.4, where it's highlighted the selected α value for the model with the vertical dashed black line, and its respective performance is depicted in figure 6.6.

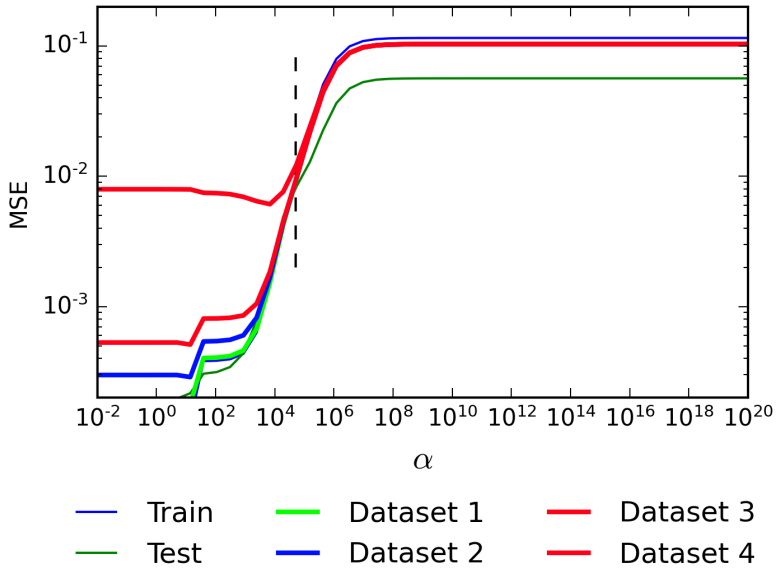

FIGURE 6.4: Mean squared error as a function of the regularisation parameter α for ridge regression, for several datasets. The dashed black line represents the value chosen for the model.

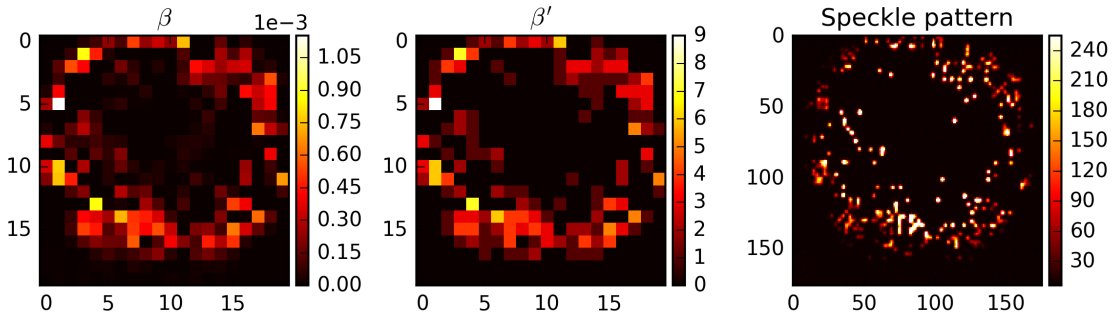

FIGURE 6.5: Calculated matrix β, as well as β' as per equation 6.4 and resulting speckle pattern with β' matrix applied on the DMD, following the amplitude modulation discussed in chapter 4.

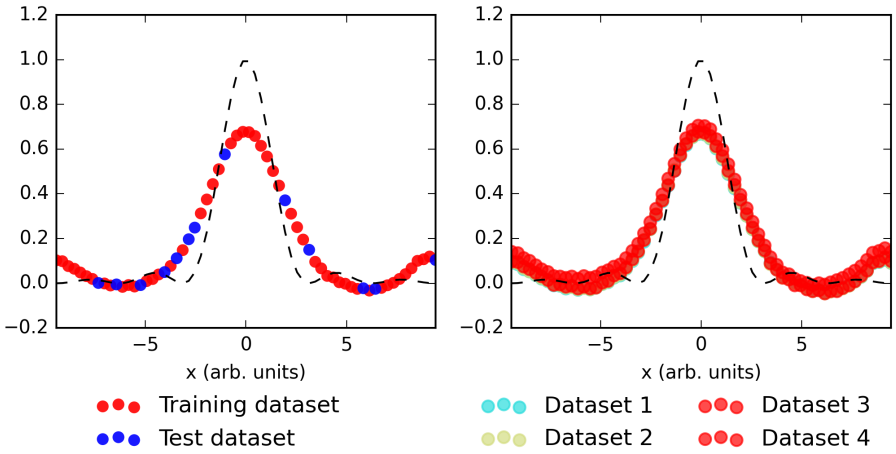

FIGURE 6.6: Regression performance on all datasets.

Finally, we test the machine in the analog domain, and the results can be seen in figure 6.7. As can be seen from the figure, it is clear that we were able to replicate the matrix multiplication operation in the analog domain, as the curves share a highly similar profile. The discrete version of β resulted in a small vertical shift, but the curve shape was kept. Do note that in this plot, we applied the compensation described in equation 6.3 so as to get both curves (β and β') on the same scale.

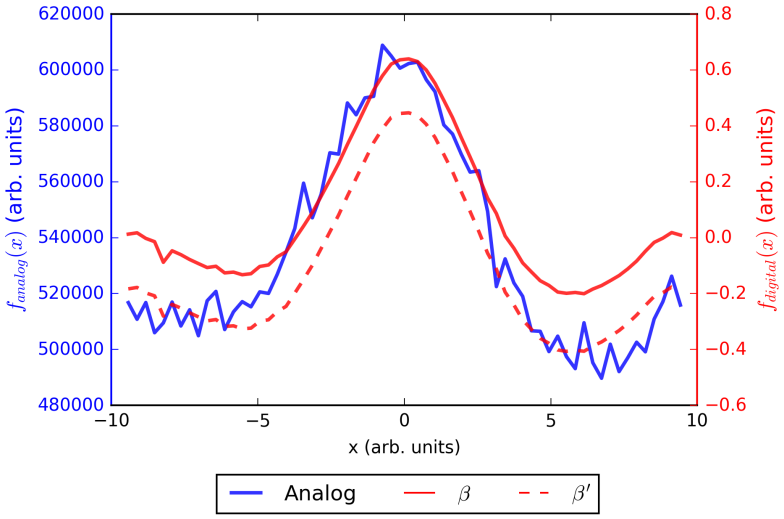

FIGURE 6.7: Analog performance of the machine. In the red lines we have the predictions resultant from digital calculation, whereas the blue curve stems from a simple sum across the camera's pixels.

6.4 Final remarks and future work

In this chapter we have sought to further extend our optical implementation of an extreme learning machine into the analog domain by implementing a matrix multiplication operation with physical devices. Our implementation allows the theoretical framework developed in chapter 2 to retain validity, thus proving that this is in fact a fully analog computing machine. Nonetheless, we have seen that we were not able to achieve the same levels of performance regarding the regression task as we did in chapter 5. This can be attributed to the limitation to only positive weights in β, and not the implementation itself. In fact, on that regard, we have been remarkably successful as the analog curve profile follows closely the best achievable curve with digital calculation, given the current constraints. For this reason we propose an experiment for a future implementation in figure 6.8 that should be able to surpass these limitations. The operating principle is in every way similar to the previous work, but with a key difference present in DMD2. Our major limitation thus far has remained at a fundamental level, by restricting $\beta_i \geq 0$. To solve this, we simply note that for any $\beta_i \in \mathbb{R}$ it is always possible to write $\beta = \beta^+ - \beta^-$, such that $\beta_i^+ \geq 0$ and $\beta_i^- \geq 0$. Allied with our previous results, we can now implement the two matrix multiplications (β^+ and β^-) independently on DMD2 and the resulting sums can be subtracted by simple electronics to recover $X_i\beta$.

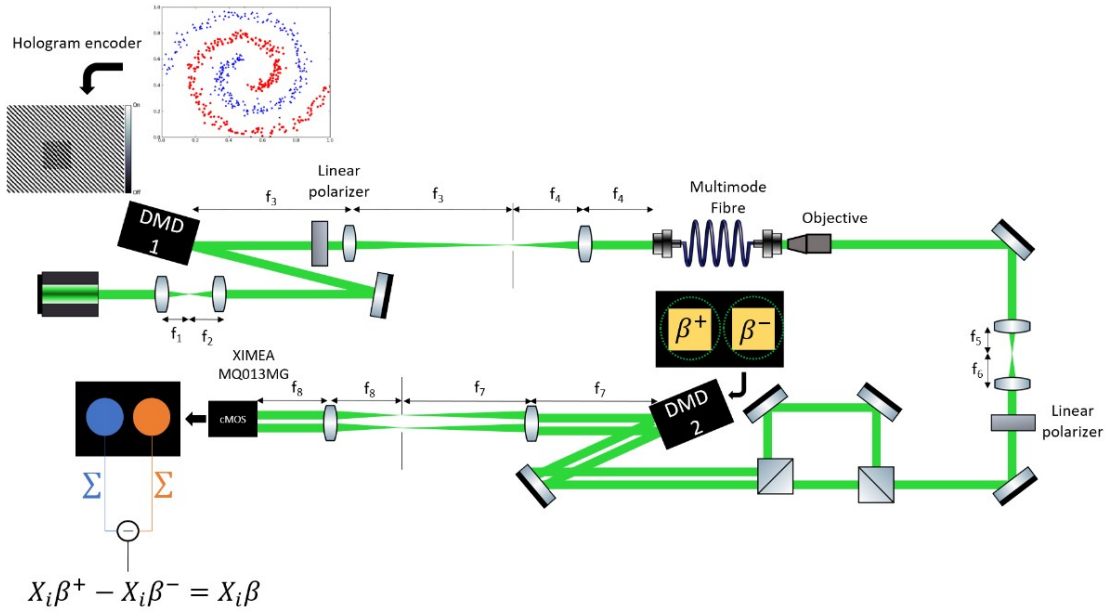

FIGURE 6.8: Proposed future experiment.

A question remains: have we built an all optical analog computer? In both of our implementations we have relied on two non linear physical operations to guarantee the non linear activation functions on the output channels required by the ELM framework. Such operations have been the non linear response curve of photodetectors, and the intensity measurement in itself. Relying on them have allowed us to envision and build an analog computer. Generally, an optical computer is characterised by light controlling light for the various tasks underlying a computer: data processing, routing, transmission and storage. Despite this, at the end of any computation there is an output, which will be turned into a signal that must be interpreted. When using optics, such output generally takes the form of an intensity measurement, as the electric field is a far too expensive measurement to make. Thus, even the most advanced optical computer will need a detection stage in order to interpret the outcome. Nonetheless, such detection takes no active part in the computation, thus, in that perspective, we have not built an all optical computer. Instead, our machine relies on an harmonious interplay between optics and electronics for the computation. Consequently, from an etymological point of view, we can affirm that we have built a fully analog optoelectronic extreme learning machine.

Finally, it is interesting to note that our system can be transformed into a fully optical computer by the insertion of an optical non-linearity, such as a $\chi^{(3)}$ non linearity. Admittedly, our theoretical framework in chapter 2 would no longer be valid. Instead, the non-linear transformations would be guaranteed by the evolution of the speckle pattern through a non linear shrodinger equation, as per equation 2.17. Plus, due to the fast response of Kerr mediums (typically on the pico- or sub-pico-second response times [75]) our machine could still be operated within the ELM regime. Another plausible approach would be to add a saturable absorber such as grafene. The output weights could still be implemented in DMD2, as per figure 6.8, but now the photodetectors would act merely as intensity detectors. They would inadvertently contribute a further source of non-linearities, but the crucial computation would be largely taken cared of by the non-linear medium. An all-optical ELM also constitutes a goal within our research and work is being done in such directions. We have already developed some necessary tools for such set-ups (see Appendix B and C), and we aim to develop further towards that objective.

Chapter 7

Conclusions

Deep neural networks have become a ubiquitous tool across many scientific domains and areas of knowledge. Yet, the performance of neural networks is intimately tied to its scalability, and with the impending plateau of Moore's law, emerges a need to continue performance scaling in the absence of electronics miniaturization. A promising route lies within hardware specialization, especially for machine learning applications. Leveraging optics for such implementations can bring significant performance gains due to the high energy efficiency, intrinsic parallelism, high bandwidth and speed of operation. Among the various neural network architectures, ELMs are particularly attractive for hardware implementations due to its simplicity. Thus, the first part of this thesis was dedicated to understanding the mathematical foundations of these networks, with emphasis on a particular optical ELM set-up. The second part consists of a set of numerical simulations that allowed us to confirm our previously studied theoretical model. In the third part, we present the experimental results for our physical implementation of an optical ELM. Finally, in the last part, we extend our machine further into the analog domain, and show our experimental results that demonstrate the realization of a primitive optoelectronic computer based on a non-Von Neumann architecture. In detail, the conclusions of each chapter are outlined as follows.

In chapter 2 we review the mathematical foundations of ELMs and how it fits within the ML landscape. We also review the state of the art on accounts of optical ELMs and conclude that there is a lack of a more fundamental understanding of the inner workings of such machines. To that end, we develop a theoretical framework for an optical ELM based on optical complex media, which allows us to not only identify the effective activation functions of the output channels, but also infer on the dimensionality of the output

space, a key parameter for the learning capabilities of an ELM. Alas, we come to the conclusion that the number of dimensions of the output space (rank($\mathbf{H}$)) scales quadratically with respect to the number of input encoding features on the wavefront, be it in amplitude, phase or their combined encoding mechanisms.

In chapter 3 we present a set of numerical simulations based on speckle generation, which allowed us to corroborate the theoretical framework by demonstrating the rank scaling laws for the various encoding mechanisms. At the same time, we have also benchmarked the system in standard machine learning tasks, and were able to identify the rank($\mathbf{H}$) as an important performance metric. Finally, we also evaluated the effect of a strong physical non-linearity and concluded that it leads to a significant increase in overall performance of the ELM.

In chapter 4 we simply introduce the experimental equipment transversal to every experiment and outline the amplitude and phase modulation techniques employed.

In chapter 5 we have given experimental evidence of the rank scaling laws in phase and amplitude modulation, in the low dimensionality cases. The observed discrepancies can be attributed to noise in the system. Through our benchmarks, we have concluded that the rank($\mathbf{H}$) is an important performance metric, but the nature of the non-linear projection of the inputs cannot be overlooked. Thus, rank($\mathbf{H}$) is not ideal in predicting the performance of an ELM, but can be helpful.

In chapter 6 we demonstrate a fully analog optoelectronic ELM. We tested our set-up in a regression task, and we were able to faithfully reproduce a curve that closely resembled that of the best achievable curve in the digital domain, thus proving the computing capabilities of our machine. We outline the main challenges faced, and we propose a future experiment that should be able to surpass them.

To put it simply, this thesis explores, from an experimental and theoretical point of view, the physical implementation of an emerging neural network architecture, ELM, in an optical platform. Nature has inspired us in many technologies: from velcro to aircrafts, modern wind turbines, and even paint. It was only a matter of time until we started to do the same for computing. It is within this perspective that physical implementations of ELMs have become particularly relevant. They are able to harness the complex dynamics, energy efficiency and speed of operation that only nature provides, and then combines these feats with the digital domain in a harmonious way to provide a computing platform. Doing so with optics brings natural advantages, and the widespread use of it as

information carrier nowadays, makes optical ELMs particularly relevant for end-of-line applications. Despite this, the field of optical ELMs is still its infancy, and we hope that our work has contributed to a deeper understanding of these kinds of machines, allowing for a better design and task selection, paving the way to the future of computing.

Appendix A

Mathematical derivations

Linear regression solutions

In this section we wish to derive the analytical expression for the solution of the linear system of equations given by

$$\mathbf{H}\beta = y \tag{A.1}$$

where $\mathbf{H} \in \mathbb{R}^{N\times L}$, $\beta \in \mathbb{R}^{L\times 1}$ and $y \in \mathbb{R}^{N\times 1}$. There are two scenarios here: i) $N \geq L$, where the system os *overdetermined*, and ii) $N < L$, where the system is *underdetermined*. In the first case, the solution to equation A.1 can be found via minimising $||y - \mathbf{H}\beta||$, but in the latter, doing so does not provide a unique solution [76]. We will examine both cases carefully, and arrive at closed form solutions to both.

Overdertermined system: $N \geq L$

The goal is to solve

$$\min_{\beta\in\mathbb{R}^{L\times 1}} ||y - \mathbf{H}\beta||^2 \tag{A.2}$$

We'll define $J(\beta) = ||y - \mathbf{H}\beta||^2$, and let β^* be the solution to A.2. In that case, we know that $\frac{\partial J}{\partial \beta}|_{\beta=\beta^*} = 0$:

$$\begin{aligned}
\frac{\partial J}{\partial \beta} &= (\mathbf{H}\beta - y)^T(\mathbf{H}\beta - y) \\
&= \frac{\partial}{\partial \beta}\left[(\mathbf{H}\beta)^T\mathbf{H}\beta - (\mathbf{H}\beta)y - y^T\mathbf{H}\beta + y^Ty\right] \\
&= \frac{\partial}{\partial \beta}\left[(\mathbf{H}\beta)^T\mathbf{H}\beta - 2(\mathbf{H}\beta)y + y^Ty\right] \\
&= 2\mathbf{H}^T\mathbf{H}\beta - 2\mathbf{H}^Ty
\end{aligned}$$

Setting this derivative to 0 we have

$$\frac{\partial J}{\partial \beta} = 0 \tag{A.3}$$

$$\mathbf{H}^T\mathbf{H}\beta^* = \mathbf{H}^T y \tag{A.4}$$

Note that equation A.4 only has a unique solution for β if $\mathbf{H}^T\mathbf{H}$ is invertible. As it happens, $\mathbf{H}^T\mathbf{H}$ is a Gram matrix [77] and it is invertible. The solution then follows

$$\beta^* = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T y \tag{A.5}$$

Underdetermined system: $N < L$

For the case when $N < L$ the system of equations A.1 is underdetermined, and there are infinite solution β^* for which $J(\beta)$ is minimized. Nonetheless, it would still be useful to find a closed form solution. Do note that equation A.5 is no longer valid here since the matrix $\mathbf{H}^T\mathbf{H}$ is no longer invertible as it is not full rank, having a non-trivial null space. While there may be plenty of formal ways to tackle this problem, we will do it in a rather ingenious way. We start by looking into the null space of $\mathbf{H}$, $\mathcal{N}(\mathbf{H})$, defined as

$$\mathcal{N}(\mathbf{H}) = \left\{v \in \mathbb{R}^{L\times 1} : \mathbf{H}v = 0\right\} \tag{A.6}$$

Let us also define the column space of $\mathbf{H}$, $\mathcal{C}(\mathbf{H})$ as

$$\mathcal{C}(\mathbf{H}) = \left\{v \in \mathbb{R}^{N\times 1} : v = \mathbf{H}w, \quad w \in \mathbb{R}^{L\times 1}\right\} \tag{A.7}$$

By the fundamental theorem of linear algebra [78], it can be shown that:

$$\mathbb{R}^{L\times 1} = \mathcal{N}(\mathbf{H}) \bigoplus \mathcal{C}(\mathbf{H}^T) \tag{A.8}$$

$$\mathbb{R}^{N\times 1} = \mathcal{N}(\mathbf{H}^T) \bigoplus \mathcal{C}(\mathbf{H}) \tag{A.9}$$

By definition, we have $y \neq 0$, thus β cannot belong to $\mathcal{N}(\mathbf{H})$. Since $\beta \in \mathbb{R}^{L\times 1}$, this implies that $\beta \in \mathcal{C}(\mathbf{H}^T)$, which allows us to write $\beta = \mathbf{H}^T w$, where $w \in \mathbb{R}^{N\times 1}$. In that case, we can write

$$\mathbf{H}\beta = y$$

$$\mathbf{H}\mathbf{H}^T w = y \tag{A.10}$$

Note that $\mathbf{H}\mathbf{H}^T$ is full-rank, and is invertible, in which case

$$w = \left(\mathbf{H}\mathbf{H}^T\right)^{-1} y \tag{A.11}$$

from which we can write β as

$$\beta = \mathbf{H}^T \left(\mathbf{H}\mathbf{H}^T\right)^{-1} y \tag{A.12}$$

We have started by saying that for $N < L$ there were infinite solution to the linear system A.1. Then, how come we have arrived at a single solution as in A.12? It can be shown [76] that what we have done was to find a particular solution of A.2 which simultaneously minimises $||\beta||$. Despite this nice formulae, oftentimes we recur to the regularized version of A.2. Let us see how to arrive at the solutions.

Regularized least squares

Again, we have the case for $N \geq L$ and $N > L$.

Overdetermined systems, $N \geq L$

We redefine $J(\beta) = ||\mathbf{H}\beta - y||^2 + \lambda||\beta||^2$, for $\lambda \in \mathbb{R}$. In matrix form, it reads

$$J(\beta) = (\mathbf{H}\beta)^T\mathbf{H}\beta - 2(\mathbf{H}\beta)y + y^T y + \lambda\beta^T\beta \tag{A.13}$$

Taking the derivative and setting it to 0 we get

$$\begin{aligned} 2\mathbf{H}^T\mathbf{H}\beta - 2\mathbf{H}^T y + 2\lambda\beta &= 0 \\ \beta\left(\mathbf{H}^T\mathbf{H} + \mathbb{1}\lambda\right) &= \mathbf{H}^T y \\ \beta &= \left(\mathbf{H}^T\mathbf{H} + \mathbb{1}\lambda\right)^{-1}\mathbf{H}^T y \end{aligned} \tag{A.14}$$

Underdetermined systems, $N < L$

For the case when $N < L$, the only restriction we have placed was that β remained in the column space of $\mathbf{H}^T$, $\beta = \mathbf{H}^T w$. We can then rewrite $J(\beta) = J(\mathbf{H}^T w) = J(w)$

$$J(w) = (\mathbf{H}\mathbf{H}^T w)^T\mathbf{H}\mathbf{H}^T w - 2(\mathbf{H}\mathbf{H}^T w)y + y^T y + \lambda\left(\mathbf{H}^T w\right)^T\mathbf{H}^T w \tag{A.15}$$

Again taking the derivative with respect to w and setting it to 0, we have

$$\mathbf{H}\mathbf{H}^T\mathbf{H}\mathbf{H}^T w - \mathbf{H}\mathbf{H}^T y + \lambda\mathbf{H}\mathbf{H}^T w = 0 \tag{A.16}$$

since $\mathbf{H}\mathbf{H}^T$ is invertible, we can multiply both sides by $\left(\mathbf{H}\mathbf{H}^T\right)^{-1}$. After some algebra we get

$$w = \left(\mathbf{H}\mathbf{H}^T + \mathbb{1}\lambda\right)^{-1} y \tag{A.17}$$

from which we can now find β as

$$\beta = \mathbf{H}^T \left(\mathbf{H}\mathbf{H}^T + \mathbb{1}\lambda\right)^{-1} y \tag{A.18}$$

Mathematical model of an optical extreme learning machine

In this section we present the mathematical derivation of equation 2.20 in greater detail. Our starting point will be 2.19, which we replicate here:

$$E^l_{out}(\mathbf{x}_i) = \sum_{j=1}^{K}\sum_{k=1}^{K} f_j(\mathbf{x}_i) M_{lk} (\mathbf{e}^{in}_j)_k \tag{A.19}$$

In the following, we will omit the functions arguments $(\mathbf{x}_i)$, and replace it by a superscript i. The intensity can now be written as:

$$\begin{aligned} I^{il}_{out} &= \sum_{j=1}^{K}\sum_{k=1}^{K}\sum_{p=1}^{K}\sum_{q=1}^{K} f^i_j (f^i_p) * M_{lk} M_{lq} \underbrace{\left(e^{in}_j\right)_k}_{\delta_{jk}} \underbrace{\left(e^{in}_p\right)_q}_{\delta_{pq}} \\ &= \sum_{j=1}^{K}\sum_{p=1}^{K} f^i_j (f^i_p)^* M_{lj} M^*_{lp} \end{aligned}$$

If we now let $M_{lk} = |M_{lk}| e^{i\phi_{lk}}$ we can write:

$$\begin{aligned} I^{il}_{out} &= \sum_{j=p=1}^{K} |f^i_j|^2 |M_{lj}|^2 + \sum_{j=1}^{K}\sum_{p\neq j}^{K} f^i_j (f^i_p)^* M_{lj} M_{lp} \\ &= \sum_{j=p=1}^{K} |a^i_j|^2 |M_{lj}|^2 + \sum_{j=1}^{K}\sum_{p\neq j}^{K} a^i_j a^i_p e^{i(b^i_j - b^i_p)} |M_{lj}||M_{lp}| e^{i(\phi_{lj}-\phi_{lp})} \\ &= \sum_{j=p=1}^{K} |a^i_j|^2 |M_{lj}|^2 + \sum_{j=1}^{K}\sum_{p>j}^{K} a^i_j a^i_p \underbrace{2|M_{lj}||M_{lp}|}_{C_{ljp}} \cos\left(b^i_j - b^i_p + \phi_{lj} - \phi_{lp}\right) \\ &= \sum_{j=p=1}^{K} |a^i_j|^2 |M_{lj}|^2 + \sum_{j=1}^{K}\sum_{p>j}^{K} C_{ljp} a^i_j a^i_p \begin{bmatrix} \cos(b^i_j - b^i_p) \underbrace{\cos(\phi_{lj} - \phi_{lp})}_{\zeta^e_{ljp}} \\ -\sin(b^i_j - b^i_p) \underbrace{\sin(\phi_{lj} - \phi_{lp})}_{\zeta^o_{ljp}} \end{bmatrix} \end{aligned}$$

Finally, we can write:

$$I_{out}^{il} = \sum_{j=p=1}^{K} |a_j^i|^2 |M_{lj}|^2 + \sum_{j=1}^{K} \sum_{p>j}^{K} C_{ljp} a_j^i a_p^i \left\{ \begin{array}{c} \xi_{ljp}^e \left[\cos(b_j^i)\cos(b_p^i) + \sin(b_j^i)\sin(b_p^i) \right] \\ -\xi_{ljp}^o \left[\sin(b_j^i)\cos(b_p^i) - \cos(b_j^i)\cos(b_p^i) \right] \end{array} \right\} \tag{A.20}$$

On the equality of the rank-sum inequality

The rank-sum inequality states that

$$\text{rank}(A+B) \leq \text{rank}(A) + \text{rank}(B) \tag{A.21}$$

We wish to study the case for equality. The proof for what follows is most likely contained in an advanced linear algebra book, however, we have been unable to find a source. Instead, we have found a proof on a public website [79] which we have adapted for our purposes and replicate below.

Let us assume that equality holds. Let $m, n \in \mathbb{N}$, and define $\mathbf{M}_{mn}(\mathbb{R})$ as a set of all $m \times n$ matrices. Let $A, B \in \mathbf{M}_{mn}$ such that $\text{rank}(A) = k$ and $\text{rank}(B) = l$. Let $\{Av_1, \ldots, Av_k\}$ be a basis of $\mathcal{C}(A)$, and $\{Bw_1, \ldots, Bw_l\}$ be a basis of $\mathcal{C}(B)$. By definition, the set $\{Av_1, \ldots, Av_k, Bw_1, \ldots, Bw_l\}$ spans $\mathcal{C}(A+B)$. Since we have assumed equality on the rank-sum inequality, then $\text{rank}(A+B) = k+l$, then the set $(Av_1, \ldots, Av_k, Bw_1, \ldots, Bw_l)$ is a basis of $\mathcal{C}(A+B)$. Consequently we can state that $\mathcal{C}(A) \cap \mathcal{C}(B) = \{\mathbf{0}\}$. Since $\text{rank}(A) = \text{rank}(A^T)$, then $\mathcal{R}(A) \cap \mathcal{R}(B) = \{\mathbf{0}\}$. For this reason we can conclude:

> If $\mathcal{C}(A) \cap \mathcal{C}(B) = \{\mathbf{0}\}$ and $\mathcal{R}(A) \cap \mathcal{R}(B) = \{\mathbf{0}\}$, then
> $\text{rank}(A+B) = \text{rank}(A) + \text{rank}(B)$.

Appendix B

Phase-only SLM LCoS calibration

Introduction

A spatial light modulator is an electrically programmable device that modulates an optical wavefront according to a fixed and discrete spatial pattern. This modulation can be either in amplitude, phase or a combination of both through a careful combination of optical elements. There are essentially two types of SLM's that differ in the way they are addressed: Optically Addressed Spatial Light Modulator (OASLM) and Electrically Addressed Spatial Light Modulator (EASLM). In an EASLM, as the name suggests, the spatial modulation is assured by an electronic signal, as in most current electronic displays. These devices usually receive an input via VGA or DVI ports, that drive the optical cells so as to control either their absorption or phase shift. While there may be many different technology platforms to develop such devices, we are interested in liquid crystal on silicon displays.

LCoS technology

Liquid crystal over silicon (LCoS) technology combines the unique light-modulating properties of liquid crystal (LC) materials and the advantages of high performance and large scale capabilities of silicon complementary metal oxide semiconductor (CMOS) technology through a dedicated LCOS assembly processes. The typical architecture of an LCOS device is shown in figure B.1. The silicon CMOS back plane acts as one of the substrates and consists of the electronic circuitry that is buried underneath pixel arrays to provide a high "fill factor". The pixels themselves are aluminium mirrors deposited on the surface of the silicon back plane [80]. The incident light travels through a glass substrate

that is protecting the overall optical cell, then through an indium tin oxide layer which is a transparent electrode, followed by the LC layer and it's finally reflected on the aluminium layer.

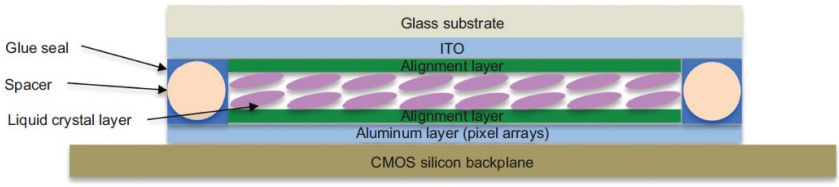

FIGURE B.1: Structure of phase-only LCOS devices, consisting of transparent top substrate with transparent ITO electrodes, alignment layers, LC material, glue seal, spacers (a gap supported by a single layer of spacers to control the thickness of the LC layer), aluminium reflective electrodes (pixel arrays) and a functional CMOS silicon back plane. CMOS, complementary metal oxide semiconductor; ITO, indium tin oxide; LC, liquid crystal; LCOS, liquid crystal on silicon. Diagram adapted from Ref.[80].

Liquid crystals (LC's) are phases of matter with unique properties that can be characterised as an intermediate between a liquid and a solid, that is, they possess some degree of spatial symmetry, typical of a crystal, but also allow themselves to realign upon external stimuli. The most important property of LC's for phase manipulation is the birrefringence, defined as $\Delta n = n_e - n_o$, with n_e as the extraordinary refractive index, which is parallel to the director of the LC molecules and n_o as the ordinary refractive index. Most LC's have a positive birrefringence ($\Delta n > 0$) ranging from 0.05 to 0.45. Among the various phases of LC's the most widely used in many devices is the Nematic because the effective birrefringence of the LC materials can be manipulated easily and continuously with an electric field, which makes it a good candidate for phase-only modulation.[80]

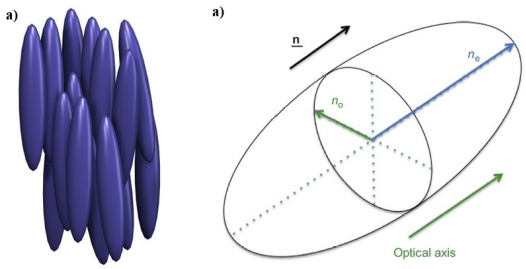

FIGURE B.2: a) Alignment in a nematic phase. Adapted from [81]; b) A schematic of a uniaxial optical indicatrix of refractive index. Adapted from [80].

SLM working principle

We now know that the most widely used phases of LC's are nematic and that their main characteristic that allows for a good wavefront shaping is their birrefringence. However, there are several structures that can be employed to allow light modulation, namely: Twisted nematic configuration, hybrid field effect in nematic LC, electrically controlled birrefringence, surface stabilised ferroelectric LC (SSFLC) and vertically aligned nematic (VAN), optical compensated birrefringence (OCB). These devices have mostly been designed for light modulators by rotating the linear polarisation of light passing through polarisers. However, in this work we are interested in the ECB mode.

Electrically controlled birrefringence

As can be seen in figure B.3, when the cell has an applied voltage the director vector changes from almost planar to vertical. As this change is continuous, we can access almost the complete range of refractive indices between n_e and n_o as the analog voltage is increased. The drawback of this mode for phase-only LCOS devices is the unwanted backflow of LC molecules during switching, which can slow down the electro-optic switching.

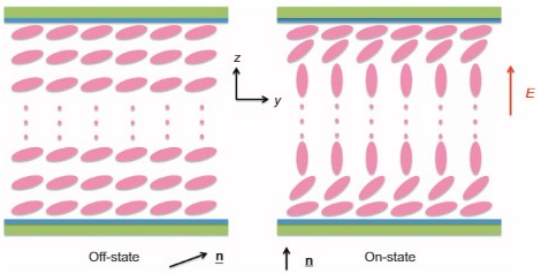

FIGURE B.3: A schematic of the initial Von and Voff states of the ECB electro-optic effect with small tilt angle. This representation is of the zero-twisted configuration in ECB. Adapted from [80].

Holoeye Pluto 2 VIS-016 SLM

The Pluto family consists of a number of phase-only spatial light modulator with a driver unit which has a standard digital video interface (DVI or HDMI). It features a phase-only LCOS micro-display with full HD resolution (1920x1080 pixel), and $8\mu m$ pixel pitch. The device can be addressed with 2D phase profiles via standard graphics cards as an extended monitor device. The green colour channel of the video signal is used for addressing 8 bit gray level patterns (the SLM's native resolution need to be addressed). Moreover,

the display operates in the ECB mode, with a digital driving scheme (pulse code modulation), and is limited to an input frame rate of 60Hz. Do note that the input frame rate relates merely to the addressing speed of the input signal, typically via DVI/HDMI cables. Nonetheless, we must be aware of the time response of the liquid crystals, which is characterised by the rise and fall times when stimulated by a signal equivalent to 2π phase change. The response time is defined as the switching time from 10% to 90% and from 90% to 10 % (rise and fall time). These time constants are determined by the properties of the liquid crystal material, the thickness of the LC layer, the used drive sequence / calibration (the actual voltages applied to a pixel) and temperature. For phase SLMs the response time typically is below the input frame rate*. In our case, the documented rise and fall times are 50ms and 65ms, measured for phase shift to 2π with a 633nm beam. This allows us to estimate the bandwidth as BW[Hz] $= \frac{0.35}{\text{RT[ms]}} \times 10^3$ as $\approx$ 7.0Hz or $\approx$ 5.4Hz (using rise and fall times respectively). We've found that with standard GUI python library PyQt5, there needs to be a wait time of 300 to 400ms between frames so as to ensure an effective frame update. This brings the effective refresh rate to 2.5Hz, which, when adding latency from the addressing signal, python interface and the fact that we're using a 532nm laser, is well within expected. Another important note to take into account is that these displays are not designed for high power laser light. This model specifically is limited to 2 W/cm^2 of incident irradiance, for incident light in the 420nm-650nm range [82].

FIGURE B.4: SLM Pluto 2 phase-only spatial light modulator. Adapted from [82].

*Information taken from email exchange with Holoeye support service.

Characterization and calibration

As we've seen in previous section, liquid crystals allow for very good phase manipulations assuming the correct incident polarisation. For the Pluto 2 VIS-016, this polarisation is recommended to be parallel to the longer side of the display, however, in the absence of a reference polariser, this proved itself to be a challenging task on its own.

Moreover, due to the digital nature of the driving scheme, the different phase levels are created by pulse code modulation, which enables a reliable small sized and cost-effective driver unit since only two voltages need to be generated. However, this efficiency comes at the cost of phase instabilities (the so called "phase flickering") introduced due to the pulse code. These instabilities arise from the limited viscosity of the liquid crystal molecules which doesn't allow for the molecules to follow the pulses instantaneously. This means that the LC molecules flicker around the average value of the pulse code per addressed phase level. Fortunately, Holoeye provides a few configuration files (in the following called "Sequence") that mainly differ in their total bit-depth. For the present device the recommended sequence is the "5-6" sequence which was reported to have a phase flicker of less than 0.15rad for a 2π modulation for incident light at 633nm [82], as can be seen in figure B.5.

Finally, it's worth mentioning that, due to manufacturing processes on wafer scale, single LCOS cells may show a residual bending of their backplane as a result of the dicing process. The deformation has been reduced to less than 1μm, however, it's not perfect. It is recommended to do a first study on this deformation to find a phase mask that will correct the wavefront. A standard practice is to mount the device on an interferometric set-up and find the correct aberration correction phase masks by fringe inspection. However, this was not explored in the present work.

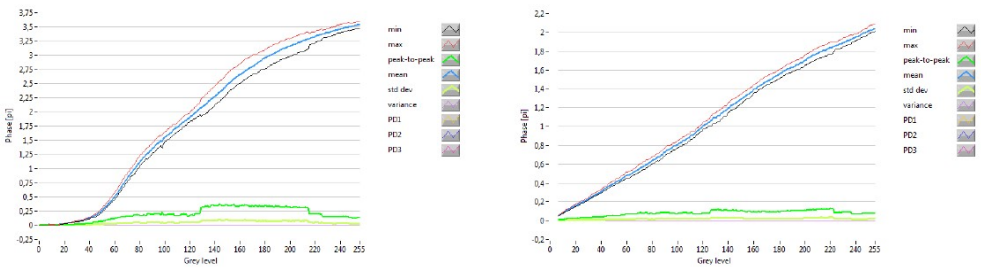

FIGURE B.5: Typical flicker of the 5-6 sequence measured at 633 nm for default voltages and voltages for 2π modulation at 633 nm. Adapted from [82].

Phase calibration

Ideally, the phase calibration would be set up to:

$$\text{Grayscale level}(\phi) = \frac{\phi}{2\pi} \times 255 \tag{B.1}$$

However, the relationship voltage-phase (and consequently grayscale-phase) is non-linear, and so a calibration curve must be used. Luckily, Holoeye also provides some calibration curves (called "gamma-curves") that are specific to the wavelength, device and sequence used that allow for various ranges of modulation. However, even though these gamma curves provide good linear modulations as per visual inspection, it was noted some discrepancies that were not compatible with a 0 to 2π modulation, hence, an experimental verification was called for.

Experimental set-up

The experimental set-up follows work done in reference [83], which is largely based on Young's fringes, and can be seen in figure B.6. For the experiment we've used a 532 nm laser of 50mW, however, both for safety and light control, it is immediately followed by a linear polariser which also controls the power across the optical path due to the laser's native polarisation. The beam then passes through a rotatable half-wave plate, which permits dynamic control of the polarisation, and is followed by a 40x objective and a 15μm pinhole at the Fourier plane, to act as a spatial filter, and it's finally colimated by a simple lens. The core of the experiment lies in the amplitude mask with two pinholes, which have to be as similar as possible to allow for a good contrast at the interference plane. The light emerging from said pinholes will be modulated by an Airy function, but the central lobe of the propagating light has an almost constant phase and can be regarded as a flat top beam. The SLM is separated into two distinct regions, and each is multiplied by a single pinhole. One of the regions is addressed to a uniform grayscale value and is used to modulate the reference beam, while the other is applied a variable grayscale value. These signals are then filtered by a linear polariser which intends to select the portion of light aligned with the extraordinary axis of the LC cells which will be the portion that is phase modulated. Afterwards, the beams are focused by a lens on a digital camera which is used to record the interference pattern at the Fourier plane.

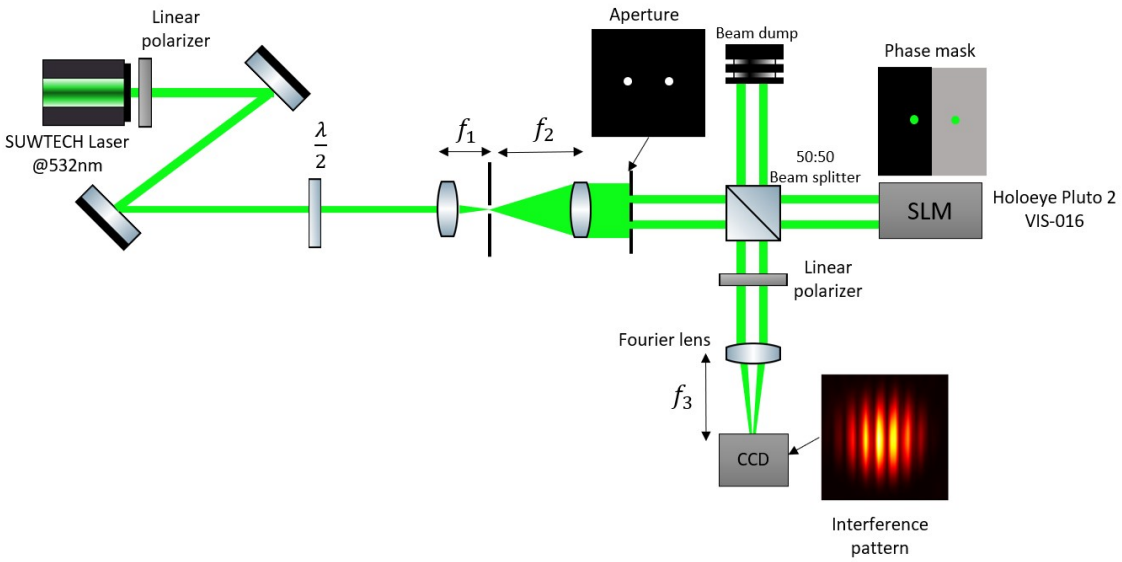

FIGURE B.6: Experimental set-up used for phase calibration procedure.

As is demonstrated in Ref.[83] the interference pattern recorded should follow:

$$|F(\nu_x,\nu_y)|^2 = \left(\frac{a}{r}\right)|J_1(2\pi\nu_r a)|^2 \times \left[(\delta A)^2 + (4A^2 + 4A\delta A)\cos^2\left(\pi\nu_x\Delta x + \frac{\phi}{2}\right)\right] \quad \text{(B.2)}$$

where $2a$ is the pinhole diameter, $r = (x^2+y^2)^{1/2}$ is the radial coordinate defined in the object space at the output of the pinholes, Δx is the distance between the pinholes, δA accounts for the difference in amplitude between the pinholes, A is the amplitude of one of the beams, J_1 is the Bessel function of the first kind of order 1, $\nu_r = (\nu_x^2+\nu_y^2)^{1/2}$ is the radial frequency in the Fourier plane, and finally ϕ is the phase difference between the two beams. In practice, we'll be looking at a small portion of the beams, so $\delta A \approx 0$, and due to its symmetry we can also just look at a slice of the interference pattern, thus the fit can be done computationally to the function:

$$f(x,\alpha,A,B,C,D) = B\frac{|J_1(Ax)|^2}{x^2}\cos^2(Cx - \frac{\alpha}{2}) + D \quad \text{(B.3)}$$

as is demonstrated in the figure B.7.

Experimental results

Using the set-up above we measure a calibration curve for 3 different gamma curves provided by Holoeye as can be seen in figure B.8. All the curves display a remarkable linearity, albeit with some fluctuations which can have many origins ranging from physical

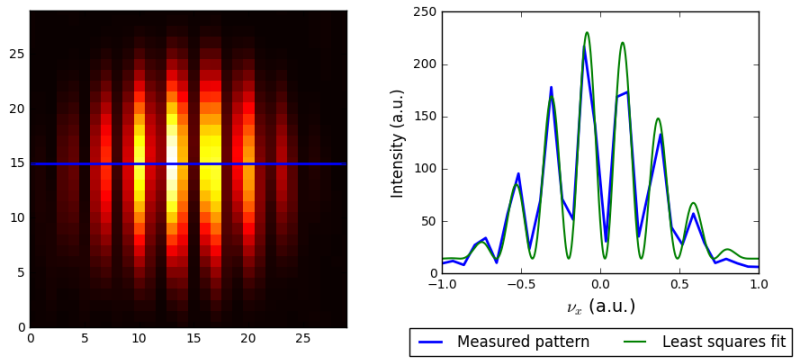

FIGURE B.7: Example of the least squares fit to the fringe interference pattern.

vibrations of the optical table, noise in the digital camera, SLM flickering, stray light, laser source fluctuations, among others. Only the green curve is able to achieve a 2π modulation, which is achieved at a gray level of 231.

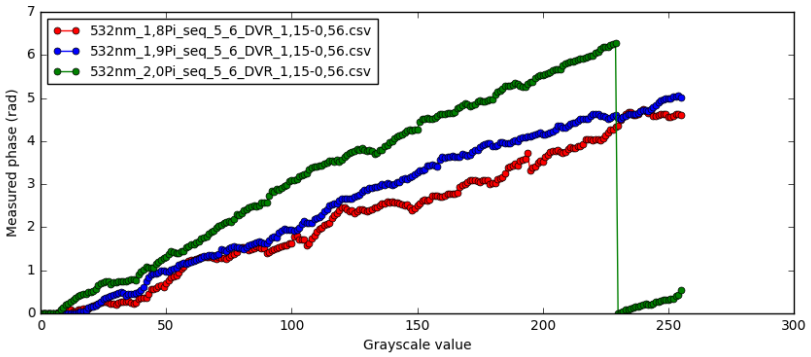

FIGURE B.8: Phase calibration results.

Having this in mind it is sufficient to merely cap the modulation range to 231, thus the calibration curve should be

$$\text{Grayscale level}(\phi) = \text{integer}\left(\frac{\phi}{2\pi} \times 231\right) \tag{B.4}$$

With the same set-up we've also retrieved results on amplitude modulation, where we've observed a maximum of 6% variation of the intensity for the different grayscale levels, which is deemed acceptable for our experimental purposes.

Appendix C

Off-axis digital holography for wavefront phase retrieval, and a real time phase retrieval python sofware

Introduction

The word *Hologram* has been greatly popularized by Sci-Fi movies, and it turns out that one of the earliest references to such a technology can be traced back to Isaac Asimov sci-fi novel series "The foundation Trilogy" in 1951 [84]. In fact, the invention of the field of holography is often credited to Dennis Garbor in 1948 [85], and in 1949 he publishes a 33-page paper entitled *"Microscopy by reconstructed wave-fronts"* where he's able to sum up the working principle of the technique in the abstract:

"The subject of this paper is a new two-step method of optical imagery. In a first step the object is illuminated with a coherent monochromatic wave, and the diffraction pattern resulting from the interference of the coherent secondary wave issuing from the object with the strong, coherent background is recorded on a photographic plate. If the photographic plate, suitably processed, is replaced in the original position and illuminated with the coherent background alone, an image of the object will appear behind it, in the original position" - Retrieved from Ref. [86]

It's interesting to note that this work was brought forth before the advent of the laser in 1960, which proved itself to be a remarkable propeller of holography that saw its first work with a laser light source in 1962 [87].

How does holography work?

A conventional photograph is a two-dimensional projection of a three-dimensional scene, and as a result we lose information that enables us to perceive depth and parallax with which we experience the real world. Looking at how an image is recorded, be it in a photographic film or in a modern CCD camera array, we see that the pattern that is imprinted in them is the average intensity of the light that reaches it, and there's no information about the phase of the wavefront that was focused. In contrast, a hologram aims to keep all this information through interference effects, and upon reconstruction allow us to view "the whole image" (as the etymology of "hologram" suggests).

Let us do a quick quantitative analysis of an off-axis technique (an on-axis technique would differ in the orientation of the diffused beam from the object relative to the reference beam, which would be colinear).

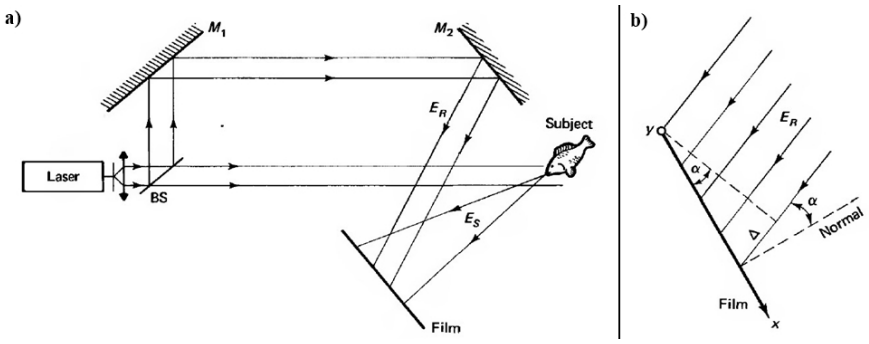

FIGURE C.1: a) Off-Axis holographic system. b) Orientation of film with reference beam. Diagrams taken from Ref.[87]

Suppose that the reference beam at the film can be represented by

$$E_R(t) = r(x,y)e^{i(\omega t+\phi)} \tag{C.1}$$

where $\phi = kx\sin\alpha$. Likewise, the beam that bounces back from our subject can be written as

$$E_S(t) = s(x,y)e^{i(\omega t+\theta)} \tag{C.2}$$

where θ is some complicated function that arises from the reflection on the subject's surface. The total electric field then is simply $E_F = E_R + E_S$. The irradiance pattern that is

imprinted on the film is proportion to $|E_F|^2$

$$I_F \propto |E_F|^2 = (E_R + E_S)(E_R^* + E_S^*) \tag{C.3}$$

$$= E_R E_R^* + E_S E_S^* + E_R E_S^* + E_R^* E_S \tag{C.4}$$

$$= r^2 + s^2 + rse^{i(\theta-\phi)} + rse^{-i(\theta-\phi)} \tag{C.5}$$

We see that the imprinted pattern preserves some phase information. With I_F imprinted on some photosensitive surface, we can use it now as a transmission mask for a certain reconstruction reference beam, which we choose it to be the same reference beam used for the making of the hologram. Therefore, the transmitted beam will take the form

$$E_H \propto I_F E_R = (r^2 + s^2)E_R + r^2 s e^{i\omega t+\theta} + r^2 e^{i\phi} s e^{i(\omega t-\theta)} = E_{H1} + E_{H2} + E_{H3} \tag{C.6}$$

The first term, E_{H1}, is the reference beam modulated in amplitude but not in phase, which is also sometimes called *zeroth-order diffraction*. The second term, E_{H2} is a beam that travels with an angle α relative to the reference, and as it is nothing more that an amplitude modulated subject beam, it appears to come from the subject itself giving us the perception of depth due to the virtual image. Finally, the third term, E_{H3}, is the same subject beam, but with an added phase of 2ϕ that translates to an added wavevector component to the transmitted wave, and the phase of the subject now comes reversed. This phase reversal forces diverging rays to become convergent and thus forms a real image. This process is schematically demonstrated in Fig.C.3

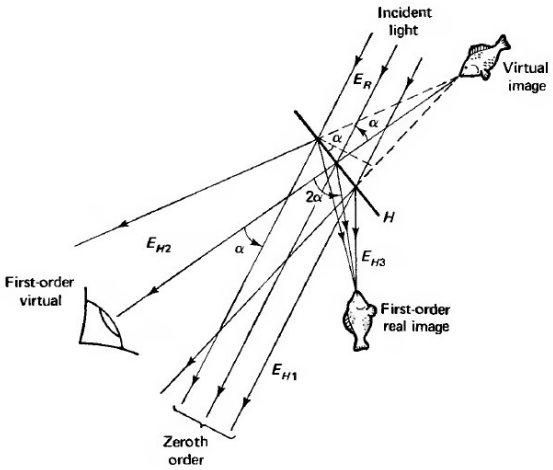

FIGURE C.2: Reconstruction of the hologram formed in Fig.C.1. Diagrams taken from Ref.[87]

Digital holography

As we've seen on the last subsection, holography is a two step-process: creating the hologram and reconstruction. It is unquestionable that the invention of the laser has made this process easily accessible. However, along the years several new technologies were developed, and amongst those are the electronic displays such as a CRT display, which constitutes the basis of the earliest work on digital photography by replacing the photographic film with such display, followed by an optical reconstruction of the image [88, 89]. Furthermore, in 1965, Cooley and Tukey published the well known Fast Fourier Transform algorithm, which allowed an unprecedented speed of computation of the Fourier transform, opening many possibilities within digital holography by employing novel methods of digital image processing. However, at the time they were still limited to low resolution detectors, but nowadays, due to the advancements of the semiconductor industry, we can use high resolution CCD and CMOS detectors with arrays of thousands of pixels. Nowadays, a typical approach consists of performing the hologram acquisition with a digital camera, and to reconstruct the image, we numerically solve the diffraction integral up to a certain distance. This method of reconstruction has the advantage of being cheaper, more accessible and allows for more control of the whole process since all the image process can be done digitally. The major downside is the speed of computation since it will always be slower than the analog computation. In our case we are not interested in image reconstruction and, therefore, refer to Ref.[90] for a comparative study on digital holographic and reconstruction techniques, and to Ref. [91] for a *"General theoretical formulation of image formation in digital Fresnel holography"*.

Nonetheless, it is of practical importance to recall that this is a digital process, and as such is subject to sampling. In fact the sampled signal at the camera would correctly be described as

$$I_{Fs} = \sum_{k=0}^{N_x-1} \sum_{l=0}^{N_y-1} I_F(k\Delta x, l\Delta y)\delta(x - k\Delta x)\delta(y - l\Delta y) \tag{C.7}$$

where it was assumed that $I(x, y)$ does not vary significantly within a pixel. Since our goal is to correctly measure the interference fringes, we must define our maximum spatial frequency for them. The fringes will have a spatial angular frequency $k = \frac{2\pi}{\lambda} \sin\alpha$, and

according to the sampling theorem we must have

$$k < \frac{k_s}{2} \tag{C.8}$$

$$\frac{2\pi}{\lambda}\sin\alpha < \frac{2\pi}{2\Delta x} \tag{C.9}$$

$$\boxed{\alpha < \sin^{-1}\frac{\lambda}{2\Delta x}} \tag{C.10}$$

Note that if we were to use a He-Ne laser at at 632nm with a standard digital camera with a pixel size of about 5 micrometers, we would be left with $\alpha < 3.62^{o}$. Of course, the same principle applies to any transverse spatial frequency that the wavefront may carry, that is, as $\mathbf{k}_\perp = \nabla_\perp \varphi$, where $\varphi = \mathbf{k}\cdot\mathbf{r} - \omega t$, we must have $(\mathbf{k}_\perp)_i < \frac{2\pi}{2\Delta i}$ with $i = x, y$.

Digital holography for wavefront phase retrieval - experimental results

In this section we explain the working principle of digital off-axis Fresnel holography to retrieve the phase profile of the wavefront as it reaches the camera. The following treatment follows closely the works done by Cuche et al. [92].

Working principle

Let us assume a plane wave treatment and write our reference wave as $R = \sqrt{I_R}e^{ik\sin\alpha x}$. The hologram intensity then becomes

$$I_H(x,y) = I_r + I_o + \sqrt{I_R}e^{ik\sin\alpha x}\mathbf{O} + \sqrt{I_R}e^{-ik\sin\alpha x}\mathbf{O}^* \tag{C.11}$$

Where $\mathbf{O}$ is the field from the object beam. Taking the Fourier transform of I_H we have

$$\begin{aligned}\tilde{I_H}(k_x,k_y) = \tilde{I_R} + \tilde{I_0}(k_x,k_y)&\\ + \sqrt{I_R}\mathcal{F}\{e^{ik\sin\alpha x}\} * \tilde{\mathbf{O}}(k_x,k_y)&\\ + \sqrt{I_R}\mathcal{F}\{e^{-ik\sin\alpha x}\} * \tilde{\mathbf{O}}^*(k_x,k_y)&\end{aligned} \tag{C.12}$$

Since $\mathcal{F}\{e^{ik\sin\alpha x}\} = 2\pi\delta(k - k\sin\alpha)$ and from the properties of the dirac-delta function $\tilde{X}(k) * \delta(k - k_0) = \tilde{X}(k - k_0)$ we get:

$$\begin{aligned}\tilde{I_H}(k_x, k_y) = \tilde{I_R} + \tilde{I_0}(k_x, k_y) \\ + 2\pi\sqrt{I_R}\Big(\tilde{\mathbf{O}}(k_x - k\sin\alpha, k_y) \\ + \tilde{\mathbf{O}}^*(-k_x + k\sin\alpha, k_y)\Big)\end{aligned} \tag{C.13}$$

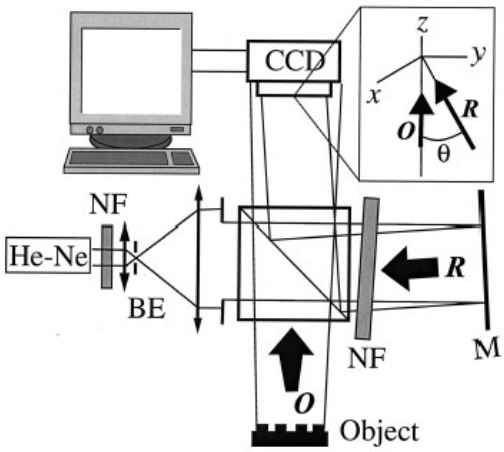

FIGURE C.3: Experimental setup: BE, beam expander;NF, neutral density filter; **M**, mirror; **O**, object wave;**R**, reference wave. Inset, detail of the off-axis geometry. Diagram taken from [92]

Looking at eq.C.13 we see that if we apply a 2-dimensional bandpass filter centered at $k_x = k\sin\alpha$ and $k_y = 0$, with a width large enough to admit all the bandwidth of the signal O, we are able to isolate $\tilde{O}(k_x - k\sin\alpha, k_y)$. Since all of the signal processing is done computationally, all we have to do next is translate our signal to the center of the fourier plane and perfom an inverse FFT, and we recover $O(x, y)$.

Preliminary results

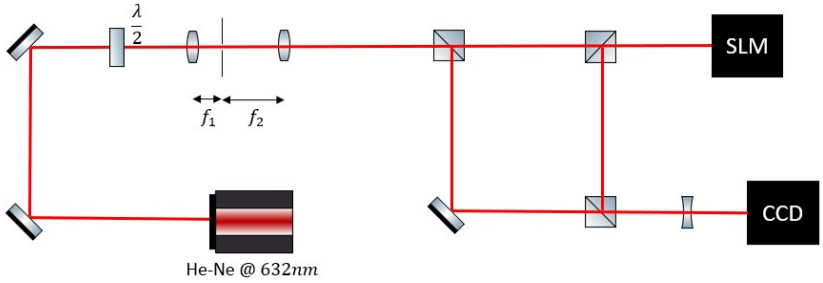

FIGURE C.4: Experimental setup used for early measurements.

With the setup shown in figure C.4 we were able to perform some measurements to prove the validity of this technique. We use a He-Ne laser at 632nm, which passes through a confocal lens system where it is placed a spatial filter so as to guarantee a gaussian wavefront. At the same time, the lens system also works as a beam expander, thus making use of the maximum number of pixels of the SLM as possible. The SLM used is a LC-R 2500 by Holoeye with a resolution of 1024×768 pixels which is controlled by a custom software that allows for arbitrary control of the phase mask. Finally, the digital camera is a BOSCH Dinion Fx LTC0495/50, with an effective resolution of 616×474 pixels. The two interfering beams are expanded by a diverging lens before the acquisition by the camera to guarantee that the reference beam resembles as closely as possible a plane wavefront. However, it should be noted that this method, even though valid, also expands the beam coming from the SLM and plenty of information is lost by the recording. A better approach would be to add a beam reduction stage to the object beam, hence guaranteeing that the radius of curvature of the reference is much greater than the object beam.

We test our set-up and algorithms with four different phase masks: i) single vortice with positive circulation, ii) single vortice with negative circulatio, iii) sea of vortices of 1x1 and iv) sea of vortices 2x2. The phase masks "sea of vortices nxm" are obtained by superimposing n single vortice masks with positive circulation and m single vortice masks with negative circulation. The results shown below do not account for a calibration of the SLM.

Single vortice mask with positive circulation

The phase mask given to the SLM is obtained by

$$\theta = +\arctan\left(\frac{x - x_0}{y - y_0}\right) \tag{C.14}$$

where (x_0, y_0) is the phase singularity location. As can be seen in figure C.5, we can identify the presence of the phase singularity by direct inspection of the branching of the interference fringes.

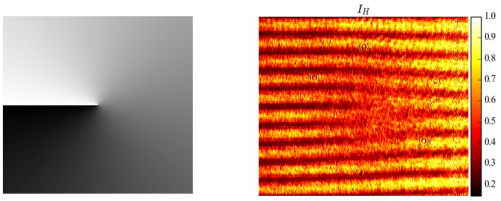

FIGURE C.5: Left: phase mask passed to the SLM screen. Right: image recorded on the digital camera.

After performing the FFT of I_H we can clearly see from figure C.6 the three peaks as predicted by equation C.13. We now have to isolate one of the offset peaks, translate it to the center of the spectrum, and perform an inverse FFT to it. This process is illustrated in Fig.C.6

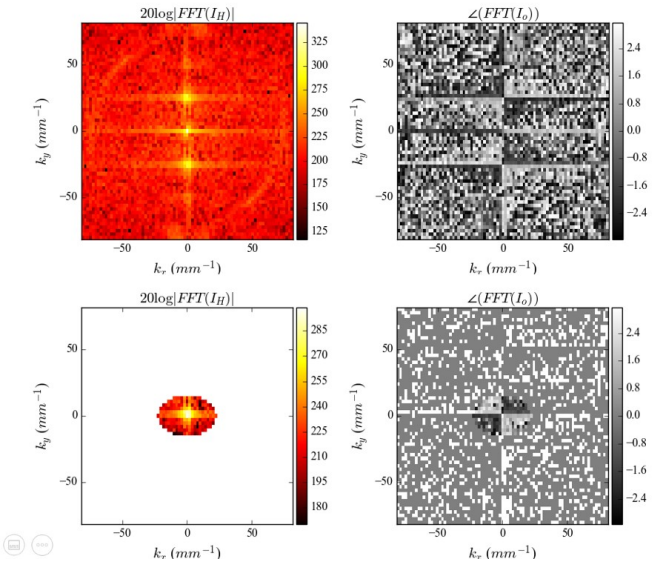

FIGURE C.6: a)Fourier transform of the measured intensity pattern. This image is zoomed in for the region of interest. b)Filtered and translated signal of the object beam.

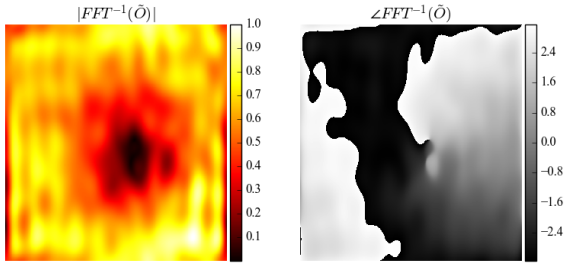

FIGURE C.7: Wavefront reconstruction of the object beam.

From figure C.7 we can clearly see the presence of the phase singularity, characteristic of the vortice phase mask.

In the following subsection we only show the recovered wavefronts, without reference to the data analysis.

Single vortice with negative circulation

The phase mask given to the SLM is obtained by

$$\theta = -\arctan\left(\frac{x - x_0}{y - y_0}\right) \tag{C.15}$$

The effect of the circulation is visible on the direction of the splitting of the fringes. As can be seen in figure C.8, the splitting occurs in the opposite direction as in figure C.5.

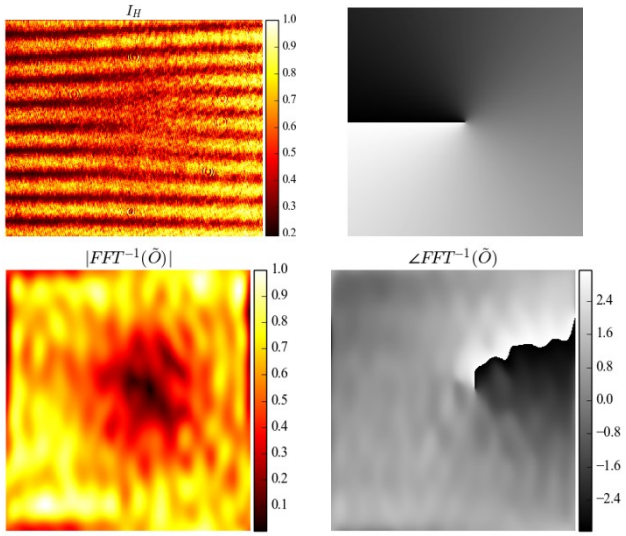

FIGURE C.8: Wavefront reconstruction of the object beam for a single vortice mask with negative circulation. Top: raw data (right) and mask given to the SLM (left). Bottom: wavefront reconstruction.

Sea of vortices 1x1

In this case the phase mask passed to the SLM is given by:

$$\theta = \left(\sum_{i=0}^{N_+ - 1} arctan\left(\frac{x - x_i}{y - y_i}\right) - \sum_{i=0}^{N_- - 1} arctan\left(\frac{x - x_i}{y - y_i}\right)\right) \mod (2\pi) \tag{C.16}$$

Where the positions (x_i, y_i) are randomly generated within a predetermined area. From figure C.9 we see that we were able to correctly reconstruct the two phase singularities.

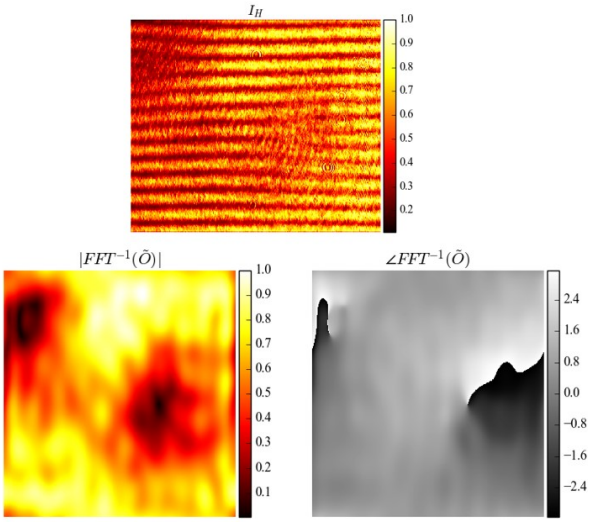

FIGURE C.9: Wavefront reconstruction of the object beam for a sea of vortices with $N_+ = N_- = 1$. Top: raw data. Bottom: wavefront reconstruction.

Sea of vortices 2x2

This test aims to evaluate the resolution of our camera. From figure C.10 we note that it becomes difficult to pinpoint the locations of the 4 singularities. This result needs more study.

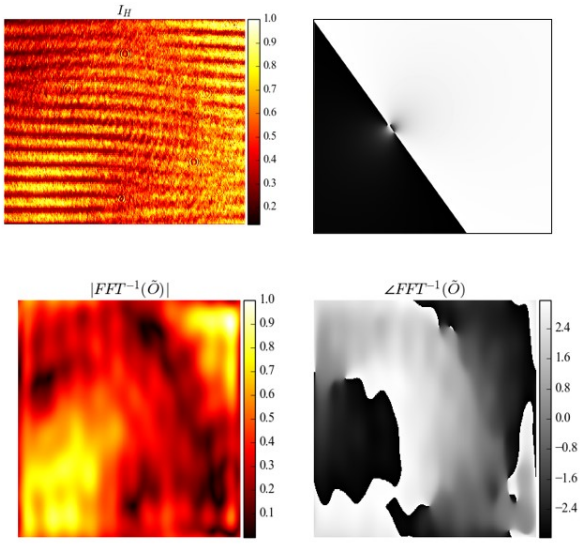

FIGURE C.10: Wavefront reconstruction of the object beam for a sea of vortices with $N_+ = N_- = 2$. Top: raw data. Bottom: wavefront reconstruction.

A real-time phase retrieval Python GUI based on off-axis digital holography

As we've seen, phase retrieval through off-axis digital holography can be quite powerful. In experiments where it is necessary to have full control and knowledge of the complex electric field, it be quite handy. Within our research group, this technique has been of extensively studied and used in a free space optics experiment on quantum fluids of light in photorefractive media. In such an experiment, being aware of the complex wavefront during optical alignment can greatly ease the task and lead better results. To this end, we've decided to create a Python GUI that would implement the algorithm in real time.

For the user interface we've decided to use the PyQt5 python package [93]. PyQt5 is a set of python bindings for Qt5, which is a set of cross-platform C++ libraries that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. As for the plotting engine, we've used PyQtGraph [94], an open-source, pure Python graphics library for PyQt5. It has been developed with scientific, engineering and mathematics uses in mind, and despite being written in Python, it is capable of offering outstanding performance for real-time data analysis. In this section we'll outline the most crucial design aspects of the software, and we'll leave the detailed inner workings of the script for the curious reader, as the source code can be made available.

The front-end of the user interface is shown in figure C.11. It is composed of five main panels: a) it contains a monochromatic view of the image retrieved by the camera. It is assumed the camera module returns a monochromatic image, and in the cases it doesn't, it can always be converted; b) contains the amplitude of the logarithm scale of the fourier transform of the intensity measurement; c) and d) contain the amplitude and phase profiles, respectively, of the reconstruction of the filtered fourier spectrum; e) contains a menu that allows to start the live feed, as well as 4 line plots that allow us to analyse line profiles of each plot. Panels a), b), c) and d) feature a line ROI (Region Of Interest) which allows retrieve the data that lies beneath such line. The respective plots in panel e) are updated in real time. Panel b) has an extra ROI geometry: an ellipse. This ellipse allows us to extract a region of the Fourier spectrum, as in figure C.3, and carry out the phase retrieval algorithm. All the 2D plots are interactive as the images can be zoomed in and out and moved without freezing the application, and also feature an interactive colorbar.

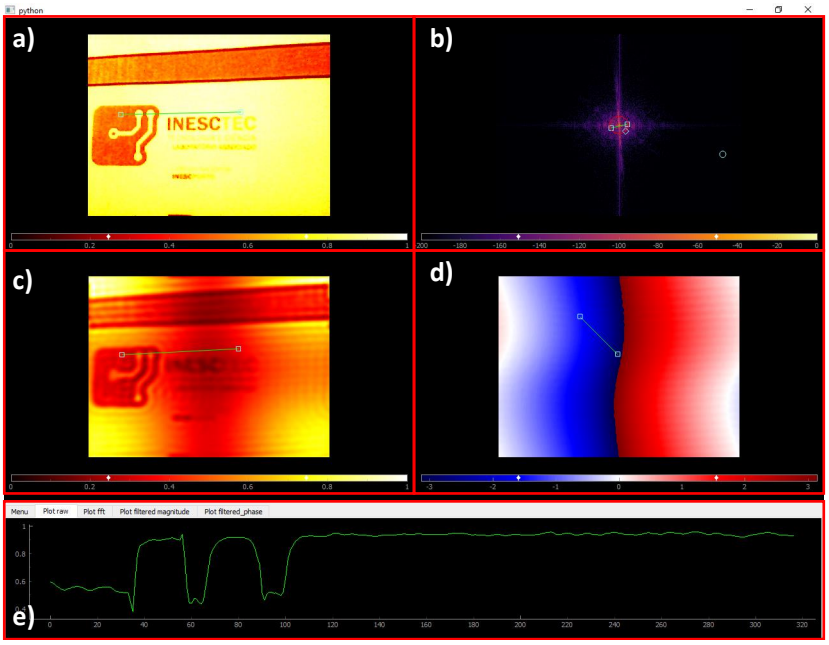

FIGURE C.11: User interface for real-time phase retrieval.

As for the software structure, it was a requirement that the interface remained responsive. To this end, we've decided to make use of concurrent programming by making use of PyQt's threading feature. It is important to recognise the distinction between concurrent and parallel programming. In parallel programming there are several instructions or processes that are carried out simultaneously. In Python, due to its GIL (Global Interpreter Lock), this is not possible*, as it forces a single thread to hold Python's interpreter at a time. Nonetheless, software engineers have been quite smart and came up with concurrent programming which allows instructions from different threads to be cleverly scheduled to increase the overall performance of the interpreter. This way, we can delegate long running tasks for separate threads and free the interpreter for other small workloads to keep a GUI responsive. Having this in mind we have opted for the architecture illustrated in figure C.12. In the main thread, we keep all the widgets necessary for visualisation and respective signals and slots properly connected. Within the main thread, we start a secondary thread which will be responsible for sending new images to update on panels a), b), c) and d). Within this thread, the computer will submit a request to the camera for a new frame, and upon receiving it, the image follows for a data processing module which is runs the phase retrieval algorithm. When all is done, the thread emits a signal† which is captured by the main thread, and then proceeds to update the plots.

*Python does offer a multiprocessing module in which each process is attributed its own GIL. However, such approach was not carried out.

†Here we refer to the native signals and slots mechanism of PyQt5.

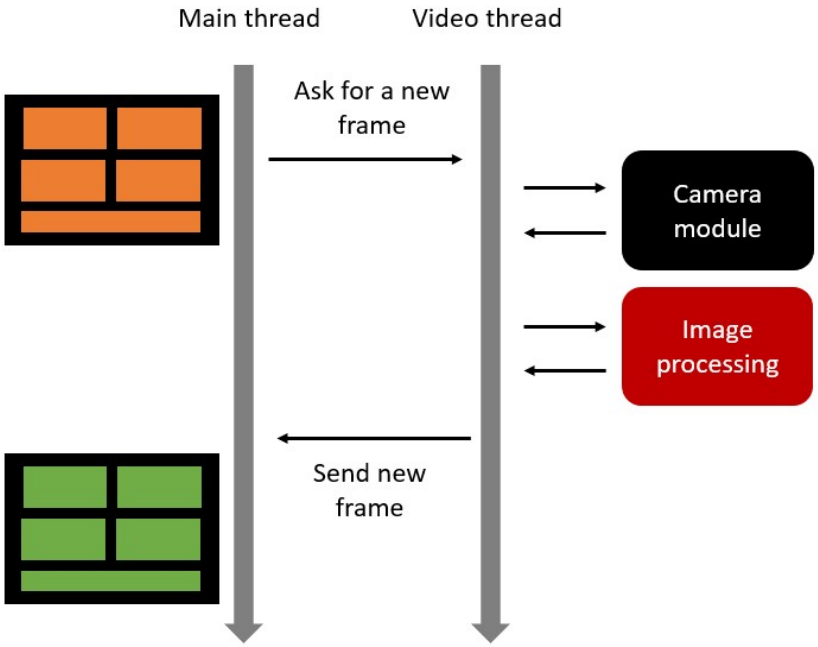

FIGURE C.12: Thread architecture and information flow during a cycle of the application.

Although this software has been built for the specialised task of phase-retrieval, its underlying mechanisms can be applied to any task which requires real time data analysis, and it is compatible with various camera modules as well as data processing routines.

Appendix D

Optical complex media

What is a speckle pattern?

Whenever propagating light encounters a boundary to a medium there will be an interaction between radiation and matter, and we will see some reflected and transmitted light. This interaction is commonly described by simple laws such as Snell's law and the Fresnel's equations. However, these assume homogeneity and isotropy within the media, which greatly simplifies the mathematics and allows us to easily describe the dynamics of the radiation. However, things get more complicated when we look at the interaction of light and small particles. This problem was first studied by Gustav Mie in 1908 [95]. After his discovery, much more effort was put forth by the scientific community in order to understand this effect, and today it is a well established theory, and can be used to explain certain daily phenomena, such as the blue sky during the day and the characteristic red sky during sunsets, as well as the existence of white and dark clouds.

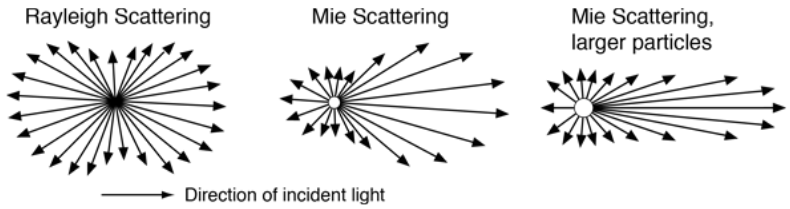

FIGURE D.1: Illustration of light scattering from homogeneous spheres according to Mie solution. For small particles, whose radius is smaller than the incident wavelength, the scattering is well described by the Rayleigh approximation, whereas for particles with a radius larger than the wavelength Mie scattering is predominant. Image taken from Ref. [96].

The takeaway point is: whenever light hits a particle of size comparable or smaller than it's wavelength it emits radiation in many directions with an anisotropy that is dependent of the particle's size.

Now, Let us suppose that we let many of such particles to be close together. When a light wave hits this group of particles, the first ones interacting with light will radiate light in many directions, that in turn will interact with other particles, and so on and so forth. This is the problem of multiple scattering and is present whenever there's a high optical inhomogeneity in the medium. In figure D.2 we can see an illustration of a particular light ray bouncing from scattering particles.

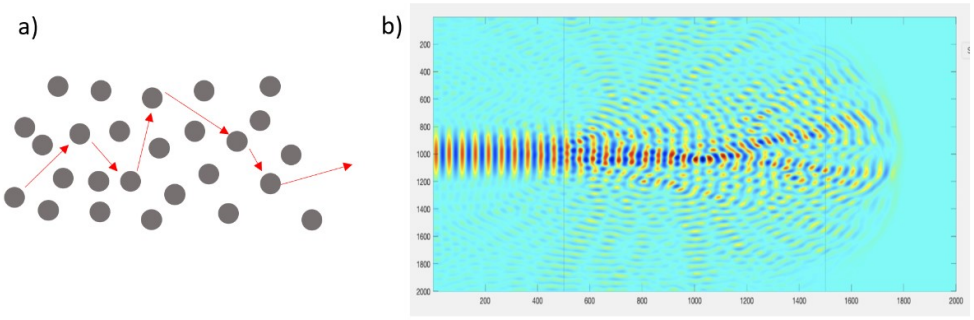

FIGURE D.2: a) Illustration of multiple scattering of a single light ray within a medium; b) Propagation of a coherent beam into a random optical medium. Speckle is intrinsically three dimensional while 2D speckle is the cross section of the light filaments (Image taken from Ref.[97]).

This phenomenon is particularly interesting when we let the incident light have a high coherence such as that of laser light. In fact, in the early 1960s, when continuous-wave lasers first became commercially available, researchers noted that when the light was reflected from a surface such as paper, or the wall of the laboratory, a fine-scale, granular and high contrast pattern would appear to the observer looking at the scattering spot. This pattern became known as "speckle". The origin of these intensity fluctuations was soon recognised to be the random roughness of the surfaces from which light was reflected [98]. Various macroscopic facets of the rough scattering surface contribute randomly phased elementary contributions to the total observed field, and those contributions interfere with one another, thus creating high and low intensity regions. Although this concerns surfaces, the same thing happens for a volumetric scattering regime, such as the one illustrated in figure D.2. The reason being that each exiting light ray has gone through a random path inside the material and thus gives rise to a similar speckle pattern. It is important to note that the speckle pattern is mere consequence of the wave nature of

light and it only visible to us if the incident light is coherent, since it relies on the interference of light. If we were to illuminate an object with incoherent and coherent light, and observe it from a diffusive medium, we would have a very hard time discerning information of such object as can be seen from figure D.3, although this is not the same as saying that the image recovered with laser light has "less information", we merely can't process it with our eyes.

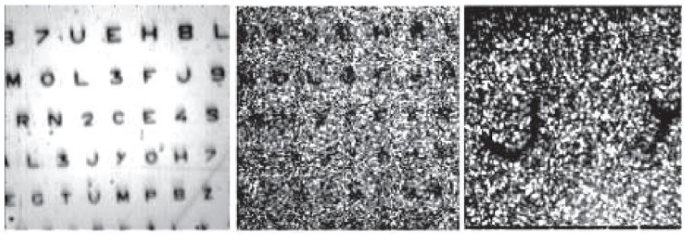

FIGURE D.3: Images of a rough object: (a) image taken with incoherent light; (b) image taken with coherent light; and (c) a magnified portion of the image shown in (b). Image taken from Ref.[62].

Properties of a speckle pattern

We've come to the conclusion that the speckle pattern is a result of interference effects of waves that carry randomly distributed phases and amplitudes. Therefore, if we look at single point in an observation plane, the electric field there is described by a very large sum of random complex phasors as

$$A = \sum_{n=0}^{N} a_n e^{i\phi_n} \tag{D.1}$$

This is the well-known problem of a random walk, as is illustrated in figure D.4, and we already foresee that the way to characterise a speckle is through its statistical properties. However, it is important to stress as of now that, even though the speckle pattern is inherently a statistical problem, the pattern itself is deterministic, and the physics that we're considering so far are linear.

The mathematical details behind the probability distributions that we'll look at are beyond the scope of this document, but for the curious reader we refer to Ref.[62].

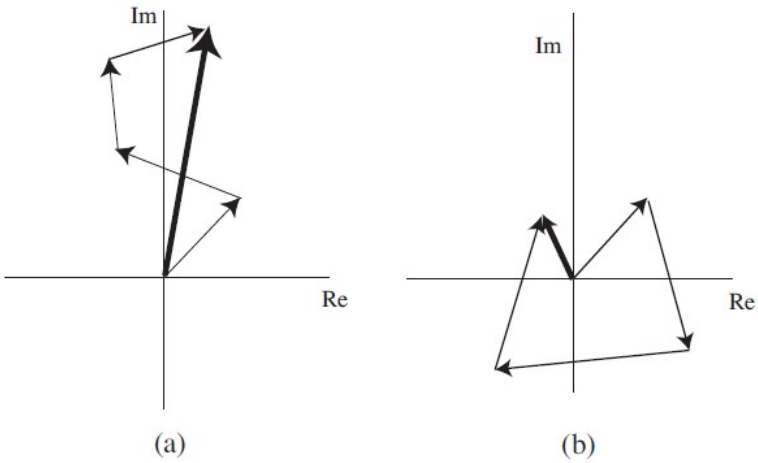

FIGURE D.4: Random walks showing (a) largely constructive addition and (b) largely destructive addition. Image taken from Ref.[62].

Random phasor sum

We'll start looking at the results for a simple random phasor sum. This regime may seem odd because there can be no speckle without an incident laser beam which is in itself a complex wave*, therefore we should consider a sum with a known phasor. However, if light propagates inside a medium for long enough there will be no ballistic photons left and any information of the incident beam is essentially lost, and the solution approaches the results below. For mathematical simplicity let us assume that:

1. a_n and ϕ_n are statistically independent of a_m and ϕ_m;
2. a_n and ϕ_n are statistically independent;
3. ϕ_m follows a uniform distribution from $(-\pi, \pi)$;
4. The number of scattering steps approaches infinity. This ensures the validity of the central limit theorem.

With these assumptions it can be shown that the real and imaginary parts of the resultant complex phasor follows a joint probability distribution as:

$$p_{\mathcal{R},\mathcal{I}}(\mathcal{R},\mathcal{I}) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{\mathcal{R}^2+\mathcal{I}^2}{2\sigma^2}\right) \tag{D.2}$$

*The electric field is in fact a real quantity, but we're working with the complex exponential formalism as is employed in Ref.[62].

Because of the circular symmetry we say that the resultant complex phasor **A** is a "circular" complex Gaussian variate. With this result, it is just a matter of change of variables to show that:

$$p_A(A) = \frac{A}{\sigma^2}\exp\left(-\frac{A^2}{2\sigma^2}\right) \qquad \text{Field amplitude PDF} \tag{D.3}$$

$$p_\theta(\theta) = \frac{1}{2\pi} \qquad \text{Phase PDF} \tag{D.4}$$

$$p_I(I) = \frac{1}{2\sigma^2}\exp\left(-\frac{I}{2\sigma^2}\right) \qquad \text{Intensity PDF} \tag{D.5}$$

We can see that the field amplitude distribution is Rayleigh distributed, whereas the intensity follows a negative exponential, which are all characteristics of a so-called "fully developed speckle". One should notice at this point that if a speckle pattern does not have these properties, then it is proof that the initial assumptions are not met. This implies that there are unknown correlations between the phases and amplitudes of individual contributions.

Sums of speckles and polarisation

Another important aspect to highlight is that the intensity distribution of two independent fully developed speckle patterns can be written as:

$$p_s(I_s) = \frac{1}{\bar{I}_1 - \bar{I}_2}\left[\exp\left(-\frac{I_s}{\bar{I}_1}\right) - \exp\left(-\frac{I_s}{\bar{I}_2}\right)\right] \qquad , \bar{I}_1 > \bar{I}_2 \tag{D.6}$$

$$p_s(I_s) = \frac{I_s}{\bar{I}^2}\exp\left(-\frac{I_s}{\bar{I}}\right) \qquad , \bar{I}_1 = \bar{I}_2 = \bar{I} \tag{D.7}$$

Furthermore, the scattering process greatly influences the polarisation properties of the outgoing speckle pattern as can be seen in figure D.5. Thus, the image that we're in fact capturing at the output of a scattering medium is the sum of two speckles.

Speckle size and imaging a speckle pattern

A common way to quantify the average size of a speckle is through the equivalent area of the normalised covariance function of speckle intensity, which is called the "correlation area" or the "coherence area" represented by $\mathcal{A}_\rfloor$ [62]:

$$\mathcal{A}_\rfloor = \int c_I(\Delta x, \Delta y) d\Delta x d\Delta y \tag{D.8}$$

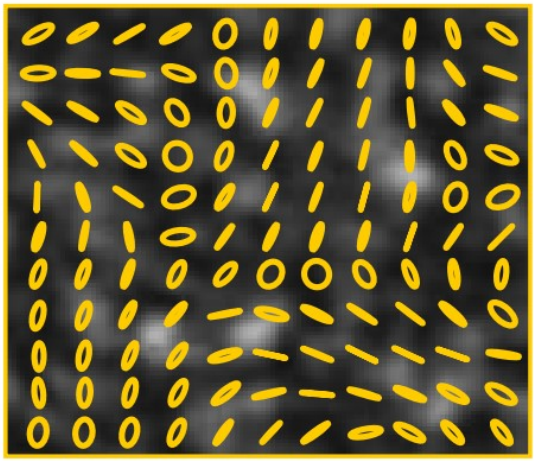

FIGURE D.5: Example of polarisation speckle. Image taken from Ref.[99].

where c_I is the normalised covariance function of speckle intensity, which can be written as:

$$c_I(x,y) = \frac{\langle I(x_1,y_1)I(0,0)\rangle - \langle I(x,y)\rangle^2}{\langle I(x,y)^2\rangle - \langle I(x,y)\rangle^2} \tag{D.9}$$

where we've written $\Delta x = x - 0$ and $\Delta y = y - 0$, so as to take the centre as reference, and $\langle\cdot\rangle$ is the spatial average. Of experimental interest is noting that $\langle I(x_1,y_1)I(0,0)\rangle$ is the auto-correlation function and as implied by the Wiener-Khintchine theorem, the auto-correlation function of the intensity is given by the Inverse Fourier Transform ($\mathcal{F}^{-1}$) of the Power Spectral Density (PSD) of the intensity:

$$\langle I(x_1,y_1)I(0,0)\rangle = \mathcal{F}^{-1}\left\{|\mathcal{F}\left\{I(x,y)\right\}|^2\right\} \tag{D.10}$$

For a region that is illuminated by a circular and constant beam of light, the speckle size can be simplified to:

$$L = \frac{\lambda z}{A} \tag{D.11}$$

where λ is the incident radiation wavelength, z is the distance to the observation plane and A is the area illuminated. It can be shown that in an imaging configuration equation D.11 still holds if we substitute $z \to z_i$ (see figure D.6).

Now, in many experimental set-ups we will see an imaging set-up, so it is worthwhile to spend a bit of time understanding what implications can this have. The first question to answer is: what are we imaging? When we place an imaging lens there is always an object plane and an observation plane. However, in an experimental set-up we want to fix the

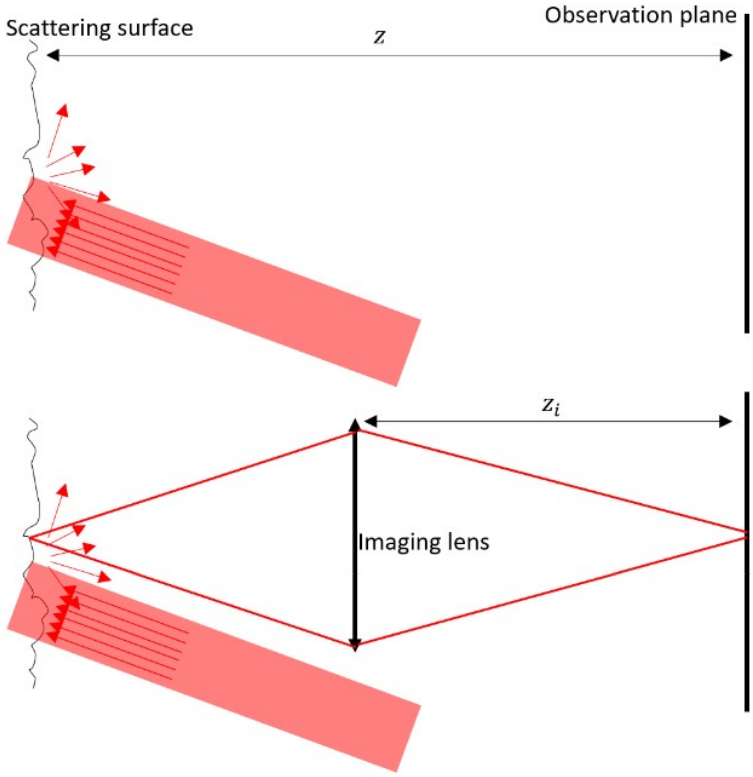

FIGURE D.6: Speckle imaging configurations: free space configuration (top) and imaging configuration (bottom)

observation plane by fixing a recording camera, typically an electronic digital camera, and then we adjust the imaging lens. Let us suppose that we are in fact in an focused exactly in the scattering spot. In that case it can be shown that the speckle pattern recorded will no longer, in general, have the circular gaussian statistics, however these new correlations stem only from the set-up and do not imply the presence of e.g. ballistic photons or undesired non-linearities. In an out-of-focus situation as those shown in figure D.7, the circularity of the statistics is recovered. This fact then allows us to distinguish between correlations originating from the scattering media or from the set-up.

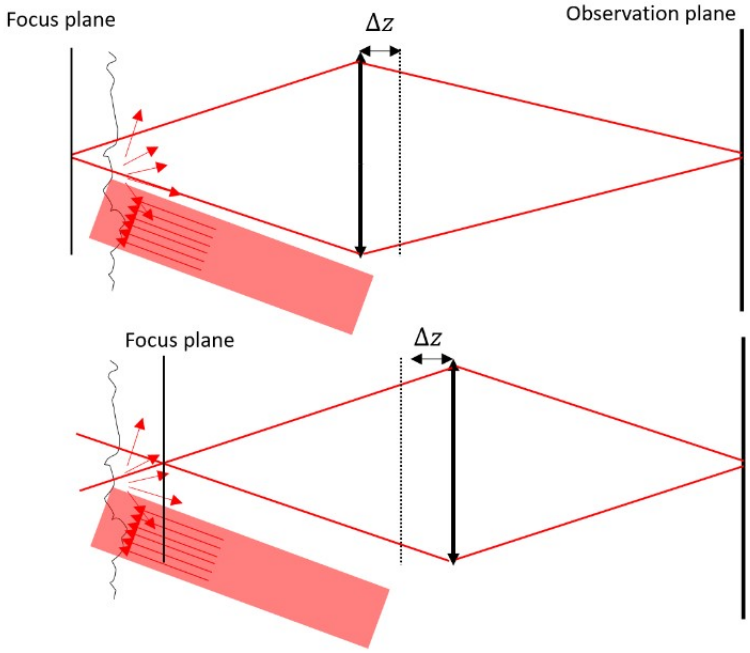

FIGURE D.7: Speckle imaging configurations out of focus.

Appendix E

Transmission matrix measurement

Introduction

The transmission matrix approach was first proposed as a tool for the study of complex optical media by Popoff et. al in 2010 [63, 100]. It as an approach that relies on the linearity of light propagation, and thus can connect input modes to output modes through a linear transformation. Let us suppose that we sample our input space into N modes, E_{in}^{n}, and sample the output space with M modes, E_{out}^{m}. In that case, we can write:

$$E_{out}^{m} = \sum_{n=0}^{N-1} k_{mn} E_{in}^{n} \tag{E.1}$$

How to sample the transmission matrix?

An important question to address here is what we mean by "sampling the input(output) space". Let's focus on free space propagation. There, the propagation eigenmodes are plane waves characterised by a vector $\mathbf{k}$ as $\mathbf{E_k}(\mathbf{r}) = \mathbf{E_0} \exp\left(-j\mathbf{k}\cdot\mathbf{r}\right)$, where $\mathbf{E_k}\cdot\mathbf{E_{k'}} \propto \delta(\mathbf{k}-\mathbf{k}')$, therefore, in order to measure a complete transmission matrix in free space we'd have to give every possible value of $\mathbf{k}$ as input to our media. In addition to this, we'd have to factor in both polarisations of the incoming light and measure both responses. All of this makes the task of measuring the full transmission matrix in free space propagation a highly difficult task, even though there have been some attempts to achieve such feat [101]. The situation gets more friendly when we deal with guided light, such as multi-mode fibres. There, the high number of allowed propagation modes makes the propagation of light seemingly chaotic due to the coupling between different modes, however, the number of modes is finite and discrete [102, 103].

How many modes are present in the outgoing light of a scatterer?

Despite this, it is worth spending some time trying to answer the question "How many modes should there be on outgoing light from a scatterer?". This is a question of fundamental and practical relevance. The first one tells us more about the process of light scattering, while the second one is less obvious, but considering the advances of wavefront shaping technologies, specifically in spatial light modulators (SLM), we can have a particular static configuration of the scatterer and the beam, and control only the input light. Thus, if we know how many outgoing modes are present for a particular beam and scatterer, we have a better chance of estimating the efforts needed to build a (near-) complete transmission matrix. For example, let us say that we have some way of knowing that a particular set-up allows M outgoing modes. Then, due to it's linearity, if we insert M orthogonal input modes, the scatterer is reduced to a simple change of basis with a square transmission matrix. Surely, the matter of actually being able to measure the whole output field is another challenge on it's own, but at least we've reduced the error. With this in mind, at the best of our knowledge, the usual way in literature to estimate such number is [104–106]:

$$N = 2\pi \frac{A}{\lambda^2} \tag{E.2}$$

Where A is the area of the scatterer being illuminated and λ the wavelength of the incident light. However, this expression is presented without any explanation as to why it should be like this other than the fact that the number of outgoing modes is the same as that of a waveguide with an equivalent area. Such explanation lacks physical intuition, and can be complemented by the work done by Winkler et al. in 1994 [107] where he explains the scattering process as a coupling process to higher spatial degree modes. The orthonormal modes of an optical cavity can be found in many textbooks [108], and in one dimension, when normalised appropriately, can be written as:

$$\psi_n(u) = \exp(-u^2/2) \frac{H_n(u)}{\sqrt{n! 2^n \sqrt{pi}}} \tag{E.3}$$

The beam radius is called w, the transverse coordinate u is normalised to $w/\sqrt{2}$ and is dimensionless, and H_n are the hermitian polynomials. The fundamental mode is given by:

$$\psi_0(u) = \frac{\exp(-u^2/2)}{\pi^{\frac{1}{4}}} \tag{E.4}$$

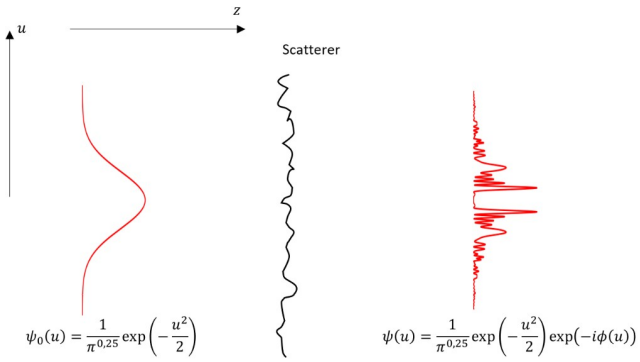

FIGURE E.1: Illustration of the scattering process.

The goal here is to look at the refractive index inhomogeneities, $\delta n(\mathbf{r})$ as mediators to optical mode coupling. To do so, we assume that the scatterer induces a wavefront deformation given by a complex phase as:

$$\psi(u) = \frac{\exp(-u^2/2)}{\pi^{\frac{1}{4}}} \exp(-i\phi(u)) \tag{E.5}$$

however, $\phi(u)$ can be looked at in the fourier domain, and due to the linearity of the whole process, the end result is just an integral over all possible contributions of the irregularities. Thus, in order to gain some fundamental insight, it is sufficient to look at a single harmonic of the form $\phi(u) = \phi_0 \cos(\Omega u + \varphi)$. Here, $\phi_0 = 2kh_0$, where k is the wavenumber, h_0 is the maximum height of the deformation and the factor of 2 is an homage to the original work looking at reflection of scatterer instead of transmission; $\Omega = \sqrt{(2)}\pi w_0/\Lambda$ where Λ is the spatial frequency of the scatterer harmonic under consideration; φ is meant to account for even and odd harmonic by setting it to 0 or $\pi/2$, respectively. By doing this, we can now calculate the overlap of $\psi(u)$ with any harmonic, $\psi_n(u)$ and estimate the power transferred to any mode. By doing so we arrive at:

$$P_n = \frac{1-(-1)^n \cos(2\varphi)}{2} \frac{\phi_0}{n!} \left(\frac{\Omega^2}{2}\right)^n \exp\left(-\frac{\Omega^2}{2}\right) \tag{E.6}$$

In order to better discuss what consequences equation E.6 can have, let us do a small numerical simulation, which can be seen in figure E.2. We can clearly see that since the distortion is even, only even modes are stimulated. Furthermore, we see that the larger the spot size, the higher the mode indexes that receive energy. By noting that $\Omega \propto \frac{w_0}{\Lambda}$ we see that if we consider the whole range of contributions of Λ from the fourier decomposition of $\phi(u)$, then, due to the steep curvature seen in the figure, we see that high values of w_0 will result in the stimulation of a very high number of modes, whereas a smaller are

guarantees smaller number of modes. This is in accordance to the predictions of equation E.2. Sadly, the number of modes predicted by such equation, even for $w_0 = 10\mu m$ and $\lambda = 532$nm is 7000 modes, which still makes it quite a difficult task. Furthermore, it is interesting to relate this result with equationD.11 which gives the speckle size as $L = \lambda z/A$. Therefore, we see that the larger the average speckle size, the fewer modes are present on the scattered light.

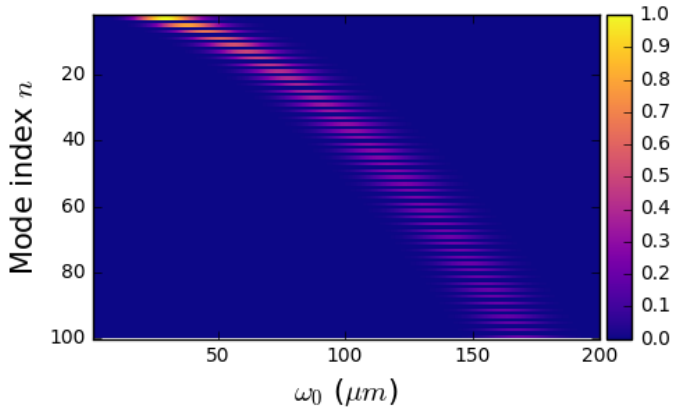

FIGURE E.2: Numerical simulation of equation E.6 for $\lambda = 532$nm, $h_0 = 5\lambda$, $n \in [2, 100]$, $w_0 \in [1, 200]\mu m$, $phi = 0$ and $\Lambda = 100\lambda$. The values are normalized to the highest power transfered to a single mode.

Yet another way to see this area dependence is by mere fourier analysis:

$$E(x, y) = \frac{1}{\sqrt{2\pi}} \int E(k_x, k_y) e^{i\mathbf{k}\cdot\mathbf{r}} d\mathbf{r} \tag{E.7}$$

With this it's clear that in order to have tightly confined light you need to have a large amount of plane waves.

Measuring the transmission matrix

Returning to the problem of sampling the transmission matrix (TM), we now see that we have a good "freedom" as to which incident modes to give to our media because not only we'll deal with an incomplete TM most of the times, but also because we only need to give a number of independent modes and not necessarily orthogonal. Originally, the TM has been sampled on a square basis (the hadamard basis) by means of a phase-only spatial light modulator [100]*, but in principle any set of modes could be used as long as they're mutually independent. Such examples include binary amplitude modulation [109] and

*Note that in this work the input modes are not orthogonal. However, since the basis modulating the phase of the light wave is orthogonal, makes the input modes independent of each other.

phase modulation via a hexagonal lattice [104]. However, this apparent "freedom" of choice of the input modes naturally leads to the question of "How to choose the basis?". This question is an active area of research and to answer it is out of the scope of this document. However, an interesting point to note is that a non-orthogonal input set of modes will lead to a severe distortion of the statistics of the TM, resulting in the occurrence of spurious correlation that manifests itself with notorious high (as well as low) singular values. One approach to solve this issue is a re-sampling of the transmission matrix to an orthogonal set of states [106].

Experimental set-up and acquisition method

The experimental set-up for the transmission matrix measurement is composed of only three particular stages: a wavefront modulation stage, a complex media and a detector array. For this reason, the set-up in chapter 6 was used, but it should be noted that for this specific application it could have been greatly simplified. For the input basis set we've chosen the so-called Hadamard basis functions*. The hadamard matrix of order m is generated by:

$$H_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{E.8}$$

$$H_{m\geq 1} = H_1 \otimes H_{m-1} \tag{E.9}$$

For $m = 3$ we can write:

$$H_3 = \frac{1}{2^{3/2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \tag{E.10}$$

*This set of basis can also be found with the name "Walsh-Hadamard basis". The distinction between them is in the order of the basis vectors, which may be crucial in signal processing applications [110]. However, for our purposes, either is fine.

Now, let $\mathbf{h}_m$ and $\mathbf{h}^n$ be the mth column vector and the nth line vector from H_m. The hadamard basis element $H(m,n)$ is generated by:

$$H(m,n) = \mathbf{h}_m \cdot \mathbf{h}^n \tag{E.11}$$

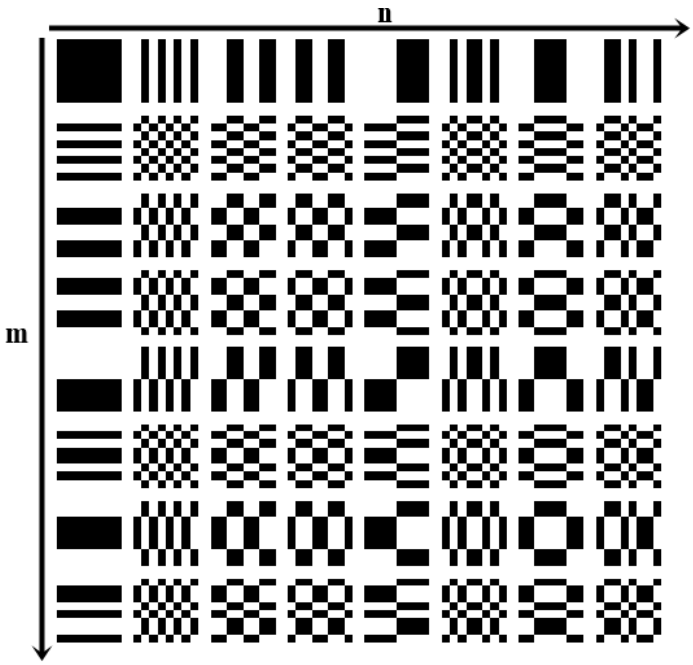

FIGURE E.3: 64 Hadamard matrices. The set stems from an unordered 8x8 hadamard basis, with each mode generated according to equation E.11

It's important to note that the resolution of spatial light modulators usually extend far beyond the typical basis sizes used. As an example, the screen size of the DMD described in chapter 4 and a basis sizes above 64x64 elements are typically not used. For this reason, it is natural to expand a basis element to bigger matrices, so as to achieve a better wavefront modulation on the SLM. To do so, we recur to the Kronecker product as:

$$H(m,n)^{\text{expanded}} = H_m \otimes \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{E.12}$$

The set of pixels that represent a single entry $(H(m,n))_{ij}$ we'll call "macro-pixel". Finally, we remark that this basis set, due to its binary nature fits quite well both for phase-only modulation, by mapping -1 to 0 rad and +1 to π rad.

The TM can have complex values, which makes it a difficult task to measure, particularly in optics, since we only have access to time averaged intensity patterns through standard photodetectors. We thus need to use interferometric methods. Some examples include off-axis digital holography [92] which is capable of giving remarkably accurate

results, although it relies on a Mach-Zehner set-up, which involves a more complex set-up. A solution to this would be to perform self-reference interference by dividing the SLM screen into several parts and making use of diffraction gratings phase masks for beam steering [111]. If interferometric methods aren't an option, it is still possible to infer the phase of a wavefront through phase retrieval algorithms [112–114] that rely either on multiple or single shot intensity measurements. In this work, we'll use a slightly more subtle technique as was done in Ref.[100], yet, it can still be classified as a self-referenced interferometric method. First, let us define I_m^α as:

$$\begin{aligned} I_m^\alpha = |E_m^{out}|^2 &= \left| s_m + \sum_{n=0}^{N-1} e^{i\alpha} k_{mn} E_n^{in} \right|^2 \\ &= |s_m|^2 + \left| \sum_{n=0}^{N-1} e^{i\alpha} k_{mn} E_n^{in} \right|^2 + 2\mathcal{R}\left(e^{i\alpha} s_m^* \sum_{n=0}^{N-1} k_{mn} E_n^{in} \right) \end{aligned} \tag{E.13}$$

Where s_m is a reference wavefront, and the m subscript denotes the mth output node, in our case, the mth pixel on the digital camera. If we now consider a single input mode, we can show after some algebra that:

$$\frac{I_m^0 - I_m^\pi}{4} + i\frac{I_m^{3\pi/2} - I_m^{\pi/2}}{4} = s_m^* k_{mn} \tag{E.14}$$

Thus, we see that if we perform 4 intensity measurements with different phase shifts on the encoded light we're able to recover the complex nature of the TM. Also note that the observed transmission matrix is not the real one:

$$K_{obs} = KS_{ref} \tag{E.15}$$

where S_{ref} is a diagonal matrix of elements, where $(S_{ref})mm = s_m^*$ represents the static reference wavefront in amplitude and in phase. In order to achieve this, we will use the background of the SLM display as the reference, as illustrated in figure E.4. Why does this work? After all, following the SLM, the wavefront is $E_{out} = E_{in}e^{i\phi(x,y)}$, where $\phi(x,y)$ is a function representing the phase modulation from the SLM, which is certainly not of the form $E_{out} = s_m + E_{in}e^{i\phi(x,y)}$. Even though this is true, by allowing some unmodulated light into the diffuser, because it scatters in many directions, then the light that does in fact reach the detector screen will have some constant contribution that is never modulated and always interferes with the portion of the light that actually gets modulated, hence we recover the self-reference interferometry.

To verify this theory, we've performed the measurement of a transmission matrix of

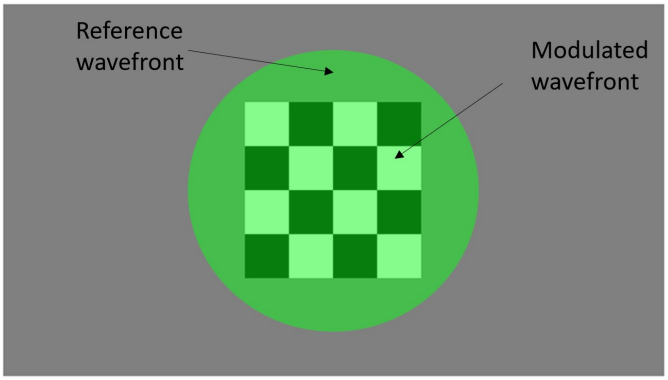

FIGURE E.4: Illustration of the wavefront modulation for acquisition of the complex transmission matrix K_{obs}. The green circle represents the light that is captured by the first objective and focused on the diffuser.

a multi mode fibre. We've used a 16x16 hadamard basis, as outlined above, and we've recovered images of 300x320 pixels on the camera. In figure E.5 it is shown a measurement of an element of the transmission matrix. It's important to note that the values of the electric field* are rather small, as they barely stand out from random shot noise. This is indication of poor phase modulation of the output which can be explained by poor coupling of the input wave to the multi mode fibre.

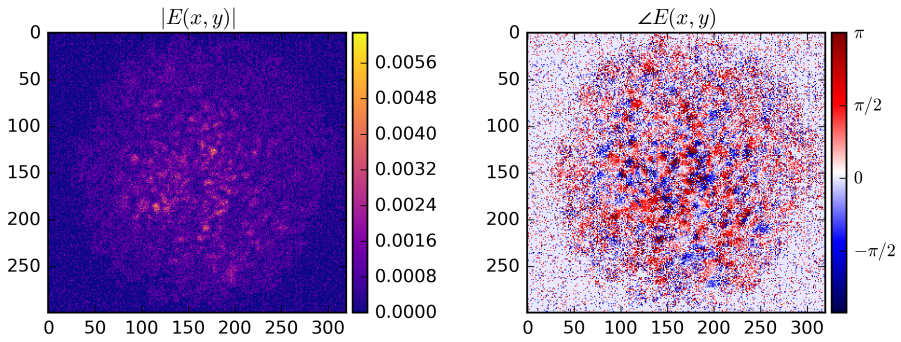

FIGURE E.5: Experimental measurement of the output of a single input basis element, according to equation E.14.

After getting the TM, we can perform some tests to verify our experiments. One way to do it is through the statistical properties of the transmission matrix. It is known that a transmission matrix of shape N by M of a system dominated by multiple scattering amounts to a random matrix of independent identically distributed entries of Gaussian statistics [100]. Thus, it is possible to resort to random matrix theory (RMT) and study the distribution of the singular values, which is predicted to follow the Marcenko-Pastur law

*For simplicity we'll refer to the element of the transmission matrix as "electric field", though it must be stressed that this method does not measure the complex electric field, but rather an output field influenced by the reference wave as per equation E.14.

[115]. For $\gamma = M/N$:

$$\rho(\tilde{\lambda}) = \frac{\gamma}{2\pi\tilde{\lambda}}\sqrt{(\tilde{\lambda}^2 - \tilde{\lambda}^2_{min})(\tilde{\lambda}^2_{max} - \tilde{\lambda}^2)}, \forall\tilde{\lambda} \in [\tilde{\lambda}_{min}, \tilde{\lambda}_{max}] \tag{E.16}$$

where $\tilde{\lambda} = \frac{\lambda}{\sqrt{1/N\sum_i \lambda_i^2}}$, $\tilde{\lambda}_{min} = (1 - \sqrt{1/\gamma})$ and $\tilde{\lambda}_{max} = (1 + \sqrt{1/\gamma})$. For the case of $\gamma = 1$ we recover the well-known "quarter-circle law":

$$\rho(\tilde{\lambda}) = \frac{1}{\pi}\sqrt{4 - \tilde{\lambda}^2} \tag{E.17}$$

However, equations E.16 and E.17 are valid for a TM without any correlations between its elements. This is generally not the case experimentally, since correlations can appear from many places such as digital camera pixel crosstalk, non-orthogonality of input basis set, presence of ballistic photons and the influence of the reference field. In any case, we can get remarkably close to the Marcenko-Pastur law in two filtering steps:

1. As detailed in Ref.[100] we can define:

$$k^{filt}_{mn} = \frac{k^{obs}_{mn}}{\sqrt{\langle|k^{obs}_{mn}|^2\rangle_n - \langle|k^{obs}_{mn}|\rangle^2_n}} \tag{E.18}$$

   Where $\langle\cdot\rangle_n$ denotes the average over all the input modes. Under certain assumptions, it can be shown that the distribution of SVD's of K^{filt} is the same as that of K alone. This step removes the effect of the reference;

2. In order to remove inter-element correlations, we can take a sub-matrix of the TM by choosing only one element out of two.

In our case, the images recovered were 300x320 pixels, thus we were not able to reach $\gamma = 1$, however, we've come fairly close with a 20x downsampling, achieved via local averaging, which resulted in $\gamma = 1.06(6)$. The singular value spectrum of the downsampled transmission matrix as well as its filtered version as just described, are shown in figure E.6. As can be seen, our results deviate quite significantly from the expected curve from RMT. Nonetheless, it's interesting to note that since our set-up was likely poorly aligned, as evidenced by the low values of the electric field from figure E.5, the singular value spectrum is likely to deviate from what random matrix theory predicts. In spite of this disparity, we follow our study on the transmission matrix

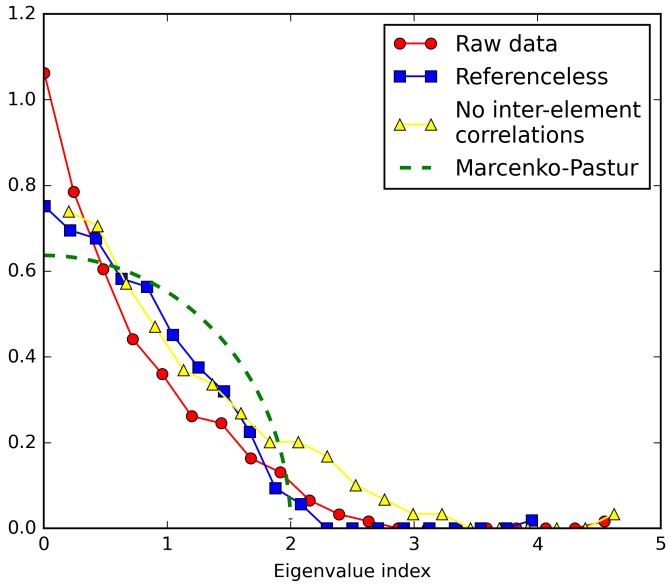

FIGURE E.6: Probability distribution of the singular values from an experimental TM. In red we have the results from a singular value decomposition on raw data; in blue, we repeat the analysis for a filtered TM as per equation E.18; in yellow, we have removed the inter-element correlations; and finally in green we have the expected tendency given by Marcenko-Pastur law, in equation E.16.

Time-reversal

In optics, time-reversal is achieved by phase-conjugation, and before the advances in spatial light modulation it was done analogically through the use of third order non-linear crystals via a four-wave mixing process. It is possible to show that if there is an optical element that can generate a backward-going wave whose amplitude is the complex conjugate of that of the forward-going wave at any one plane, then the field amplitude of the backward-going wave will be the complex conjugate of that of the forward-going wave at all points in front of the mirror. In particular, if the forward-going wave is a plane wave before entering some aberrating medium, then the backward-going (i.e., conjugate) wave emerging from the aberrating medium will also be a plane wave [75].

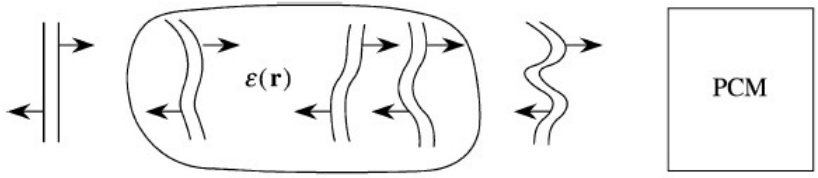

FIGURE E.7: Conjugate waves propagating through an inhomogeneous optical medium. Image taken from Ref.[116].

Having this in mind, we can define a target electric field E_{out}^{target}, defined in terms of the detector pixel basis*, and write:

$$E_{in} = K^{\dagger} E_{out}^{\text{target}} \tag{E.19}$$

Following this approach, we can try to focus light onto a single pixel. The results are shown in figure E.8.

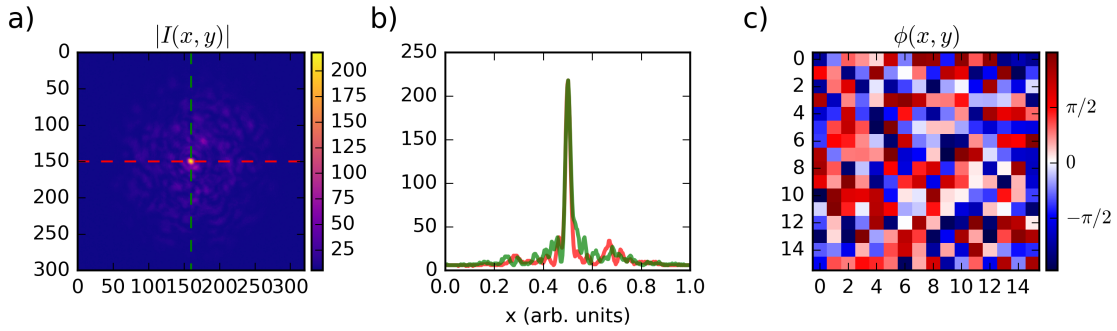

FIGURE E.8: Experimental results for single spot focusing through a multi mode fibre. a) Measured intensity patter; b) Intensity cross sections as per a), and c) phase mask applied to the input field.

Finally, we outline a method which can, in principle, give a better performance when controlling light through complex media. he previous approach, while effective, has a clear performance problem, which can be readily seen:

$$\begin{aligned} E_{out} &= K E_{in} \\ &= K K^{\dagger} E_{out}^{\text{target}} \end{aligned} \tag{E.20}$$

Generally, $KK^{\dagger} \neq I$, thus distorting the output field. A more robust and ideal approach would be to treat this as an inverse problem, thus defining:

$$E_{in} = K^{-1} E_{out}^{\text{target}} \tag{E.21}$$

Where the inverse operator here also extends to the Penrose pseudo-inverse, to include the case when $M \neq N$. However, in the presence of noise the inverse matrix can become very unstable. Thus, an intermediate approach to this would be to include the Tikhonov regularisation and define:

$$E_{in} = [K_{obs}^{\dagger} K_{obs} + \alpha I]^{-1} E_{out}^{\text{target}} \tag{E.22}$$

*Here we explicitly say the pixel basis to highlight that if we re-sampled our TM to some other basis, for example the Bessel mode basis, then the target field would have to written in terms of Bessel modes.

This operation stabilises inversion through the addition of a constraint depending on the noise level. This method has not been explored in this work, and is left as future work.

Appendix F

Wavefront optimisation algorithms

Throughout our work we've had the chance to study different wavefront shaping techniques. One of the most popular methods, the transmission matrix method, is described in appendix E and can be used to control light through any linear material, and is of particular interest when considering strongly scattering media. Nonetheless, historically, there were other methods that came before the transmission matrix, which treat the problem as an optimisation task, and are capable of finding an optimal phase mask through iterative processes. In this appendix, we'll look over three different wavefront optimisation algorithms, of which we'll present experimental results of two of them, and perform a comparative analysis between those and the transmission matrix approach. In order to benchmark the methods we'll try to focus light onto a single spot at the centre of the digital camera. As for the experimental set-up, we will use the one described in figure 6.2 of the main text.

CSA - Continuous Sequential Algorithm

In 2007 Vellekoop and Mosk [117] introduced an iterative algorithm that allowed to find the optimal wavefront which would focus light through a rutile (TiO_2) pigment sample, onto a target area of the size of a single speckle. For this algorithm, they determine the optimal phase for a single macro-pixel at a time by cycling its phase from 0 to 2π, and store the optimal phase. Then, due to the linear dynamics, the final phase mask can be found by joining all the optimal phase values for each macro-pixel. A year later the same researchers have introduced a summary and comparative analysis of 3 different, yet similar, algorithms: i)Stepwise sequential algorithm (SSA); ii)Continuous sequential

algorithm (CSA) and iii) Partitioning algorithm (PA) [118]. Based on their reported results, we've decided to explore only the CSA since it exhibited the most consistent and robust convergence performance in noisy and noiseless environments. The algorithm is depicted in figure F.1. We start with an initial phase mask, which can either be random or zero everywhere, we select a macro-pixel and cycle its value from 0 to 2π and store all the speckle patterns generated by them. With these patterns we calculate the value of a function $f(\phi)$ which we aim to maximise and select the optimum phase value, ϕ_{best}, and replace the macropixel's value with ϕ_{best}, and then repeat the process for every element of the phase mask. In our case, the target function $f(\phi)$ is defined as $f(\phi) = \sum_{i=1}^{N} I_i(\phi)T_i$, where $I(\phi)$ is the speckle pattern corresponding to the phase value ϕ and T is the target intensity pattern. In our case, T is merely a circle centred in the image with the size of a single speckle, where it takes the value of 1 inside this circle and 0 everywhere else. In a real experiment there will be various sources of noise, thus the behaviour of $f(\phi)$ won't be smooth, but rather erratic. There are various approaches to circumvent this problem, but we've found that a simple running mean filter can be quite effective.

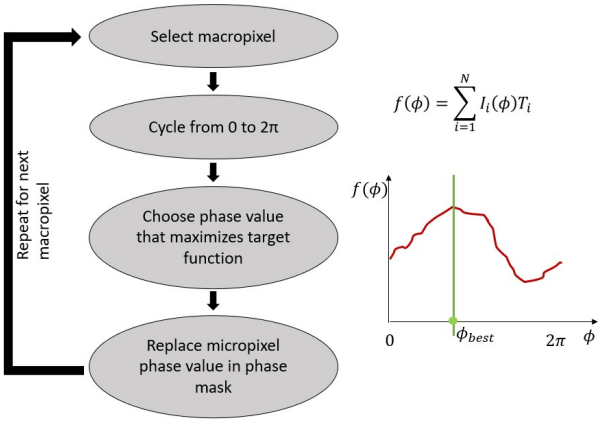

FIGURE F.1: Continuous sequential algorithm diagram.

The difference with respect to the SSA is that the last step would not take place, that is, the optimum phase value for each macropixel would be found without any information of the previous optimised macropixels. It's important to note that, while the CSA may converge faster in terms of iterations, the SSA could enable a more parallel computation due to the DMD large memory, thus compensating the Lee hologram computation time overhead. In the PA, the key difference relative to the CSA is that instead of choosing only one macropixel, we would choose many at random, and then you'd repeat the process until convergence.

COAT - Coherent Optical Adaptive Technique

In 2011 Cui [119] introduced a wavefront optimisation method towards focusing through highly scattering media, which can be highly parallelisable. It is inspired by the multidither coherent optical adaptive technique (COAT) [120], a method developed at the Hughes Research Laboratories in the 1970s for focusing a laser beam through air turbulence. To understand the algorithm, let us first look at the case of free space propagation as in figure F.2. In this scenario we have two beams, a reference beam and a phase modulated beam, and we let them interfere at a screen where we will see the typical fringe pattern where the beams overlap. If we now control the intensity of a spot much smaller than the fringe spacing, $I_\omega(t)$, and allow for the phase $\phi(t) = \omega t$, we can write:

$$I_\omega(t) = A + B\cos(\omega t + \phi_0) \tag{F.1}$$

where ϕ_0 is the phase difference between the reference and modulation beams, and A and B are real constants. Suppose now that we wanted that, at the monitored spot, the intensity was minimum. In that case, one should have $I_\omega = A - B$, which can only be accomplished if $\phi(t) + \phi_0 = \pi$. Thus we see that the goal is to use the phase modulation to compensate for ϕ_0. Looking at the fourier transform of equation F.1 we have:

$$\mathcal{F}\{I_\omega(t)\} = A_f\delta(f) + B_f\delta(f-\omega)e^{-i\phi_0} + B_f\delta(f+\omega)e^{i\phi_0} \tag{F.2}$$

where A_f and B_f are real constants. If we isolate the $+\omega$ component of equation F.2, $I_{+\omega}(f) = B_f\delta(f-\omega)e^{-i\phi_0}$, it's straightforward to see that $\phi_0 = \arctan\left(\frac{\mathcal{I}\{I_{+\omega}(f)\}}{\mathcal{R}\{I_{+\omega}(f)\}}\right)$. Looking back to the problem of static strongly scattering media, the output pattern would not be a clean fringe pattern, but rather a speckle pattern, however, each spot at the target screen is the nothing more than the result of a complex interference pattern, thus a modulation as $\phi(t) = \omega t$ on the modulation beam would still result in a sinusoidal intensity variation when monitoring a spot much smaller than the average speckle size, thus validating the technique. Furthermore, due to the linearity of the light propagation, we can analyse various frequency components on different input channels, which will allow for a completely parallel computation of the optimal wavefront towards focusing.

To apply this technique to strongly scattering media, we should look into figure F.3. The input field is modulated with a phase mask where each macropixel is assigned to a specific modulation frequency. From an experimental point of view, one should choose

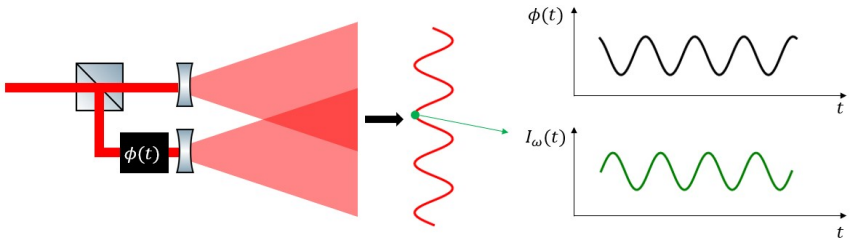

FIGURE F.2: Coherent optical adaptive technique diagram in free space.

these frequencies so as to match to the ones that the FFT (Fast Fourier Transform) algorithm uses to sample the DFT (Discrete Fourier Transform). Then, we retrieve the intensity at a specific spot as a function of time (note that this time, does not need to be measured in seconds, but rather just some arbitrary unit), and then we perform a fourier transform of $I_\omega(t)$ and finally retrieve the phase of each of the modulation components (in red in the figure). After replacement of these phase values in the phase mask in the respective channels, the output should be maximised for total intensity at the monitored pixel.

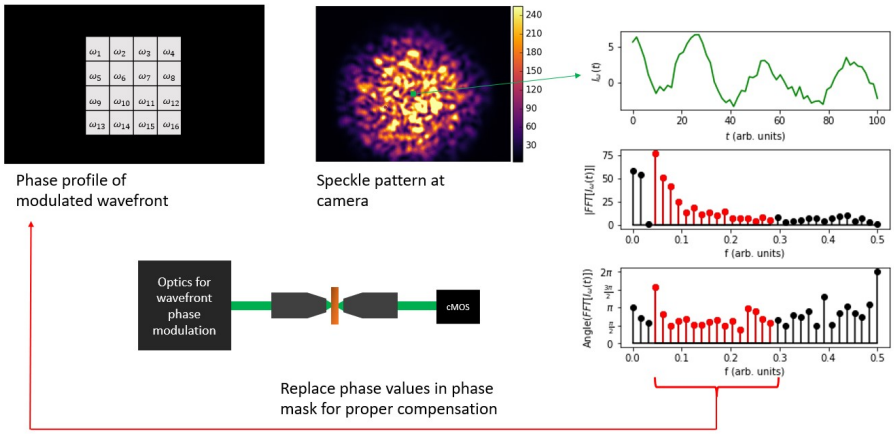

FIGURE F.3: Coherent optical adaptive technique diagram in strongly scattering media.

This method has not been properly explored experimentally in our work, and we haven't made further progress, as our interest laid with iterative methods. Nonetheless, the elegance of the method has been captivating and we decided to include it in this thesis as both from a pedagogical point of view, as well as a suggestion for future work to be carried out in wavefront optimisation techniques.

GA - Genetic Algorithm

In 2012 Conkey et al. [121] introduced a genetic algorithm optimisation for focusing through turbid media, and was able to demonstrate a remarkable performance compared with existing methods. Genetic algorithms are based on the principle of evolution through natural selection, coined by Charles Darwin in 1859 in his magnum opus, The Origin of Species [122]. This principle has been generally accepted among scientists, and some general remarks are of particular relevance:

1. Each individual tends to pass on its traits to its offspring;
2. Nevertheless, nature produces individuals with differing traits;
3. The fittest individuals - those with the most favourable traits - tend to have more offspring than do those with unfavourable traits, thus driving the population as a whole toward favourable traits;
4. Over long periods, the variation can accumulate, producing entirely new species whose traits make them especially suited to particular ecological niches.*

When designing a computational algorithm based on this evolutionary principle, one must identify the key operations that take place in the algorithm, namely the crossover, mutation and generational survival. The original work has been devised for binary amplitude masks, but here we will formulate the algorithm for continuous phase masks, however minimal alterations have to be made go back to binary mask optimisation. Let us go step-by-step through the algorithm:

Initial population

For this algorithm we need an initial population, which in our case is a set of phase masks. The most usual way is to generate this set at random, that is, where each macropixel is sampled from a probability distribution, in our case we chose uniform from 0 to 2π. However, one can argue that there could be better initial guesses that could potentially lead to faster convergence and better avoid local minima.

*This enumeration has been transcribed from Ref.[122].

Fitness rank

A crucial part in any GA is the fitness rank, which is used to give a quantitative value to the question "How good is this individual to our specific goal?". It's important to note that the fitness function will dictate the solution landscape and consequently greatly affect the influence of mutation and crossover operations. In our case, we are interested in focusing light in specific regions of space, thus, let T be the target intensity pattern where it takes the value 1 where we want focused light and 0 everywhere else, Y the pattern recorded on camera, and A be the phase mask that gave rise to Y. We propose two methods:

1. **Integral**: Here we integrate all the light in the target regions and normalise to the maximum value it can take, which in the case of an 8 bit resolution is:

$$f(A) = \frac{\sum_{i=1}^{N} Y_i T_i}{\sum_{i=1}^{N} 255 \times T_i} \tag{F.3}$$

2. **SNR (Signal to Noise Ratio)**: The goal of this fitness function is to prioritise the contrast rather than intensity only. Objectively they should converge to the same solution, but the solution landscape will be different.

$$f(A) = \frac{\sum_{i=1}^{N} Y_i T_i}{\sum_{i=1}^{N} Y_i (1 - T_i)} \tag{F.4}$$

As we'll see next, it's useful to define the ranks with the criteria "the higher the better".

Choosing crossover individuals

As we've seen, in nature the fittest individuals tend to have more offspring than those with unfavourable traits. In order to replicate this behaviour in our algorithm, we've followed three approaches: i) fitness-aware sampling, ii) fitness-unaware sampling and iii) fitness and diversity-aware sampling. In the first, we have sampled Q individuals from the population with respective probabilities:

$$P_i = \frac{f_i}{\sum_i^M f_i} \tag{F.5}$$

After choosing the first Q elements, we remove them from the population to avoid crossover of the elements with themselves, and repeat the process with the remaining individuals. The second approach does not take in consideration the value of the fitness of an individual, but rather it's relative ordered rank. If we let the P_1 be the probability

of choosing the most fit individual and P_M be the probability of choosing the least fit individual, we can write:

$$\begin{aligned} P_1 &= P_c \\ P_2 &= P_c(1-P_c) \\ &\vdots \\ P_{M-1} &= P_c(1-P_c)^{M-2} \\ P_M &= 1 - \sum_{i=1}^{M-1} P_i \end{aligned} \tag{F.6}$$

Where P_c is a real number between 0 and 1. Finally, one can also include a form of diversity ranking which will allow to choose crossover pairs that will generate an offspring that will have the best balance between fitness and diversity thus allowing to keep the population well scattered across the solution space and not stagnate in a local minimum/-maximum. In order to do this, we first choose the first Q elements based on fitness. Then, we rank the remaining $M - Q$ elements based on how different they are relative to the first Q individuals chosen as:

$$d_i = \frac{1}{N} \sum_{j=1}^{Q} \sum_{k=1}^{N} |(A_j)_k - (A_i)_k|^2 \tag{F.7}$$

Then, we join the fitness and diversity rankings in a single metric:

$$l_i = \lambda f_i + (1-\lambda) d_i \tag{F.8}$$

with λ being a real number between 0 and 1. With this new ranking, we now choose the remaining Q elements in the same manner as we did for the first Q, replacing only the respective ranking.

Crossover operation

The crossover operation is what determines how the traits of two individuals can be combined to generate an offspring that will inherit qualities from both parents. In our case we've decided to study two approaches: i) binary selection and ii) linear combination. In the first approach, we generate a phase mask, S, such that each element is either 0 or 1

chosen at random. Given two parents Q_{ma} and Q_{pa} we can generate two offsprings as:

$$(O_i(Q_{ma}, Q_{pa}))_k = S_k(Q_{ma})_k + (1 - S_k)(Q_{pa})_k \tag{F.9}$$

$$(O_i'(Q_{ma}, Q_{pa}))_k = (1 - S_k)(Q_{ma})_k + S_k(Q_{pa})_k \tag{F.10}$$

The second approach is only possible when dealing with a continuous solution space, which is our case as we're dealing with phase masks. The offspring is generated as:

$$(O_i(Q_{ma}, Q_{pa}))_k = \frac{1}{2}\left((Q_{ma})_k + (Q_{pa})_k\right) \tag{F.11}$$

Mutation operation

Mutations are inherently random, thus the first step in this process is to randomly select a set of macropixels in the phase mask and then modify them. In our work, we've followed the approach taken by Conkey et al. [121] when choosing the number of macropixels to modify in each generation as:

$$R(n) = \text{int}\left[\left((R_0 - R_{end})e^{-n/\lambda} + R_{end}\right) * N\right] \tag{F.12}$$

Where R_0 and R_{end} define the initial and final mutation rate, and λ defines the mutation decay rate. This ensures that initially we have a high mutation rate, allowing for the population to search the solution space more aggressively, and for later generations we have a low mutation rate thus allowing to converge to an optimal solution.

Having selected the macropixels to alter, we need to define how they should be modified. We propose 4 methods:

1. **Incremental**: Each of the selected macropixels are modified as $A_k \to A_k + d\phi$, where $d\phi$ is a fixed real value;

2. **Random**: Each of the selected macropixels is set to a random number between 0 and 2π;

3. **Phase Flip**: Each of the selected macropixels is modified as $A_k \to \pi - A_k$;

4. **Random incremental**: Each of the selected macropixels is modified as $A_k \to A_k \pm d\phi$, where $d\phi$ is a fixed real value, but the $+$ or $-$ sign is randomly selected.

Generational survival

In this final step we choose how many individuals from the current population survive to the next generation. In our case we've decided to implement this step as follows: from the mutated individuals and offspring, we choose the V most fit elements, and replace the worst V individuals in the current population with these fitter elements.

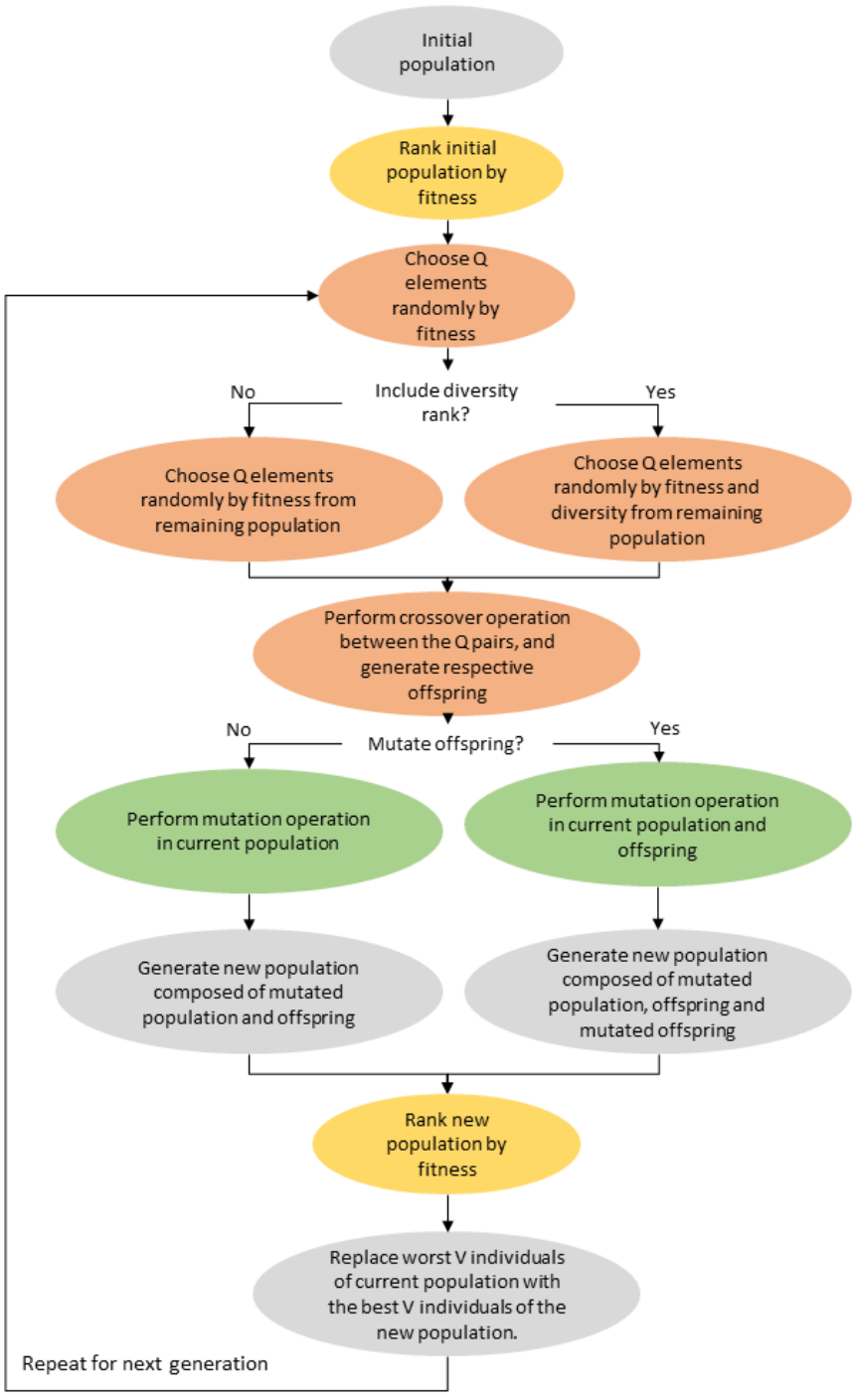

FIGURE F.4: General description of the genetic algorithm employed.

Focusing through a multimode fibre

In this section we're going to go over the performance of different algorithms, namely TM, CSA and GA towards the common goal of single spot focusing through a multimode fibre. In order to have a quantitative comparison of the focus spot, we have chosen to do it with respect to the average speckle size of the fibre's output. To calculate this, we use the full width at half maximum of the 2D autocorrelation peak of the intensity pattern [62, 123]. The 2D autocorrelation of a function $f(x,y)$ is defined as:

$$\Gamma_I(x,y) = \int\int_{-\infty}^{+\infty} f(\xi,\eta) f^*(\xi - x, \eta - y) d\xi d\eta \tag{F.13}$$

We can recognize equation F.13 as a linear convolution and thus make good use of the fourier domain:

$$\mathcal{F}\left\{\Gamma_I(x,y)\right\} = |F(\mu,\nu)|^2 \tag{F.14}$$

where $F(\mu,\nu)$ is the fourier transform of $f(x,y)$. The results are shown in figure F.5 and the calculated average speckle sizes in the x and y direction are:

$$\Delta x = (11.7 \pm 0.3)\text{pixels} \tag{F.15}$$

$$\Delta y = (12.6 \pm 0.3)\text{pixels} \tag{F.16}$$

Recognising that the camera was capturing only every two pixels and considering the $8\mu m$ pixel pitch, these measures translate directly to:

$$\Delta x = (112 \pm 3)\mu m \tag{F.17}$$

$$\Delta y = (121 \pm 3)\mu m \tag{F.18}$$

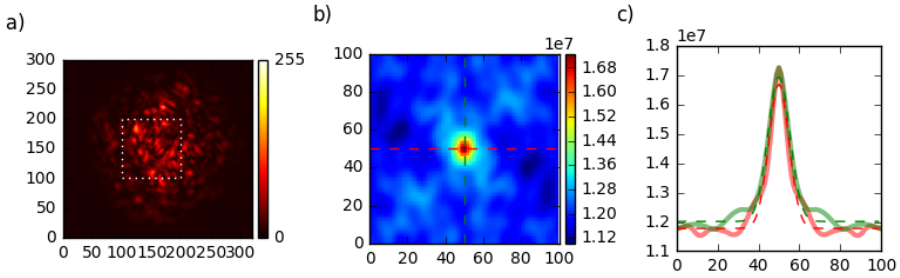

FIGURE F.5: Calculation of the average speckle size based on the autocorrelation peak. a) Output speckle pattern for a constant phase mask as input; b) 2D autocorrelation of the highlighted area in a), calculated as in equation F.14; c) Vertical and horizontal (green and red lines) cross sections of the autocorrelation function. The solid semitransparent lines are the cross sections, and the dashed lines represent a nonlinear fit to a gaussian curve.

With this in mind we choose as target mask a centered 5 pixel radius circle. Furthermore, as we'll see all the focal spots have similar shape, and after a careful analysis of one case obtained with 1024 input modes with the genetic algorithm, we see that the obtained spot has dimensions of the order of the average speckle size as calculated previously.

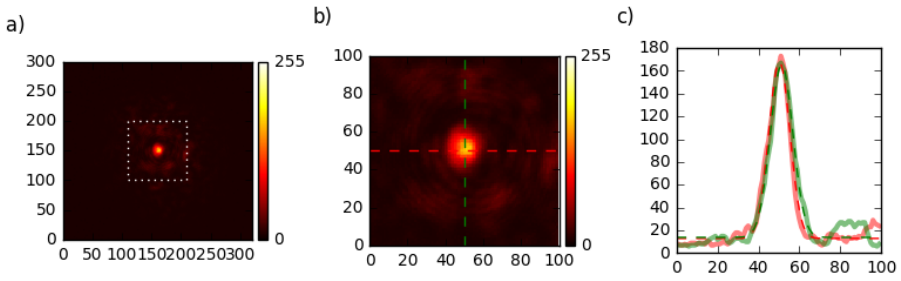

FIGURE F.6: Calculation of the spot size. a) Output speckle pattern for an optimised phase mask; b) close-up image of the highlighted area in a); c) Vertical and horizontal (green and red lines) cross sections of the autocorrelation function. The solid semitransparent lines are the cross sections, and the dashed lines represent a nonlinear fit to a gaussian curve.

Upon a nonlinear fit to a gaussian function the spot dimensions are:

$$\Delta x = (12.0 \pm 0.2)\text{pixels} \tag{F.19}$$

$$\Delta y = (12.9 \pm 0.3)\text{pixels} \tag{F.20}$$

Experimental rules of thumb for the genetic algorithm

Before diving in to a comparative analysis of the different methods, we have performed several performance tests with the genetic algorithms to gain some insight on the influence of the several parameters. In the following we will not present all of our studies. Instead, we will show a typical study, and we will then present a summary of our conclusions on the application of a GA to focusing through a multimode fibre.

Case study - Population size dependence

In figure F.7 we can see the results for 3 different study cases with population sizes, M, of 10, 30 and 100 individuals. The algorithm parameters are outlined in table F.1.

As seen from the figure, the results are consistent with what we'd expect since a larger population means a larger span across the solution space, thus the probability of an initial individual to land near an optimal solution is higher and can then drive the population

Initial population type	**Population size**	**Fitness function**	**Number of input modes**	**Number of crossover operations**
Random	M	Integral	256	$M/2$
Number of survivals	**Crossover operation**	**Initial mutation rate, R_0**	**Final mutation rate R_{end}**	**Mutation decay rate, γ**
$M/2$	Binary	0.1	0.001	200
Mutation operation	**Mutate offspring**	**Include diversity ranking**	**Diversity importance, λ**	**Prob. dist. for crossover individuals**
Random	Yes	Yes	0.5	Fitness aware

TABLE F.1: Genetic algorithms parameters for population size dependence study.

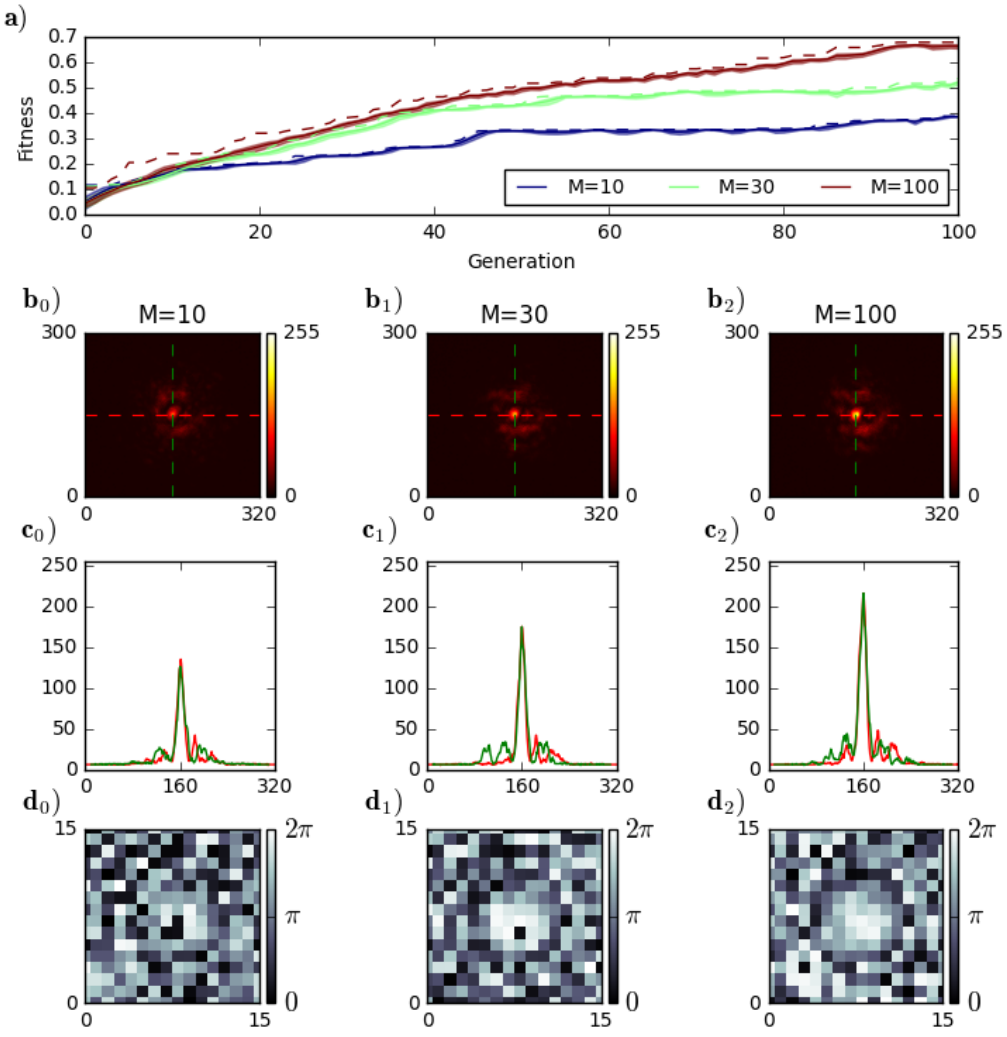

FIGURE F.7: Population size dependence of the single spot focusing performance. a) Fitness evolution throughout the generations. Solid lines represent the average fitness of the population, and the shaded region is a representation of the standard deviation of the fitness. The dashed lines represent the evolution of the fitness of the most fit individual in each generation; b0)-b2) are the output after 100 generations for each study case; c0)-c2) are the vertical (green) and horizontal (red) cross sections of the intensity outputs b0) to b2), respectively; d0)-d2) are the best phase masks after 100 generations.

towards that better solution. However, this comes at a cost of a greater computation time, since it scales linearly with the number of individuals in a population.

Summary

From all our results we can outline some rules-of-thumb for dealing with the genetic algorithm for light manipulation through a multimode fibre:

1. A big population size will lead to better results, but it can take a long time to compute;
2. What defines a what a "big" population size is is related to the dimensionality of the solution space, and consequently to the number of input modes;
3. Mutation operations should be performed either in a random or incremental manner;
4. Diversity ranking inclusion did not prove itself to be useful;
5. There is no significant advantage in letting more than 50% of the current population be replaced;
6. Electronic saturation at the camera is not helpful;
7. An integral fitness function proved itself to be rather effective for single spot focusing;
8. The mutation rate should be kept low, but not too low;

Comparative analysis

Having studied the genetic algorithm, we are now prepared to compare the different methods. We test all the methods towards single spot focusing, with variable number of input modes, namely 64, 256, 1024. The parameters for the genetic algorithms are shown in table F.2. As for the CSA we've used 30 phase samples, and the running mean filter used 5 sample window. The performance metric chosen was the integral method as used before.

The results can be seen in figure F.8 for 64 input modes. From F.8a) we have the evolution of the different algorithms, thus for each method the x axis represents something different and curve comparison should be made with care. For the GA, the x axis is the generations, while for CSA is the macropixel being optimised. The horizontal curve for

Initial population type	**Population size**	**Fitness function**	**Number of input modes**	**Number of crossover operations**
Random	30	Integral	N	15
Number of survivals	**Crossover operation**	**Initial mutation rate,** R_0	**Final mutation rate** R_{end}	**Mutation decay rate,** γ
15	Binary	0.1	0.001	200
Mutation operation	**Mutate offspring**	**Include diversity ranking**	**Diversity importance,** λ	**Prob. dist. for crossover individuals**
Random	Yes	Yes	0.5	Fitness aware

TABLE F.2: Genetic algorithms parameters for comparative analysis.

the TM represents only the performance threshold, and with no ties to the x axis. Having said this, we can see that it takes few iterations for the GA to achieve similar performance to TM method. Nonetheless, with the current parameters, it seems to have plateaued only at a slightly better fitness than the TM. On the other hand, the CSA is able to achieve much higher performance and a noticeable focusing capability. Furthermore, it's important to note that the time taken by these methods was 9.88, 73.5, 73.7 seconds (TM, CSA, GA, respectively).

We increase the number of input modes, and the results are shown in figure F.9. Do note that in this case we've increased the population size of the GA to 50 individuals. Most of the comparative conclusions from before remain, but it's important to reinforce the robustness of the CSA and GA algorithm. By looking at these results one may be tempted to question the advantage of using the GA, but when looking at the time taken by these methods - 34.4, 318, 282 seconds for the TM, CSA and GA - the computation time scaling of the CSA and TM largely outpaces the scaling of the GA since it's independent of the number of input modes, whereas the other ones are.

Finally, we repeat for 1024 input modes, and the results are shown in figure F.10. Do note that in this case we've increased the population size of the GA to 100 individuals. We can see that now the TM performs far better and it outperforms the GA. Furthermore, it can be seen that the TM allows for a much smaller spot size, however, the GA and the CSA were constrained to the target mask of 5 pixel radius. This indicates that we can choose a much smaller radius spot and perhaps choose a peak to background ratio as a

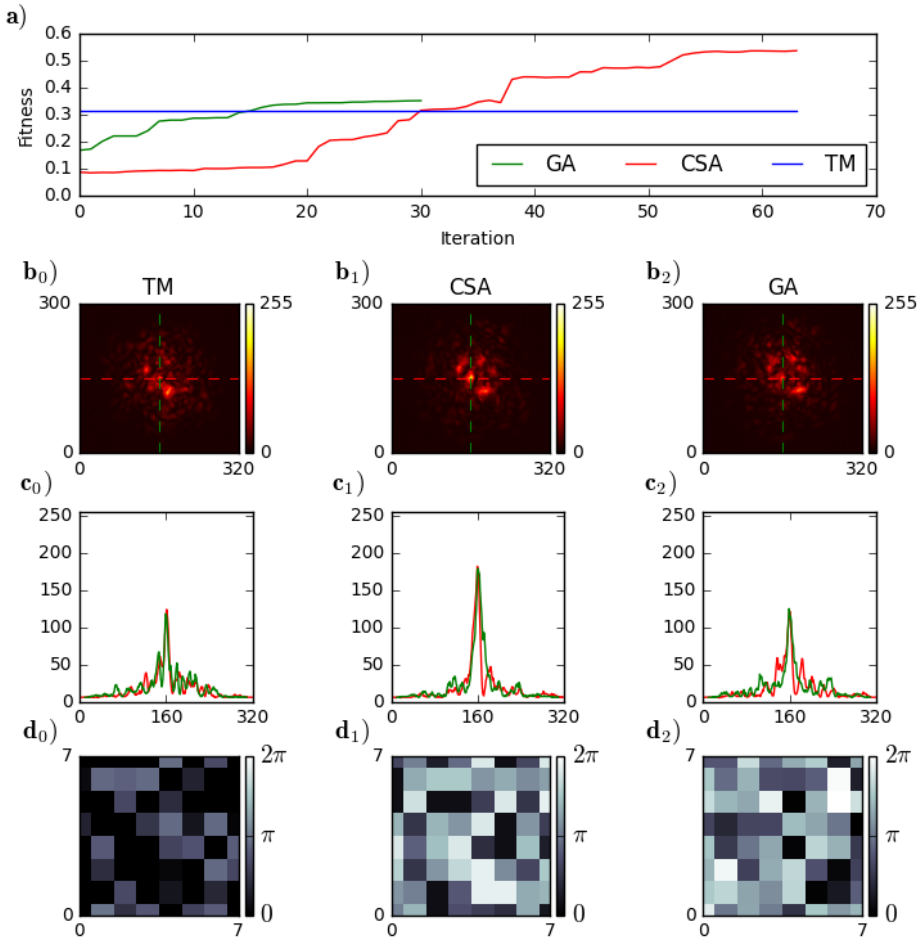

FIGURE F.8: Comparative analysis with 64 input modes. a) Fitness evolution for the different methods; b0)-b2) are the output after all the iterations for each method; c0)-c2) are the vertical (green) and horizontal (red) cross sections of the intensity outputs b0) to b2), respectively; d0)-d2) are the best phase masks obtained.

fitness function. However, it's worth pointing out that the time scaling was again quite notorious as they took 156, 1270 and 672 seconds for the TM, CSA and GA.

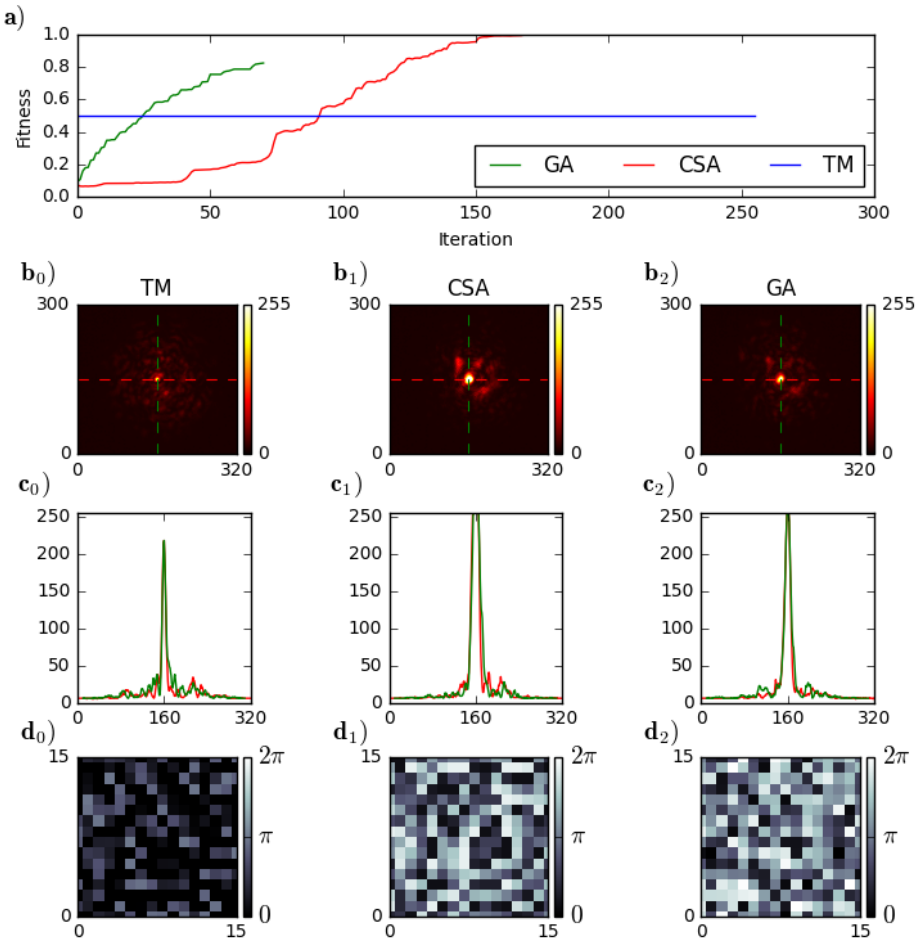

FIGURE F.9: Comparative analysis with 256 input modes. a) Fitness evolution for the different methods; b0)-b2) are the output after all the iterations for each method; c0)-c2) are the vertical (green) and horizontal (red) cross sections of the intensity outputs b0) to b2), respectively; d0)-d2) are the best phase masks obtained.

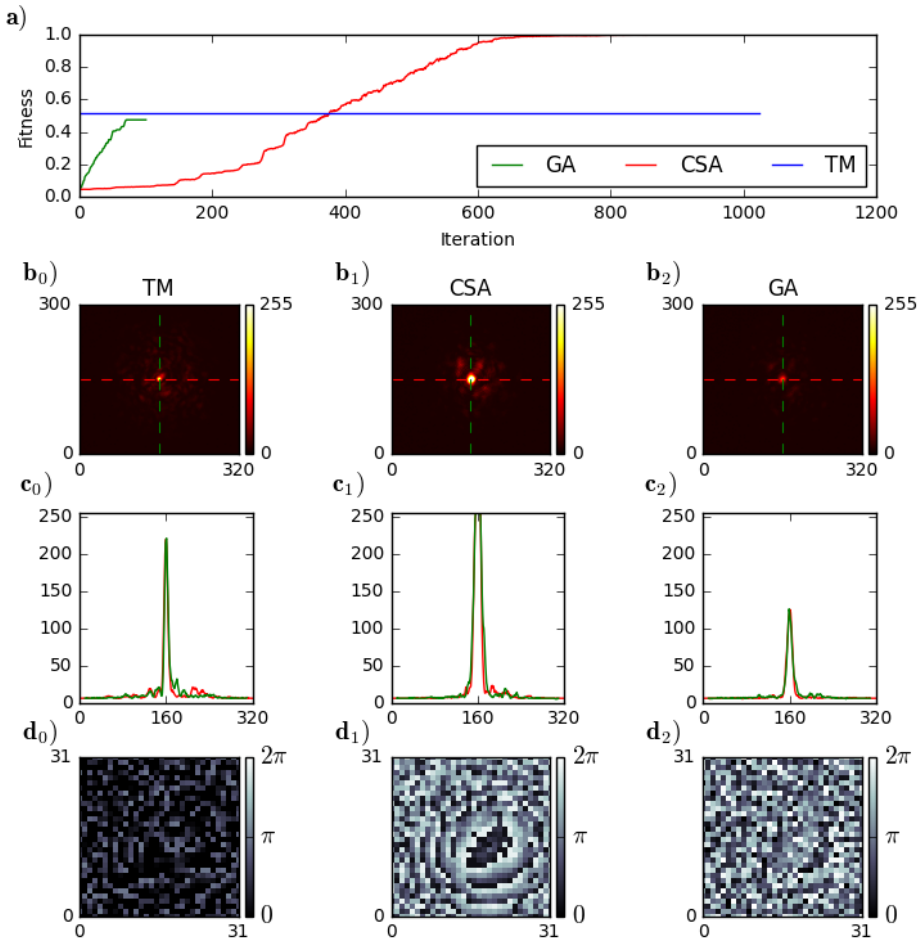

FIGURE F.10: Comparative analysis with 1024 input modes. a) Fitness evolution for the different methods; b0)-b2) are the output after all the iterations for each method; c0)-c2) are the vertical (green) and horizontal (red) cross sections of the intensity outputs b0) to b2), respectively; d0)-d2) are the best phase masks obtained.

Iterative wavefront shaping applied to light manipulation in free-space propagation

We now wish to apply the previous algorithms and knowledge acquired to the manipulation of light in free space propagation and study the limits of such manipulation. Having the final 4f system in place as in figure 6.2, we take inspiration on Cojoc and Alexandrescu [124] and place an extra convex lens within a focal distance of the imaging plane and place the camera at the fourier plane. The problem now is reduced to a typical phase retrieval problem which could, in principle, be solved via the Gerberch-Saxton algorithm. However, we've tried to do so by taking an image of the intensity profile at the 4f plane and then run the algorithm until convergence, but we were unable to reproduce an image at the fourier plane. Our suspicion is that this happened due to poor alignment of the set-up and lack of physical correspondence between DMD macro-pixels and camera image.

In order to demonstrate light manipulation capabilities, we've tested the GA on 5 different targets, both with binary phase modulation as in figure F.11 and full range phase modulation F.12.

As can be seen from the figures, we're able to converge to the intended targets thus demonstrating that phase modulation with 256 input modes alone is able to manipulate a speckle pattern.

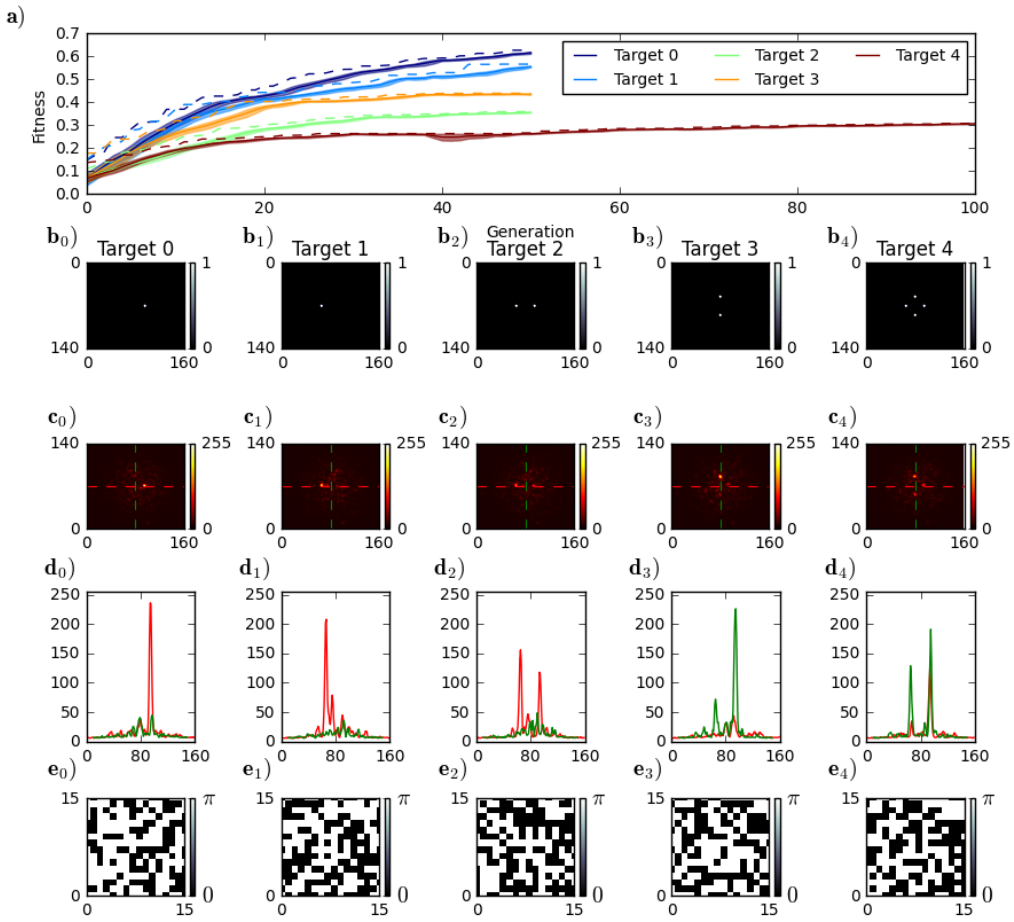

FIGURE F.11: Final results for different targets with binary phase modulation. a) Fitness evolution for the targets; b) panels are the intended target functions; c) panels are the output after all the iterations for each method; d) panels are the vertical (green) and horizontal (red) cross sections of the intensity outputs c), respectively; e) panels are the best phase masks obtained.

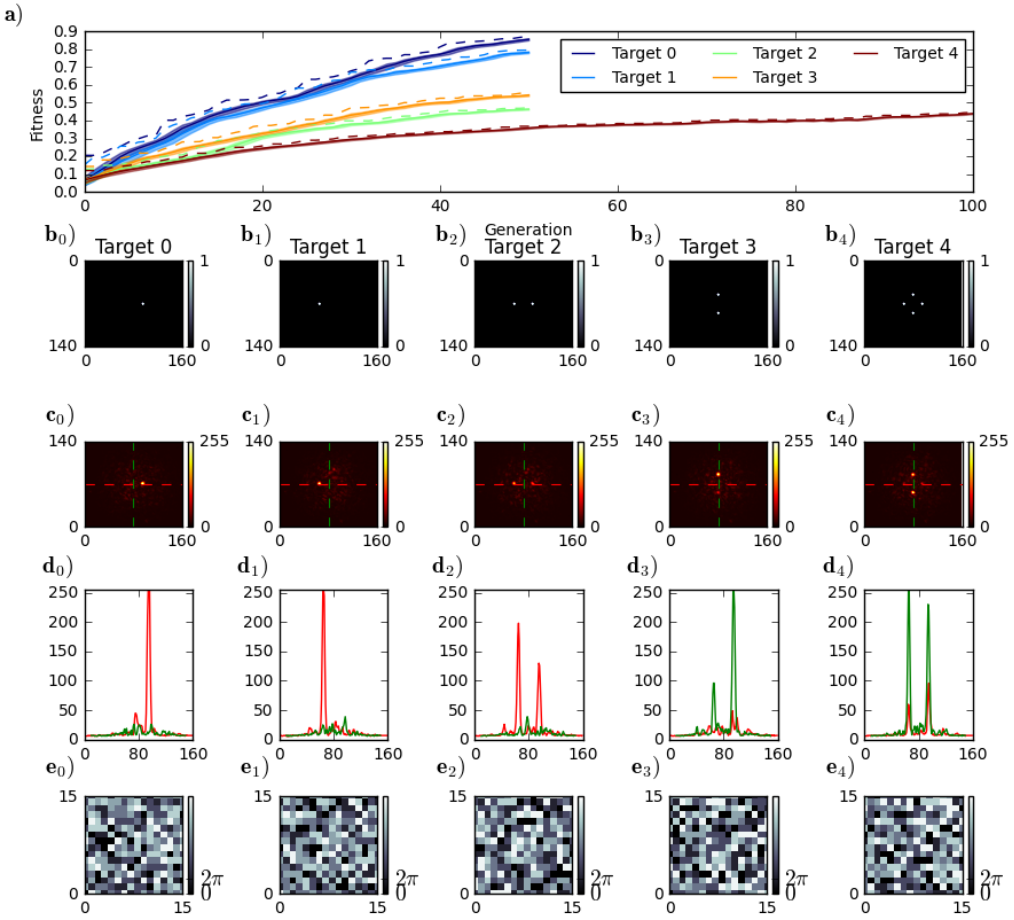

FIGURE F.12: Final results for different targets with full range phase modulation. a) Fitness evolution for the targets; b) panels are the intended target functions; c) panels are the output after all the iterations for each method; d) panels are the vertical (green) and horizontal (red) cross sections of the intensity outputs c), respectively; e) panels are the best phase masks obtained.

Appendix G

An attempt at a diffractive optical extreme learning machine

In 2018, Lin et al. [125] introduced an all-optical machine learning architecture using diffractive deep neural networks. Such set-up consists of consecutive linear layers that aim to cleverly diffract light such that the intensity at the output plane is enough to infer on the input. For training, they use a physical forward propagation model following the Rayleigh-Sommerfeld equation, in which a single neuron (i.e. a single pixel from the diffractive optical element) can be considered as the secondary source of wave $w_i^l(x,y,z)$, given by [40]:

$$w_i^l(x,y,z) = \frac{z-z_i}{r^2}\left(\frac{1}{2\pi r}+\frac{1}{j\lambda}\right)\exp\left(\frac{j2\pi r}{\lambda}\right) \tag{G.1}$$

where $r=\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2}$, j is the imaginary unit, λ is the wavelength. This allows to model the input field at the i^{th} neuron at the l^{th} layer as:

$$u_i^l(x,y,z) = w_i^l(x,y,z)t_i^l(x,y,z)\sum_k u_k^{l-1}(x_i,y_i,z_i) \tag{G.2}$$

where t_i^l denotes the complex modulation of a single pixel, i, of the l^{th} diffractive layer. Having this propagation model, it is possible to implement a backpropagation algorithm to train the network towards a specific task. An important aspect of these networks is that they only consist of linear optical elements, thus, in theory, the entirety of the network could be compressed onto a single diffractive layer, yet, they were able to demonstrate *depth* advantage of the network, as more layers resulted in a better performance. A possible reason for this can be due to small nonlinear effects at the surfaces, as well as unaccounted optical losses throughout the set-up. It's also worthwhile to note that these

networks also possess a high number of learnable parameters, thus making the training of the network quite heavy. In fact, the presented machine in [40] features 5 diffractive optical layers, resulting in 0.8 million trainable parameters. Finally, these works have been performed using THz input radiation, resulting in large optical elements (in [40] the diffractive layers were 8x8cm). However, it has recently been shown that it is possible to achieve similar performances in visible and near-infrared wavelengths [126]. In their work, each neuron has a size of approximately $4\mu m$, each layer has an area of 4x4 mm and features 1 million neurons.

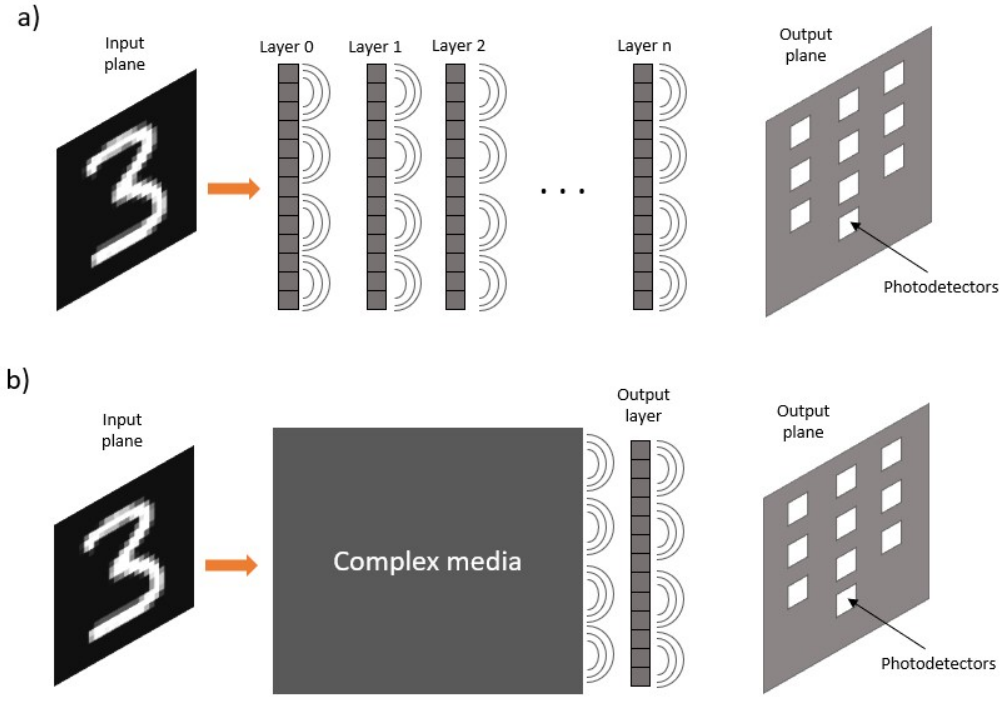

FIGURE G.1: Proposed interplay of diffractive optical neural network and extreme learning machine, exemplified for the a classification task with the MNIST dataset. a) Typical architecture of a diffractive neural network consisting of an input plane, followed by a set of trainable diffractive layers redirecting light on an output plane. b) Proposed architecture for an extreme learning machine based on diffraction. The input is fed to an optical complex media and the output is followed by a trainable diffractive layer.

To this end, we've used the set-up in figure G.2. In this set-up, the output of the MMF is sent back to the DMD where it is phase modulated through Lee holography, and the digital camera is placed on the fourier plane. It's important to note that even though we draw inspiration from diffractive neural networks, we are unable to replicate the DOE's due to the digital micromirror device's limitations. On one hand, the pixel pitch is 13.68 μm, which is about 25.7 times larger than the working wavelength of 532nm, which is far

larger than the current ratios used in the literature, having implications in the physical forward propagation model. Despite this, our previous work (light manipulation sections) shows that it is possible to manipulate light with a relatively low number of input modes with large modulation areas. On the other hand, the device only allows binary amplitude modulation, and in turn affects the phase modulation, imposing strict limitations on the resolution due to the binary quantisation, which are not yet fully documented. Thus, this device is not suitable for the backpropagation algorithm. To overcome this we've employed a genetic algorithm for training, as described in appendix F.

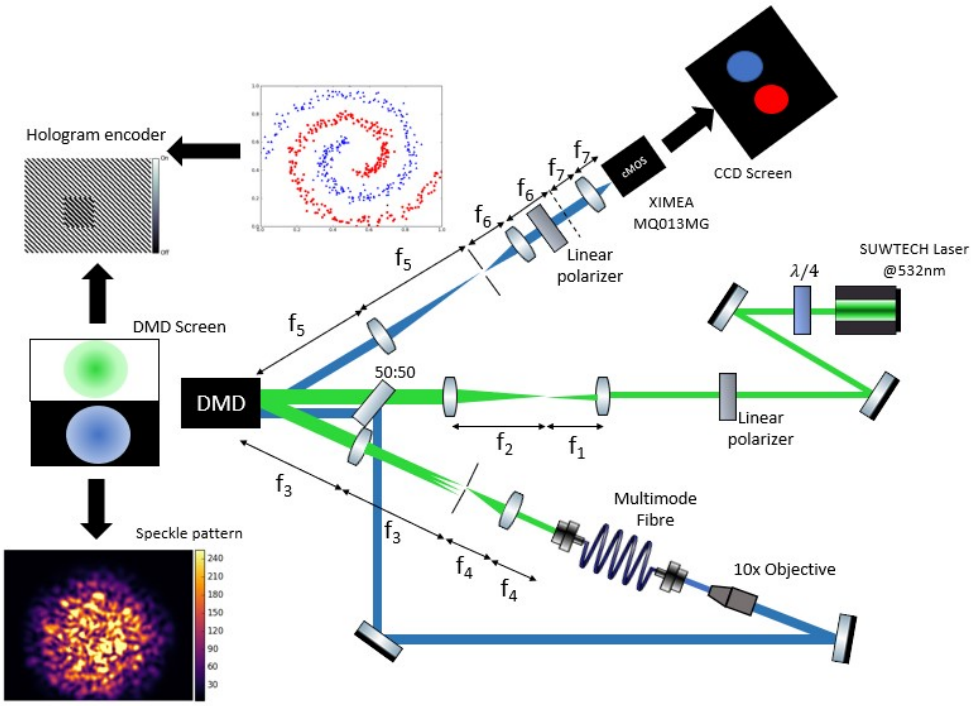

FIGURE G.2: Experimental set-up used.

We've tested our implementation in the dataset represented in figure G.3. As for the wavefront modulation schemes, we've employed those outlined in chapter 4 of the main text.

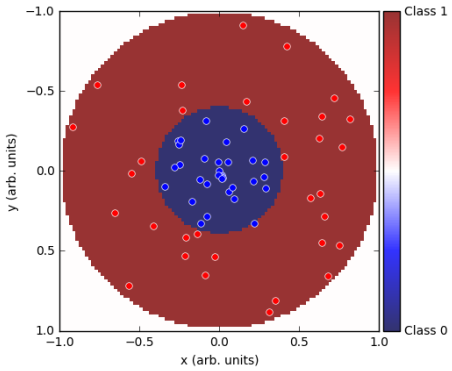

FIGURE G.3: Circular dataset

Fitness functions

The fitness function will greatly influence the convergence of the algorithm. Let T_1 and T_0 be the target masks as shown in figure G.4, and $X_k^{ij}(G_i, x_j)$ be the intensity recorded on the kth pixel on the camera with the mask G_i and input data x_j, with corresponding label $y_j \in \{0, 1\}$.

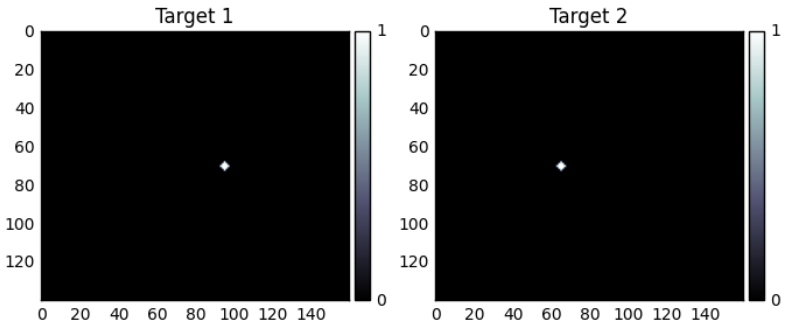

FIGURE G.4: Target masks.

Let us also define the normalized intensities I_1 and I_2*:

$$I_1(x_j) = \frac{\sum_k X_k^{ij} T_1}{\sum_k 255 \times T_1} \tag{G.3}$$

$$I_2(x_j) = \frac{\sum_k X_k^{ij} T_2}{\sum_k 255 \times T_2} \tag{G.4}$$

Let $F(G_i)$ be the fitness function for the individual G_i. We've tested the following:

*These are normalized to an 8-bit digital camera.

$$f(G_i) = \sum_j \left(\begin{cases} 1, & I_1(x_j) - I_2(x_j) > \delta I \text{ and } y_j = 0 \\ 1, & I_2(x_j) - I_1(x_j) > \delta I \text{ and } y_j = 1 \end{cases} \right) \tag{G.5}$$

where δI was added to compensate noise fluctuations.

$$f(G_i) = \sum_j^N \frac{1}{N} \left(\begin{cases} 1 + I_1(x_j) - I_2(x_j), & y_j = 0 \\ 1 + I_2(x_j) - I_1(x_j), & y_j = 1 \end{cases} \right) \tag{G.6}$$

$$f(G_i) = \sum_j^N \frac{1}{N} \left(\begin{cases} 1/I_2(x_j), & y_j = 0 \\ 1/I_1(x_j), & y_j = 1 \end{cases} \right) \tag{G.7}$$

$$f(G_i) = \sum_j^N \frac{1}{N} \left(\begin{cases} \frac{I_1(x_j)}{I_2(x_j)}, & y_j = 0 \\ \frac{I_2(x_j)}{I_1(x_j)}, & y_j = 1 \end{cases} \right) \tag{G.8}$$

$$f(G_i) = \left\langle \frac{I_1(x_j)}{I_2(x_j)} \right\rangle_{y_j=0} + \frac{C}{1 + \sigma \left(\frac{I_1(x_j)}{I_2(x_j)} \right)_{y_j=0}} + \left\langle \frac{I_2(x_j)}{I_1(x_j)} \right\rangle_{y_j=1} + \frac{C}{1 + \sigma \left(\frac{I_2(x_j)}{I_1(x_j)} \right)_{y_j=1}} \tag{G.9}$$

where the $\langle \cdot \rangle_{y_j=0}$ represents the average over the samples with belonging to class 0, and $\sigma(\cdot)_{y_j=0}$ the standard deviation with respect to samples belonging to class 0. C is a hyperparameter, and can take any value. The terms with the standard deviation were introduced to give some regularization to the solution, that is, the best solution would result in a low standard deviation, so that all the intensity ratios would be similar.

$$f(G_i) = \sum_j^N \frac{1}{N} \left(\begin{cases} \left| \frac{1}{I_1(x_j) - 1} \right|, & y_j = 0 \\ \frac{1}{I_1(x_j)}, & y_j = 1 \end{cases} \right) \tag{G.10}$$

$$f(G_i) = \left\langle \left| \frac{1}{I_1(x_j) - 1} \right| \right\rangle_{y_j=0} + \frac{C}{1 + \sigma \left(\left| \frac{1}{I_1(x_j) - 1} \right| \right)_{y_j=0}} + \left\langle \frac{1}{I_1(x_j)} \right\rangle_{y_j=1} + \frac{C}{1 + \sigma \left(\frac{1}{I_1(x_j)} \right)_{y_j=1}} \tag{G.11}$$

$$f(G_i) = \sum_{j}^{N} \frac{1}{N} \left(\begin{cases} \frac{1}{1+e^{-C(I_1(x_j)-0.5)}}, & y_j = 0 \\ 1 - \frac{1}{1+e^{-C(I_1(x_j)-0.5)}}, & y_j = 1 \end{cases} \right) \tag{G.12}$$

$$f(G_i) = \sum_{j}^{N} \frac{1}{N} \left(\begin{cases} 2 + \frac{1}{1+e^{-C(I_1(x_j)-0.5)}} - \frac{1}{1+e^{-C(I_2(x_j)-0.5)}}, & y_j = 0 \\ 2 + \frac{1}{1+e^{-C(I_2(x_j)-0.5)}} - \frac{1}{1+e^{-C(I_1(x_j)-0.5)}}, & y_j = 1 \end{cases} \right) \tag{G.13}$$

Equations G.12, G.11 and G.10 aim at evaluating the class of the input merely by looking at the intensity of a single photodetector, whereas the rest intend to so by comparing the intensities of two distinct photodetectors.

Results and discussion

Binary modulation

For the binary modulation, we've used a maximum modulation area of 512x512 DMD pixels, consisting of 96x96=9216 macropixels. The algorithm ran for 30 generations with a total of 20 individuals per generation, 10 crossovers per generation and 10 survivals per generation. The initial and final mutation rates were kept at 0.1 and 0.001, and the mutation decay rate at 200. As for the fitness function, we've used equation G.13.

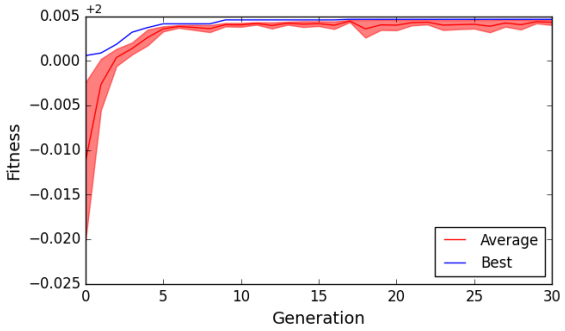

FIGURE G.5: Fitness function evolution.

From figure G.5 we can see that the algorithm converged rather quickly, however, this was not an indication of learning as can be seen when comparing figures G.7 and G.8, which can be further complemented by figure G.6.

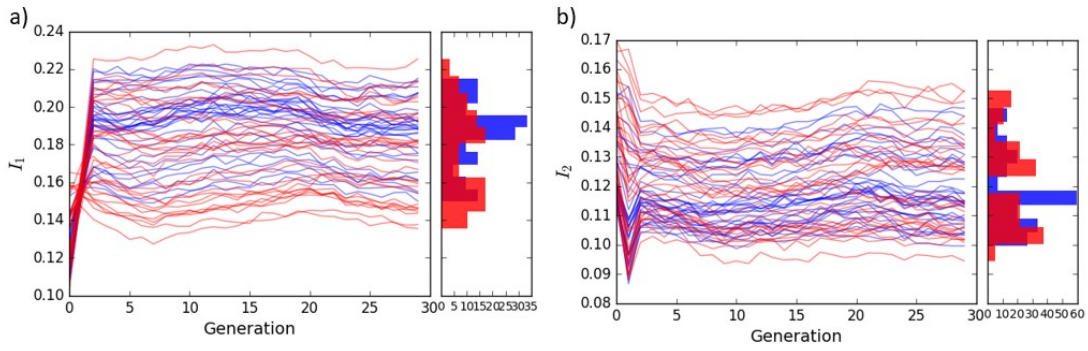

FIGURE G.6: I_1 and I_2 values for every sample, panels a) and b) respectively. Blue lines represent the evolution of the values of samples from class 0 and red lines are those of class 1.

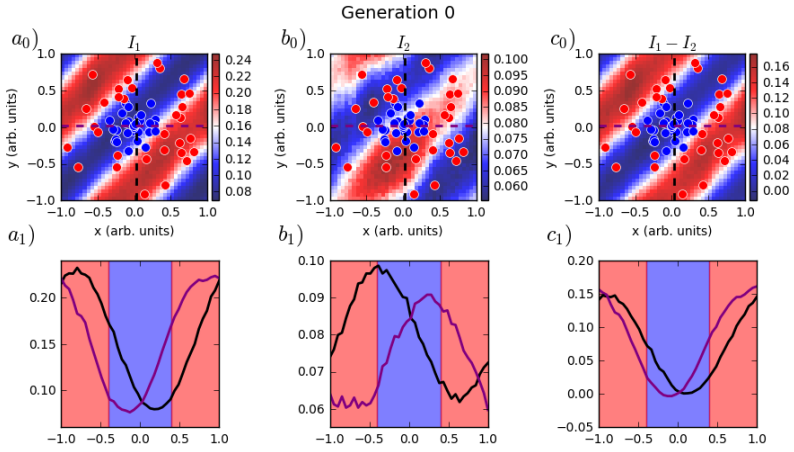

FIGURE G.7: Machine performance at the 0th generation. a_0) shows the value of I_1 for an input with arbitrary coordinates $\{x, y\}$ within the domain of the dataset in figure G.3, and a_1) shows a vertical (black) and a horizontal (purple) cross sections of the intensity map, with shaded regions of the boundaries for classification. b_0) and b_1) show the same information as the a) panels, but for I_2 and likewise for panels c) with $I_1 - I_2$.

Grayscale modulation

For the grayscale modulation, we've used a maximum modulation area of 512x512 DMD pixels, consisting of 96x96=9216 macropixels, each with 9 levels. The algorithm ran for 30 generations with a total of 20 individuals per generation, 10 crossovers per generation and 10 survivals per generation. The initial and final mutation rates were kept at 0.1 and 0.001, and the mutation decay rate at 200. As for the fitness function, we've used equation G.13.

The results have not improved, and the only remark to be done is with respect to figure G.14 which shows the problem that the GA is effectively trying to solve the focusing problem rather than the classification one.

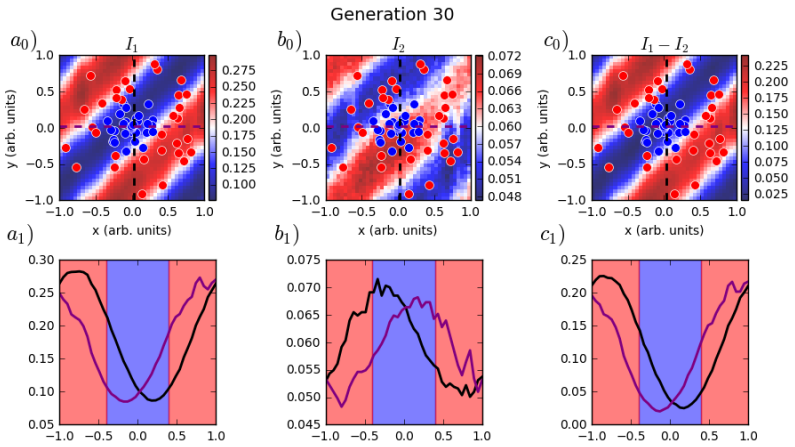

FIGURE G.8: Machine performance at the 30th generation. a_0) shows the value of I_1 for an input with arbitrary coordinates $\{x, y\}$ within the domain of the dataset in figure G.3, and a_1) shows a vertical (black) and a horizontal (purple) cross sections of the intensity map, with shaded regions of the boundaries for classification. b_0) and b_1) show the same information as the a) panels, but for I_2 and likewise for panels c) with $I_1 - I_2$.

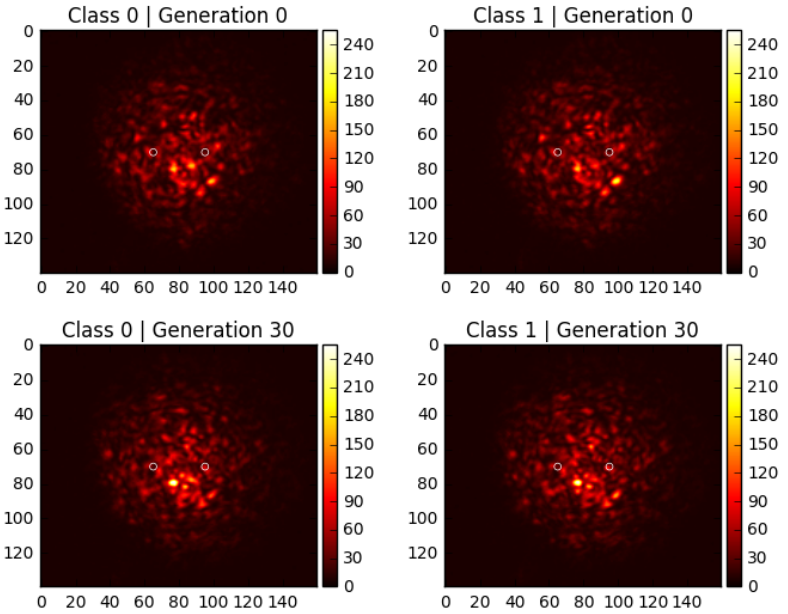

FIGURE G.9: Evolution of the speckle pattern for two samples of distinct classes.

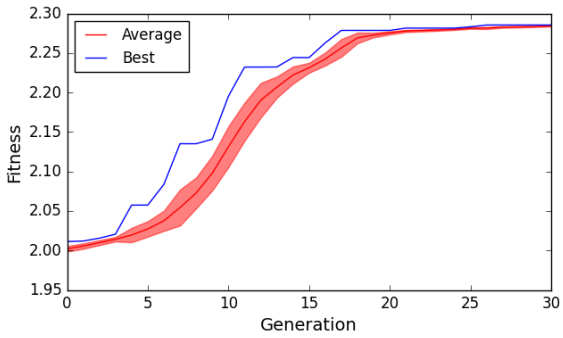

FIGURE G.10: Fitness function evolution.

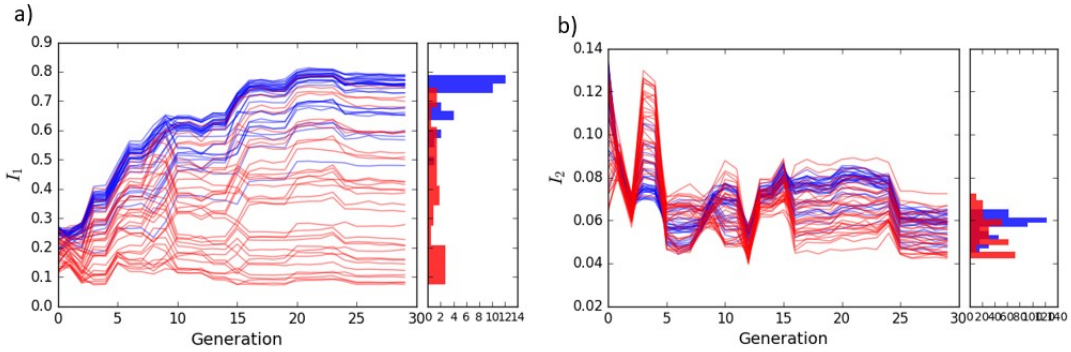

FIGURE G.11: I_1 and I_2 values for every sample, panels a) and b) respectively. Blue lines represent the evolution of the values of samples from class 0 and red lines are those of class 1.

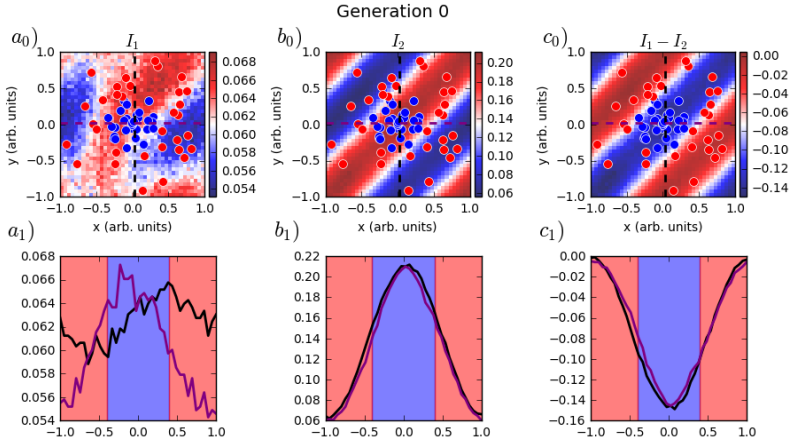

FIGURE G.12: Machine performance at the 0th generation. a_0) shows the value of I_1 for an input with arbitrary coordinates $\{x, y\}$ within the domain of the dataset in figure G.3, and a_1) shows a vertical (black) and a horizontal (purple) cross sections of the intensity map, with shaded regions of the boundaries for classification. b_0) and b_1) show the same information as the a) panels, but for I_2 and likewise for panels c) with $I_1 - I_2$.

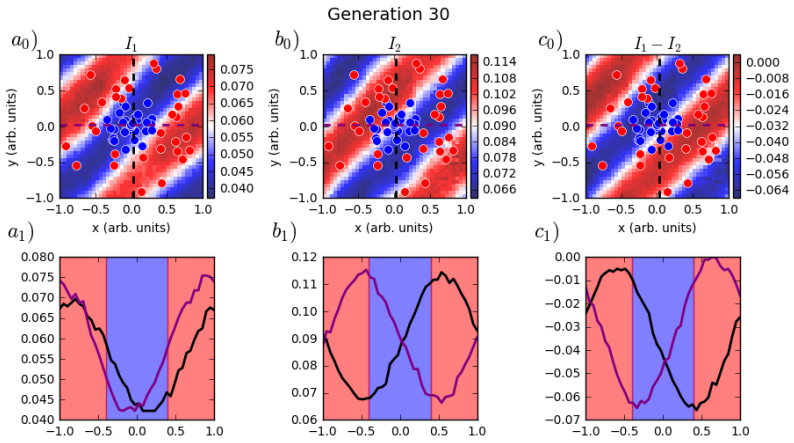

FIGURE G.13: Machine performance at the 30th generation. a_0) shows the value of I_1 for an input with arbitrary coordinates $\{x, y\}$ within the domain of the dataset in figure G.3, and a_1) shows a vertical (black) and a horizontal (purple) cross sections of the intensity map, with shaded regions of the boundaries for classification. b_0) and b_1) show the same information as the a) panels, but for I_2 and likewise for panels c) with $I_1 - I_2$.

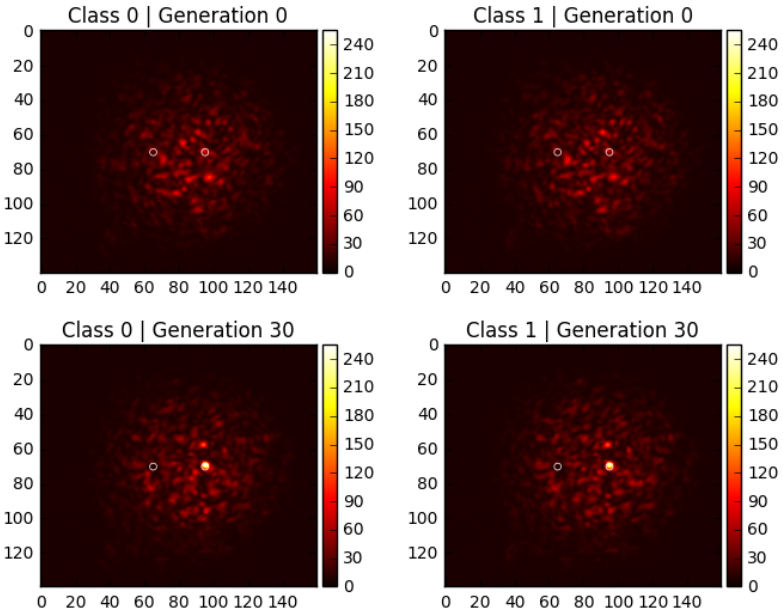

FIGURE G.14: Evolution of the speckle pattern for two samples of distinct classes.

Phase modulation

For the phase modulation, we've used a maximum modulation area of 512x512 DMD pixels, consisting of 32x32=1024 macropixels. The algorithm ran for 30 generations with a total of 20 individuals per generation, 10 crossovers per generation and 10 survivals per generation. The initial and final mutation rates were kept at 0.1 and 0.001, and the mutation decay rate at 200. As for the fitness function, we've used equation G.13.

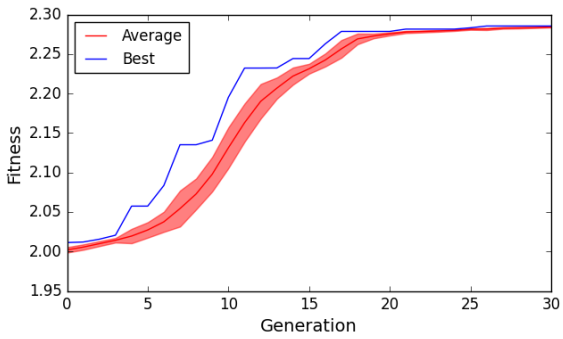

FIGURE G.15: Fitness function evolution.

The results have repeated themselves, and there are no further remarks except that, within all the above results, all of the fitness functions were tested, but all converged to the same solution. A greater spot on the target masks did not result in a better performance. Using saturation in the measurements did not give better results either.

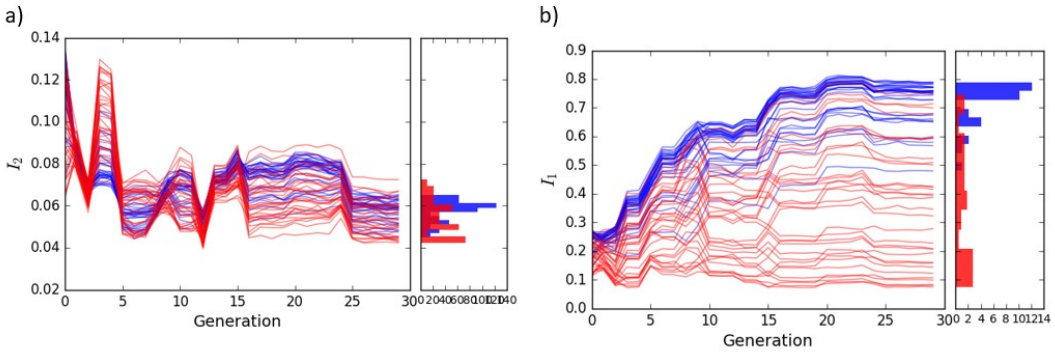

FIGURE G.16: I_1 and I_2 values for every sample, panels a) and b) respectively. Blue lines represent the evolution of the values of samples from class 0 and red lines are those of class 1.

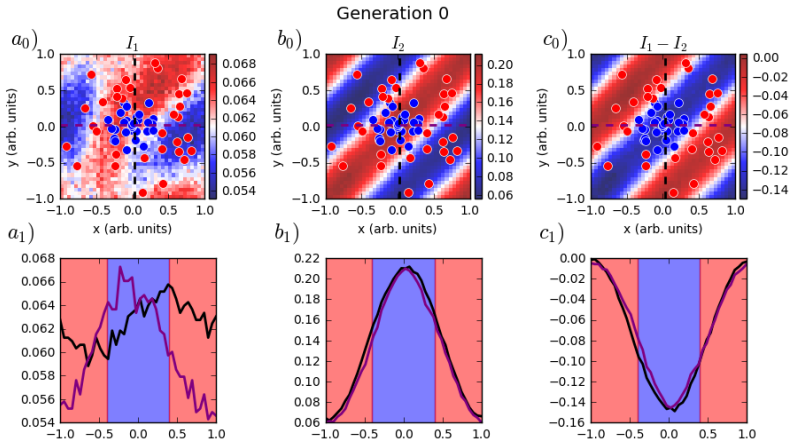

FIGURE G.17: Machine performance at the 0th generation. a_0) shows the value of I_1 for an input with arbitrary coordinates $\{x, y\}$ within the domain of the dataset in figure G.3, and a_1) shows a vertical (black) and a horizontal (purple) cross sections of the intensity map, with shaded regions of the boundaries for classification. b_0) and b_1) show the same information as the a) panels, but for I_2 and likewise for panels c) with $I_1 - I_2$.

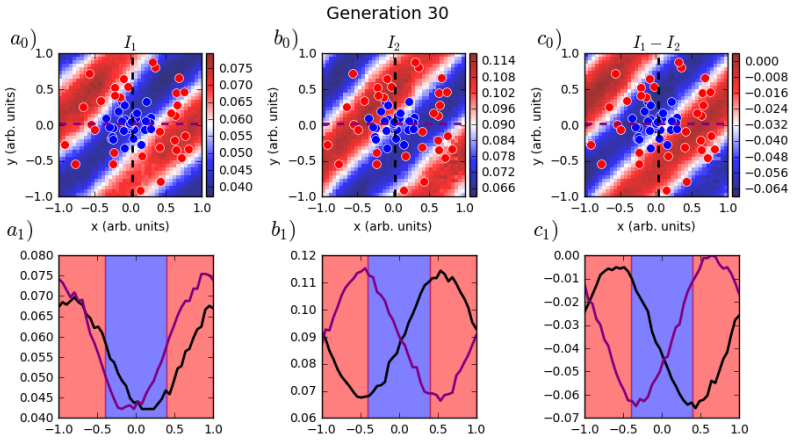

FIGURE G.18: Machine performance at the 30th generation. a_0) shows the value of I_1 for an input with arbitrary coordinates $\{x, y\}$ within the domain of the dataset in figure G.3, and a_1) shows a vertical (black) and a horizontal (purple) cross sections of the intensity map, with shaded regions of the boundaries for classification. b_0) and b_1) show the same information as the a) panels, but for I_2 and likewise for panels c) with $I_1 - I_2$.

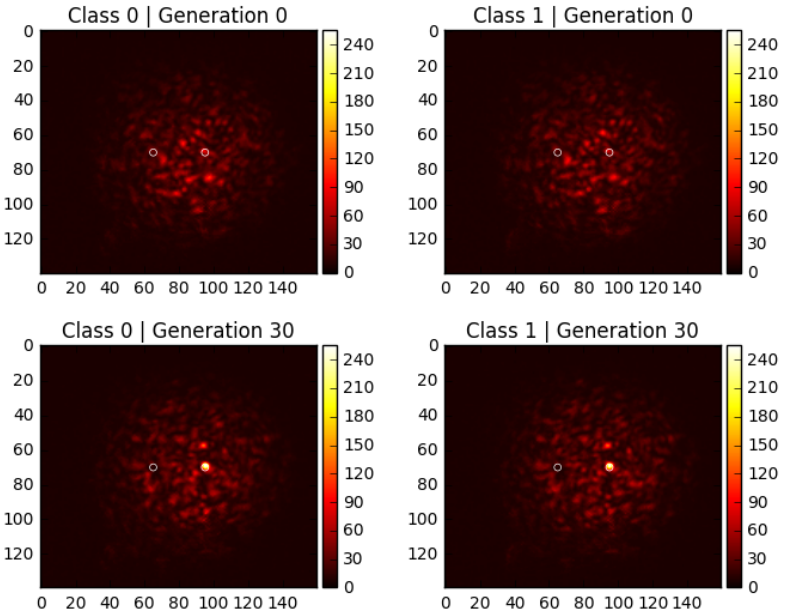

FIGURE G.19: Evolution of the speckle pattern for two samples of distinct classes.

Final remarks

Our approach has not been successful with respect to classification, but there are a few learning points that are worth highlighting which may lead to interesting future work since, to the best of our knowledge, the incorporation of the ELM framework with diffractive neural networks remains unexplored. First of all, an ELM relies on the projection towards a high dimensional space in a nonlinear manner. Despite not having a physical non-linearity in our system, the data encoding is in itself a nonlinear operation since we're doing it on the phase information, thus, had our system worked, it could still be categorised as an ELM. Secondly, even though we don't have a physical nonlinearity, we still had a projection activation function of sinusoidal nature, but clearly that was not enough. A way to improve this would be to add a physical nonlinearity through, for example, Kerr mediums or saturable absorbers, featuring fast response times so as to not retain memory of previous inputs. This way, we would greatly increase the dimensionality of the projection without relying on intensity measurements. Finally, it's important to note that we had drastically less parameters to optimise than a single diffractive layer of those present in references [40, 126]. Notably, they reported millions of parameters whereas we never got over 5000 parameters.

Bibliography

[1] "Physical aspects, operation of eniac are described." [Online]. Available: https://americanhistory.si.edu/comphist/

[2] J. v. Neumann, "First draft of a report on the edvac," Tech. Rep., 1945.

[3] "1956 nobel prize in physics." [Online]. Available: https://www.bell-labs.com/about/awards/1956-nobel-prize-physics/

[4] W. Jacobi, "Semiconductor amplifier," German Patent DE833 366, Apr., 1949. [Online]. Available: https://worldwide.espacenet.com/publicationDetails/biblio?locale=en_EP&CC=DE&NR=833366&rnd=1660043983728

[5] G. E. Moore, "Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff." *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33–35, 2006.

[6] R. Dennard, F. Gaensslen, H.-N. Yu, V. Rideout, E. Bassous, and A. LeBlanc, "Design of ion-implanted mosfet's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974.

[7] M. Irving, "Ibm's new 2-nm chips have transistors smaller than a strand of dna," May 2021. [Online]. Available: https://newatlas.com/computers/ibm-2-nm-chips-transistors/

[8] M. Horowitz, E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein, "Scaling, power, and the future of cmos," in *IEEE InternationalElectron Devices Meeting, 2005. IEDM Technical Digest.*, 2005, pp. 7 pp.–15.

[9] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *2011 38th Annual International Symposium on Computer Architecture (ISCA)*, 2011, pp. 365–376.

[10] 2022. [Online]. Available: https://irds.ieee.org/

[11] J. Shalf, "The future of computing beyond moore's law," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 378, no. 2166, p. 20190061, Jan. 2020.

[12] K. Rupp, "42 years of microprocessor trend data," Feb 2018. [Online]. Available: https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data

[13] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[14] O. V. Volodina and https://pnojournal.wordpress.com/2022/07/01/volodina-3/, "Formation of future teachers' worldview culture by means of foreign-language education," *P Sci Edu*, vol. 57, no. 3, pp. 126–159, Jul. 2022.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2016.

[16] D. Adiwardana, M.-T. Luong, D. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu, and Q. Le, "Towards a human-like open-domain chatbot," 01 2020.

[17] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," 04 2022.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[20] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell,

M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon, "In-datacenter performance analysis of a tensor processing unit," *SIGARCH Comput. Archit. News*, vol. 45, no. 2, p. 1–12, jun 2017.

[21] J. Hsu, "Nervana systems: Turning neural networks into a service [resources_startups]," vol. 53, no. 6, pp. 19–19.

[22] K. Lee, "Introducing big basin: Our next-generation ai hardware." [Online]. Available: https://engineering.fb.com/2017/03/08/data-center-engineering/introducing-big-basin-our-next-generation-ai-hardware/

[23] S. Published by Statista Research Department, "Total data volume worldwide 2010-2025," May 2022. [Online]. Available: https://www.statista.com/statistics/871513/worldwide-data-created/

[24] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

[25] A. Behl, "An introduction to machine learning," Dec 2019. [Online]. Available: https://becominghuman.ai/an-introduction-to-machine-learning-33a1b5d3a560

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[27] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, dec 2006.

[28] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32–48, jan 2015.

[29] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, apr 2012.

[30] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*. IEEE, 2004.

[31] R. Xu, P. Lv, F. Xu, and Y. Shi, "A survey of approaches for implementing optical neural networks," vol. 136, p. 106787.

[32] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities." vol. 79, no. 8, pp. 2554–2558.

[33] D. Psaltis and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," *Optics Letters*, vol. 10, no. 2, pp. 98–100, 1985.

[34] *Photonic Reservoir Computing*. Gruyter, Walter de GmbH, Jul. 2019. [Online]. Available: https://www.ebook.de/de/product/38427878/photonic_reservoir_computing.html

[35] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," vol. 115, pp. 100–123.

[36] J. W. Goodman, A. Dias, and L. Woody, "Fully parallel, high-speed incoherent optical method for performing discrete fourier transforms," *Optics Letters*, vol. 2, no. 1, pp. 1–3, 1978.

[37] Y. Nitta, J. Ohta, S. Tai, and K. Kyuma, "Optical learning neurochip with internal analog memory," vol. 32, no. 8, p. 1264.

[38] D. Psaltis, D. Brady, and K. Wagner, "Adaptive optical networks using photorefractive crystals," vol. 27, no. 9, p. 1752.

[39] H. Lee, X. guang Gu, and D. Psaltis, "Volume holographic interconnections with maximal capacity and minimal cross talk," vol. 65, no. 6, pp. 2191–2194.

[40] D. Mengu, Y. Luo, Y. Rivenson, and A. Ozcan, "Analysis of diffractive optical neural networks and their integration with electronic neural networks," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–14, 2020.

[41] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, "Deep learning with coherent nanophotonic circuits," vol. 11, no. 7, pp. 441–446.

[42] Z. Wu, M. Zhou, E. Khoram, B. Liu, and Z. Yu, "Neuromorphic metasurface," vol. 8, no. 1, p. 46.

[43] I. Saxena and E. Fiesler, "Adaptive multilayer optical neural network with optical thresholding," *Optical Engineering*, vol. 34, no. 8, pp. 2435–2440, 1995.

[44] M. Brambilla, L. Lugiato, M. Pinna, F. Prati, P. Pagani, P. Vanotti, M. Y. Li, and C. Weiss, "The laser as nonlinear element for an optical associative memory," *Optics communications*, vol. 92, no. 1-3, pp. 145–164, 1992.

[45] I. A. Williamson, T. W. Hughes, M. Minkov, B. Bartlett, S. Pai, and S. Fan, "Reprogrammable electro-optic nonlinear activation functions for optical neural networks," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–12, 2019.

[46] L. G. Wright, T. Onodera, M. M. Stein, T. Wang, D. T. Schachter, Z. Hu, and P. L. McMahon, "Deep physical neural networks trained with backpropagation," vol. 601, no. 7894, pp. 549–555.

[47] A. Saade, F. Caltagirone, I. Carron, L. Daudet, A. Dremeau, S. Gigan, and F. Krzakala, "Random projections through multiple optical scattering: Approximating kernels at the speed of light," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, mar 2016.

[48] N. A. Silva, T. D. Ferreira, and A. Guerreiro, "Reservoir computing with solitons," *New Journal of Physics*, vol. 23, no. 2, p. 023013, feb 2021.

[49] G. Marcucci, D. Pierangeli, and C. Conti, "Theory of neuromorphic computing by waves: Machine learning by rogue waves, dispersive shocks, and solitons," *Physical Review Letters*, vol. 125, no. 9, p. 093901, aug 2020.

[50] D. Pierangeli, G. Marcucci, and C. Conti, "Photonic extreme learning machine by free-space optical propagation," *Photonics Research*, vol. 9, no. 8, p. 1446, jul 2021.

[51] S. Sunada, K. Kanno, and A. Uchida, "Using multidimensional speckle dynamics for high-speed, large-scale, parallel photonic computing," *Optics Express*, vol. 28, no. 21, p. 30349, sep 2020.

[52] U. Teğin, M. Yıldırım, İ. Oğuz, C. Moser, and D. Psaltis, "Scalable optical learning operator," *Nature Computational Science*, vol. 1, no. 8, pp. 542–549, aug 2021.

[53] K. Vandoorne, P. Mechet, T. V. Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature Communications*, vol. 5, no. 1, mar 2014.

[54] D. Silva, N. A. Silva, T. D. Ferreira, C. C. Rosa, and A. Guerreiro, "Unravelling an optical extreme learning machine," vol. 266, p. 13034.

[55] A. Teixeira, "Iberic meeting of optics students 2022 - book of abstracts."

[56] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, *Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming*. Springer Netherlands, pp. 151–170. [Online]. Available: https://doi.org/10.1007/978-94-009-0279-4_9

[57] user194703, "Plot gradient descent," Mathematics Stack Exchange. [Online]. Available: https://tex.stackexchange.com/a/544832

[58] Wikipedia contributors, "Neuron — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Neuron&oldid=1101029060, 2022, [Online; accessed 5-September-2022].

[59] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16-18, pp. 3056–3062, oct 2007.

[60] ——, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16-18, pp. 3460–3468, oct 2008.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau,

M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[62] J. W. Goodman, *Speckle Phenomena in Optics: Theory and Applications, Second Edition*. SPIE, jan 2020.

[63] S. M. Popoff, G. Lerosey, R. Carminati, M. Fink, A. C. Boccara, and S. Gigan, "Measuring the transmission matrix in optics: An approach to the study and control of light propagation in disordered media," *Phys. Rev. Lett.*, vol. 104, p. 100601, Mar 2010.

[64] R. A. Horn, *Matrix Analysis: Second Edition*. Cambridge University Press.

[65] D. Duncan and S. Kirkpatrick, "Algorithms for simulation of speckle (laser and otherwise)," 01 2008.

[66] Wikipedia contributors, "Singular value decomposition — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Singular_value_decomposition&oldid=1111597286, 2022, [Online; accessed 22-September-2022].

[67] V. GmbH. [Online]. Available: https://www.vialux.de/en/hi-speed-specification.html

[68] W.-H. Lee, "Binary computer-generated holograms," *Appl. Opt.*, vol. 18, no. 21, pp. 3661–3669, Nov 1979. [Online]. Available: http://opg.optica.org/ao/abstract.cfm?URI=ao-18-21-3661

[69] XIMEA, "Ximea - mq013mg-on - product page." [Online]. Available: https://www.ximea.com/en/products/usb3-vision-cameras-xiq-line/mq013mg-on

[70] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[71] S. M. Popoff, G. Shih, D. B., and GustavePariente, "wavefrontshaping/alp4lib: 1.0.1," Feb. 2022.

[72] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds. IOS Press, 2016, pp. 87 – 90.

[73] Wikipedia contributors, "Weyl's inequality — Wikipedia, the free encyclopedia," 2022, [Online; accessed 28-April-2022]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Weyl%27s_inequality&oldid=1065059511

[74] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[75] R. W. Boyd, *Nonlinear Optics, Third Edition*, 3rd ed. USA: Academic Press, Inc., 2008.

[76] J. Lambers, "Minimum norm solutions of underdetermined systems." [Online]. Available: https://www.math.usm.edu/lambers/mat419/lecture15.pdf

[77] Wikipedia contributors, "Gram matrix — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Gram_matrix&oldid=1093332712, 2022, [Online; accessed 6-September-2022].

[78] C. Stover, "Fundamental theorem of linear algebra." [Online]. Available: https://mathworld.wolfram.com/FundamentalTheoremofLinearAlgebra.html

[79] user438618, "On the equality of the rank-sum inequality," Mathematics Stack Exchange, uRL:https://math.stackexchange.com/q/2327071 (version: 2017-06-18). [Online]. Available: https://math.stackexchange.com/q/2327071

[80] Z. Zhang, Z. You, and D. Chu, "Fundamentals of phase-only liquid crystal on silicon (lcos) devices," *Light: Science and Applications*, vol. 3, no. 10, pp. e213–e213, 2014.

[81] Wikipedia contributors, "Liquid crystal — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Liquid_crystal&oldid=1071037826, 2022, [Online; accessed 14-February-2022].

[82] Holoeye, *PLUTO Phase-only spatial light modulators*, v2.6 ed., 2017.

[83] A. Bergeron, J. Gauvin, F. Gagnon, D. Gingras, H. H. Arsenault, and M. Doucet, "Phase calibration and applications of a liquid-crystal spatial light modulator," *Appl. Opt.*, vol. 34, no. 23, pp. 5133–5139, Aug 1995. [Online]. Available: http://opg.optica.org/ao/abstract.cfm?URI=ao-34-23-5133

[84] Wikipedia contributors, "Holography in fiction — Wikipedia, the free encyclopedia," [Online; accessed 23-November-2021]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Holography_in_fiction&oldid=1053671804

[85] D. GABOR, "A new microscopic principle," vol. 161, no. 4098, pp. 777–778. [Online]. Available: https://doi.org/10.1038/161777a0

[86] D. Gabor and W. L. Bragg, "Microscopy by reconstructed wave-fronts," vol. 197, no. 1051, pp. 454–487. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1949.0075

[87] F. L. Pedrotti, L. M. Pedrotti, and L. S. Pedrotti, *Introduction to Optics*, 3rd ed. Cambridge University Press.

[88] L. H. Enloe, J. A. Murphy, and C. B. Rubinstein, "B.s.t.j. briefs hologram transmission via television," vol. 45, no. 2, pp. 335–339.

[89] B. Javidi, A. Carnicer, A. Anand, G. Barbastathis, W. Chen, P. Ferraro, J. W. Goodman, R. Horisaki, K. Khare, M. Kujawinska, R. A. Leitgeb, P. Marquet, T. Nomura, A. Ozcan, Y. Park, G. Pedrini, P. Picart, J. Rosen, G. Saavedra, N. T. Shaked, A. Stern, E. Tajahuerce, L. Tian, G. Wetzstein, and M. Yamaguchi, "Roadmap on digital holography (invited)," vol. 29, no. 22, pp. 35 078–35 118. [Online]. Available: http://www.osapublishing.org/oe/abstract.cfm?URI=oe-29-22-35078

[90] T. M. Kreis, M. Adams, and W. P. O. Jueptner, "Methods of digital holography: a comparison," in *Optical Inspection and Micromeasurements II*, C. Gorecki, Ed., vol. 3098, International Society for Optics and Photonics. SPIE, pp. 224 – 233. [Online]. Available: https://doi.org/10.1117/12.281164

[91] P. Picart and J. Leval, "General theoretical formulation of image formation in digital fresnel holography," vol. 25, no. 7, pp. 1744–1761. [Online]. Available: http://www.osapublishing.org/josaa/abstract.cfm?URI=josaa-25-7-1744

[92] E. Cuche, P. Marquet, and C. Depeursinge, "Spatial filtering for zero-order and twin-image elimination in digital off-axis holography," *Applied Optics*, vol. 39, no. 23, p. 4070, aug 2000.

[93] PyQT, "Pyqt reference guide," 2012. [Online]. Available: http://www.riverbankcomputing.com/static/Docs/PyQt4/html/index.html

[94] L. Campagnola, "Pyqtgraph." [Online]. Available: https://pyqtgraph.readthedocs.io/en/latest/

[95] G. Mie, "Beiträge zur optik trüber medien, speziell kolloidaler metallösungen," *Annalen der Physik*, vol. 330, no. 3, pp. 377–445, 1908.

[96] HyperPhysics. Mie scattering. [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/atmos/blusky.html

[97] D. D. Nolte, *Galileo Unbound: A Path Across Life, the Universe and Everything*. OXFORD UNIV PR, Sep. 2018. [Online]. Available: https://www.ebook.de/de/product/31891239/david_d_nolte_galileo_unbound_a_path_across_life_the_universe_and_everything.html

[98] L. Allen and D. Jones, "An analysis of the granularity of scattered optical maser light," *Physics Letters*, vol. 7, no. 5, pp. 321–323, dec 1963.

[99] M. Takeda, W. Wang, and S. G. Hanson, "Polarization speckles and generalized stokes vector wave: a review," in *SPIE Proceedings*, J. Armando Albertazzi Goncalves and G. H. Kaufmann, Eds. SPIE, sep 2010.

[100] S. M. Popoff, G. Lerosey, M. Fink, A. C. Boccara, and S. Gigan, "Controlling light through optical disordered media: transmission matrix approach," *New Journal of Physics*, vol. 13, no. 12, p. 123021, dec 2011.

[101] H. Yu, T. R. Hillman, W. Choi, J. O. Lee, M. S. Feld, R. R. Dasari, and Y. Park, "Measuring large optical transmission matrices of disordered media," *Physical Review Letters*, vol. 111, no. 15, p. 153902, oct 2013.

[102] S. Li, C. Saunders, D. J. Lum, J. Murray-Bruce, V. K. Goyal, T. Čižmár, and D. B. Phillips, "Compressively sampling the optical transmission matrix of a multimode fibre," *Light: Science & Applications*, vol. 10, no. 1, apr 2021.

[103] M. Plöschner, T. Tyc, and T. Čižmár, "Seeing through chaos in multimode fibres," *Nature Photonics*, vol. 9, no. 8, pp. 529–535, jul 2015.

[104] P. Pai, J. Bosch, and A. P. Mosk, "Optical transmission matrix measurement sampled on a dense hexagonal lattice," *OSA Continuum*, vol. 3, no. 3, p. 637, mar 2020.

[105] A. P. Mosk, A. Lagendijk, G. Lerosey, and M. Fink, "Controlling waves in space and time for imaging and focusing in complex media," *Nature Photonics*, vol. 6, no. 5, pp. 283–292, may 2012.

[106] P. Pai, J. Bosch, and A. P. Mosk, "Resampling the transmission matrix in an aberration-corrected bessel mode basis," *Optics Express*, vol. 29, no. 1, p. 24, dec 2020.

[107] W. Winkler, R. Schilling, K. Danzmann, J. Mizuno, A. Rüdiger, and K. A. Strain, "Light scattering described in the mode picture," *Applied Optics*, vol. 33, no. 31, p. 7547, nov 1994.

[108] O. Svelto, *Principles of Lasers*. SPRINGER NATURE, Dec. 2009. [Online]. Available: https://www.ebook.de/de/product/8633528/orazio_svelto_principles_of_lasers.html

[109] A. Drémeau, A. Liutkus, D. Martina, O. Katz, C. Schülke, F. Krzakala, S. Gigan, and L. Daudet, "Reference-less measurement of the transmission matrix of a highly scattering material using a DMD and phase retrieval techniques," *Optics Express*, vol. 23, no. 9, p. 11898, apr 2015.

[110] T. STATHAKI, "Digital image processing 2019 lecture notes." [Online]. Available: https://www.commsp.ee.ic.ac.uk/~tania/teaching/DIP2014/DIPDHT2018.pdf

[111] Y. Gao, R. Li, and L. Cao, "Self-referenced multiple-beam interferometric method for robust phase calibration of spatial light modulator," *Optics Express*, vol. 27, no. 23, p. 34463, nov 2019.

[112] G. zhen Yang, B. zhen Dong, B. yuan Gu, J. yao Zhuang, and O. K. Ersoy, "Gerchberg–saxton and yang–gu algorithms for phase retrieval in a nonunitary transform system: a comparison," *Applied Optics*, vol. 33, no. 2, p. 209, jan 1994.

[113] C. Zhang, M. Wang, Q. Chen, D. Wang, and S. Wei, "Two-step phase retrieval algorithm using single-intensity measurement," *International Journal of Optics*, vol. 2018, pp. 1–7, oct 2018.

[114] A. Dremeau and F. Krzakala, "Phase recovery from a bayesian point of view: The variational approach," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, apr 2015.

[115] V. A. Marčenko and L. A. Pastur, "DISTRIBUTION OF EIGENVALUES FOR SOME SETS OF RANDOM MATRICES," *Mathematics of the USSR-Sbornik*, vol. 1, no. 4, pp. 457–483, apr 1967.

[116] R. Boyd, *Nonlinear optics*. Amsterdam Boston: Academic Press, 2008.

[117] I. M. Vellekoop and A. P. Mosk, "Focusing coherent light through opaque strongly scattering media," *Optics Letters*, vol. 32, no. 16, p. 2309, aug 2007.

[118] I. Vellekoop and A. Mosk, "Phase control algorithms for focusing light through turbid media," *Optics Communications*, vol. 281, no. 11, pp. 3071–3080, jun 2008.

[119] M. Cui, "Parallel wavefront optimization method for focusing light through random scattering media," *Optics Letters*, vol. 36, no. 6, p. 870, mar 2011.

[120] W. B. Bridges, P. T. Brunner, S. P. Lazzara, T. A. Nussmeier, T. R. O'Meara, J. A. Sanguinet, and W. P. Brown, "Coherent optical adaptive techniques," *Applied Optics*, vol. 13, no. 2, p. 291, feb 1974.

[121] D. B. Conkey, A. N. Brown, A. M. Caravaca-Aguirre, and R. Piestun, "Genetic algorithm optimization for focusing through turbid media in noisy environments," *Optics Express*, vol. 20, no. 5, p. 4840, feb 2012.

[122] P. H. Winston, *Artificial intelligence*. Addison-Wesley Pub. Co., 1992.

[123] N. A. Fomin, *Statistical Properties of Speckle Patterns*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 27–41.

[124] D. Cojoc and A. Alexandrescu, "Optimization of computer-generated binary holograms using genetic algorithms," in *SPIE Proceedings*, O. V. Angelsky, Ed. SPIE, nov 1999.

[125] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," *Science*, vol. 361, no. 6406, pp. 1004–1008, sep 2018.

[126] H. Chen, J. Feng, M. Jiang, Y. Wang, J. Lin, J. Tan, and P. Jin, "Diffractive deep neural networks at visible wavelengths," *Engineering*, vol. 7, no. 10, pp. 1483–1491, oct 2021.