Automatic tuning of photonic beamformers

A data-driven approach

Bliek, Laurens

DOI
10.4233/uuid:8bf73354-7c68-4512-8c2b-a5f060e783f4

Publication date
2019

Document Version
Final published version

Citation (APA)
Bliek, L. (2019). *Automatic tuning of photonic beamformers: A data-driven approach*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:8bf73354-7c68-4512-8c2b-a5f060e783f4

AUTOMATIC TUNING OF PHOTONIC BEAMFORMERS
A DATA-DRIVEN APPROACH

AUTOMATIC TUNING OF PHOTONIC BEAMFORMERS A DATA-DRIVEN APPROACH

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 9 mei 2019 om 12:30 uur

door

Laurens BLIEK

ingenieur in de toegepaste wiskunde, Technische Universiteit Delft,
geboren te Amsterdam, Nederland.

Dit proefschrift is goedgekeurd door de

promotor: prof. dr. ir. M. Verhaegen
copromotor: dr. ir. S. Wahls

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. M. Verhaegen,	Technische Universiteit Delft
Dr. ir. S. Wahls,	Technische Universiteit Delft
Onafhankelijke leden:	
Prof. dr. ir. M. Reinders,	Technische Universiteit Delft
Prof. dr. ir. J. Suykens,	Katholieke Universiteit Leuven, België
Prof. dr. K. Boller,	Universiteit Twente
Dr. ir. H. Driessen,	Technische Universiteit Delft
Reservelid:	
Prof. dr. ir. J. Hellendoorn,	Technische Universiteit Delft
Overige leden:	
Dr. ir. C.G.H. Roeloffzen,	LioniX International

Dr. ir. C.G.H. Roeloffzen heeft in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.

Keywords: Photonic beamforming, microwave photonics, surrogate modeling, machine learning, costly and noisy optimization

Printed by: Gildeprint

Cover by: Laurens Bliek

ISBN 978-94-6323-538-9

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

In loving memory of my mother

ACKNOWLEDGEMENTS

If you are reading this, you are probably interested in the person behind this thesis, and not just in the scientific results. I can confidently say that that person, me, would not have been able to write this thesis without the love and support of multiple people.

Michel Verhaegen, thank you very much for your support as my promotor. We have been closely involved in this project but also in the many side activities I have done during my time at the Delft Center for Systems and Control (DCSC). I have learned a lot from you as a scientist and as an amazing person. Sander Wahls, the same counts for you as my co-promotor. Thank you very much for all your help, and for letting me walk into your office so many times for those quick questions or when I needed advice. It was a pleasure to work together with both of you and it was wonderful to transform into an independent researcher under your supervision.

Thanks to all my colleagues at DCSC for providing a fun and stimulating environment. Hans Verstraete, I really enjoyed our collaboration on very different applications with very similar solutions. Amir, Anqi, Arman, Baptiste, Bart, Carlas, Cees, Chengpu, Dean, Dieky, Edwin, Elisabeth, Farid, Gleb, Hai, Hans, Hildo, Jan-Willem, Jelmer, Jens, Jeroen, Joeri, Kim, Le, Mohammad, Nico, Nikos, Niloofar, Noortje, Oleg, Paolo, Peter, Pieter G., Pieter P., Raf, Reinier, Renshi, Ruxandra, Sachin, Sadegh, Shrinivas, Shuai, Simone, Sjoerd, Subramanya, Thao, Tijmen, Tim, Tope, Vahab, Vishal, Yasin, Yu, Zhou: thank you all for giving me a wonderful time at DCSC either by playing foosball with me, or board games, by having lunch with me, by talking about research or other matters with me or by just providing small talk, or by collaborating with me in some way or another.

Kitty, Marieke, Heleen, Kiran, Olaf, Ditske and Erica, thank you all for your incredible support in all kinds of practical or financial matters.

Hermin, Maurice, Jimmy, Iris, Riemer and Jóhann, thank you for letting me supervise your graduation projects these past years. It was great to see you grow and I learned a lot from the research you did and from being your supervisor.

Thanks to the people at LioniX International, in particular Chris, Ruud, Ilka, Roelof, Jörn, and Caterina. You were very helpful and hospitable during my several visits to the far but beautiful Twente region.

Thanks to the people in my user committee for being interested in this research project. Your questions and insights were very helpful.

Thanks to the people in my defence committee for your interest in this thesis and for your questions.

Thanks to all my friends and family, whether close or far, new or old, for your love and support. I feel truly blessed to have all of you in my life.

Thank you Lord Jesus, for all that you have done in my life, for giving me the talents to perform this work, and for being an amazing friend in the good and bad times these past years.

CONTENTS

1

INTRODUCTION

This chapter gives an introduction to the application that is considered in this thesis: photonic beamforming for aircraft-satellite communication. This is done by explaining the concept of beamforming, which is a signal processing technique, in the first section. The section proceeds by explaining the concept of photonic beamforming, as well as the current challenges in this field. In the second section, the core idea of this thesis is introduced: to provide a data-driven automatic tuning method for photonic beamforming using surrogate models. The third and final section shows an outline of this thesis.

1

1.1. PHOTONIC BEAMFORMING FOR AIRCRAFT-SATELLITE COMMUNICATION

WIRELESS communication systems have been playing an important role in our daily lives for many years. The need to stay connected has been increasing and will probably continue to increase, until it is possible to surf the world wide web, contact other people, or watch live television from anywhere on the planet. Though the required technology is already available, one example where there are more challenges than normal is the implementation of telecommunication techniques on moving vehicles such as trains and aircrafts. Especially for aircrafts on intercontinental flights there is the particular challenge that no ground connections are available when flying over sea. In this case, satellite connections are a logical alternative. In order to establish a satellite connection, a high-gain antenna needs to be mounted on the aircraft. However, typical high-gain antannae like dish antennae need to be put in a separate radome for protection, causing aerodynamic resistance (and therefore more fuel consumption) and requiring adaptations to the structure of the aircraft. Furthermore, the antenna has to be mechanically steered towards the satellite all the time, but the required mechanical parts can wear down and require regular maintenance.

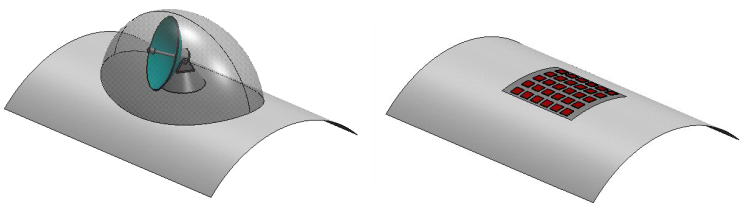


Figure 1.1: Two possible antennae for aircraft-satellite communication: a mechanically steered dish antenna (top), which needs to be protected when mounted on top of an aircraft, and an electronically steered phased array antenna (bottom), which can be integrated into the body of the aircraft. Adapted from [1].

As an alternative, an electronically steered phased array antenna has been proposed that can be integrated in the body of the aircraft [1]. See Figure 1.1. This type of antenna makes use of the concept of beamforming, explained in the remainder of this chapter, and has many advantages compared to a dish antenna: there is no aerodynamic drag and there are no movable parts, allowing for a high tuning speed and accuracy. These advantages stem from the fact that a phased arra antenna consists of an array of antenna elements that send (for a transmit antenna array) or receive (for a receive antenna array) the same signal with a certain phase difference in such a way that the corresponding wavefront travels in the desired direction. Changing the direction of the wavefront is done by changing the phase difference between the antenna elements. This is done with a beamformer.

Recent advances in the field of microwave photonics [2], a field where radio-frequency (RF) signals are processed in the optical domain, gave rise to photonic beamformers that

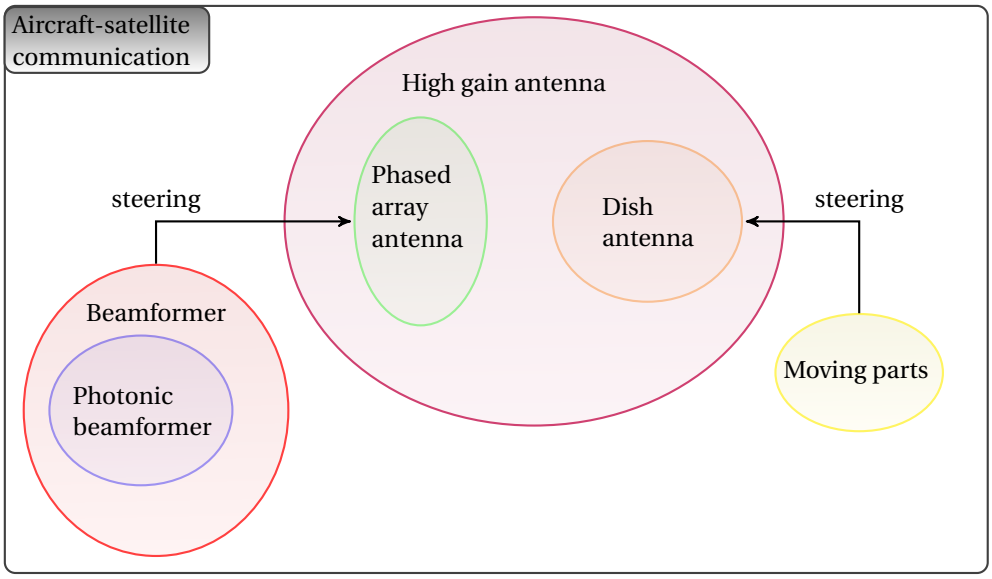


Figure 1.2: The purpose of photonic beamformers in this thesis explained. Photonic beamformers are a type of beamformer, which is a system that is used to steer a phased array antenna. Just like a dish antenna, a phased array antenna is a type of high gain antenna and can therefore be used for the purpose of aircraft-satellite communication. However, phased array antennae do not rely on mechanical movement for steering, giving them many advantages compared to dish antennae.

change the phase or group delay response of each antenna element in the optical domain. Photonic beamformers have several advantages over their electronic counterparts such as low size, low weight, low loss, large bandwidth, and immunity to electromagnetic interference. These advantages are even greater for photonic beamformers that are integrated on a chip [3]. Such a beamformer has been proposed in earlier work [4], and this system will be the main system under consideration in this thesis. See Figure 1.2 for the relations between the systems described in this chapter.

The remainder of this chapter is organized as follows. Section 1.1.1 explains the concept of beamforming for phased array antennae and how they can be steered towards a satellite without any movable parts. Section 1.1.2 explains the advantages of photonic beamforming and gives a description of the full phased array antenna system with an integrated photonic beamformer. Section 1.1.3 provides further details on the type of photonic beamformer considered in this thesis. Section 1.1.4 explains the main challenges to get this beamformer system to work properly for the application of aircraft-satellite communication. Section 1.2 explains the approach taken in this thesis to tackle these challenges. This chapter finishes with the outline of the thesis in Section 1.3.

1.1.1. BEAMFORMING

Beamforming is a signal processing technique used to steer the direction of a phased array antenna. Since a phased array antenna consists of several antenna elements close to-

1

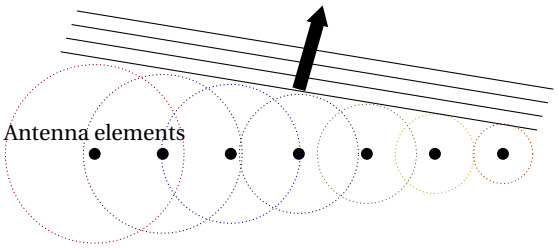


Figure 1.3: Beamforming for a phased array antenna explained. Each antenna element transmits the same signal after a certain time delay. This delay can be chosen in such a way that constructive interference occurs in a certain direction, making it possible to transmit a highly directional signal with a high gain and a focussed beam.

gether, and each element is an omnidirectional antenna, the signal arriving at or leaving the array can be heavily interfered because of a phase mismatch between the antenna elements. However, by adapting the phase or time delay of each antenna element, constructive interference occurs in a certain direction, while destructive interference occurs in the other directions. See Figure 1.3 for an illustration of beamforming with a transmit phased array antenna. The same concept holds for a receive antenna. The relation between phase and time delay is explained later in this section.

The beamformer system adds either a phase shift or a time delay to each antenna element. It matters a lot which of the two is used: time delays allow the antenna to operate under much larger bandwidths [5]. This is because of the effect of the time delay or phase shift on the beam direction. Suppose that several identical antenna elements are positioned along a single line. Let d be the distance between two consecutive antenna elements and let c be the speed of light in vacuum. Then the beam angle θ of the wavefront generated by the phased array antenna is given by

$$\theta = \sin^{-1}\left(\frac{c\Delta t}{d}\right) \tag{1.1}$$

for a time delay Δt between two consecutive antenna elements and

$$\theta = \sin^{-1}\left(\frac{c\Delta\phi}{-2\pi d f}\right) \tag{1.2}$$

for a phase shift $\Delta\phi$ between two consecutive antenna elements. Here, f is the frequency of the RF signal. The phased array antenna is steered by varying either the phase shift or time delay of each antenna element in such a way that the beam angle is changed to the desired direction.

In general, the phase and time delay of a signal are related as follows: if a signal $y(t)$ is a delayed version of a signal $x(t)$, that is, $y(t) = x(t-\Delta t)$, then this can be described in Fourier domain as $Y(j\omega) = e^{-j\omega\Delta t}X(j\omega)$. The phase of Y is then equal to $\angle Y(j\omega) = \angle\left\{e^{-j\omega\Delta t}e^{j\angle X(j\omega)}|X(j\omega)|\right\} = -\omega\Delta t + \angle X(j\omega)$. In other words, the phase difference between two delayed signals is a linear function of the frequency, with slope $-\Delta t$ in the case of angular frequency ω, and slope $-2\pi\Delta t$ in the case of frequency f. This is

where the term $-2\pi f$ in (1.2) comes from. Note: the negative derivative of the phase with respect to angular frequency is called the group delay.

From (1.1)-(1.2) it can be seen that the beam angle depends on the frequency of the signal if phase shifters are used, but not if a time delay is used. A frequency-dependent beam angle has the undesirable effect that the gain is decreased and the beam width is increased for systems operating under a large bandwidth. On the other hand, using a time delay instead of phase shifters makes the beam angle independent of the frequency, allowing the system to operate under much larger bandwidths. This is a necessity in modern applications like aircraft-satellite communication. Conventional beamformers make use of phase shifters that provide a phase shift $\Delta\phi$ as a constant function of the frequency and are therefore not fit for these applications. In contrast, so called true time delay systems [5] provide a phase shift that is a linear function of the frequency f in the bandwidth of interest, with a slope equal to $-2\pi\Delta t$, making (1.1) and (1.2) equivalent. Section 1.1.2 shows how such a linear phase response can be achieved.

1.1.2. PHOTONIC BEAMFORMING

One way to provide a linear phase response for each antenna element is via photonic beamforming. As mentioned earlier, photonic beamformers have many advantages compared to electronic beamformers, such as high bandwidth and low loss. In photonic beamforming, the signal processing is done in the optical domain. For the case of a transmit phased array antenna, this means that the signal to be transmitted is first converted from the RF frequency range to the optical frequency range via optical modulation. Then the signal is split into multiple paths that each get a frequency-dependent phase shift, after which it is converted back to an electrical signal using photodetectors. The same procedure is used in a receive phased array antenna, but in reverse order. See Figure 1.4.

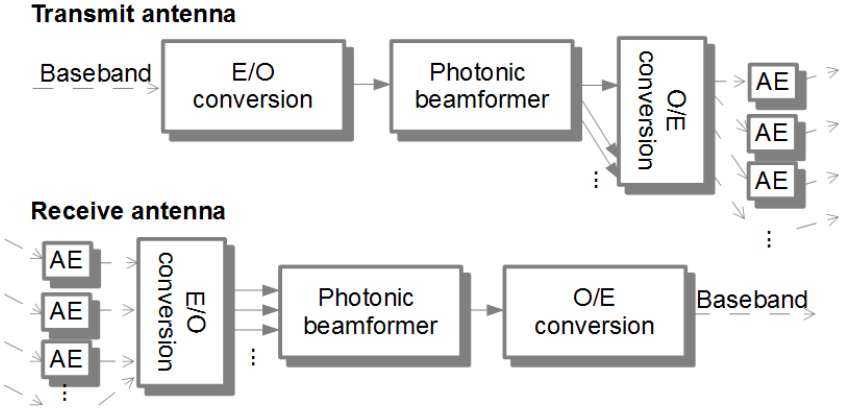


Figure 1.4: Photonic beamforming schematic for a transmit phased array antenna (top) and a receive phased array antenna (bottom). AE stands for antenna element, E/O and O/E stand for electro-optical and opto-electrical respectively. Dashed arrows indicate electrical signals, while solid arrows indicate optical signals.

There exist several ways to do photonic beamforming. With switchable delay lines [6], a set time delay is given to the optical signal, and the correct time delay is achieved by switching to the desired path of a particular length. However, the disadvantage of this method is that only a discrete number of delays can be used, which in turn limits the resolution of the beam angle of the phased array antenna. An alternative solution that does provide continuous tuning is to use a wavelength-tunable laser in combination with a dispersive optical element [7–10]. These methods provide a linear phase response over a large frequency range, but the tunable lasers are relatively expensive and the optical components are bulky.

The most compact photonic beamformer systems make use of integrated optical components [3, 4, 11–14]. These systems are based on integrated all-pass filters that shape the phase response of each path in the photonic beamformer. The used filters can be categorized into infinite impulse response (IIR) filters based on optical resonance techniques [4, 11], and finite impulse response (FIR) filters [12–14]. The IIR filters are realized by optical ring resonators and can provide a linear phase response either over a large bandwidth, or with a large slope (corresponding to a large group delay), but not both. On the other hand, the FIR filters, realized by Mach-Zehnder interferometers, have a better trade-off between bandwidth and maximum group delay [14]. However, so far these have been limited to only one such filter for each path of the photonic beamformer. By putting several optical ring resonators in series for each path of the photonic beamformer, both the bandwidth and the maximum group delay can be increased, at the cost of having a more complex system [4, 15]. It is precisely this last disadvantage that will be tackled in this thesis.

1.1.3. OPTICAL RING RESONATOR-BASED PHOTONIC BEAMFORMING

The photonic beamformer that has been investigated in this work uses optical ring resonators to provide the necessary delays. Optical ring resonators can be used to provide a linear phase response over a large bandwidth for one of the paths in the photonic beamformer [4, 15]. This is best visualized by looking at the group delay response of a beamformer path, which is equal to $-\frac{1}{2\pi}$ times the derivative of the phase response (or just -1 times the derivative if angular frequencies are used). In order to achieve a linear phase response, the group delay response should be a constant function of the frequency. However, the group delay response τ of one optical ring resonator is a nonlinear function of the frequency f, given by:

$$\tau(f) = T\left(\frac{r^2 - rc\cos z}{r^2 + c^2 - 2rc\cos z} + \frac{rc\cos z - r^2c^2}{r^2c^2 + 1 - 2rc\cos z}\right), \tag{1.3}$$

$$c = \sqrt{1-\kappa}, \tag{1.4}$$

$$z = 2\pi f T + \varphi. \tag{1.5}$$

Here, r is a constant related to the loss of the ring, T is the roundtrip time in seconds, and κ and φ are variables that can be adjusted by heater actuators. See Figure 1.5.

In order to achieve an approximately constant group delay response, several ring resonators can be put in series. The total group delay is then the sum of the group delays

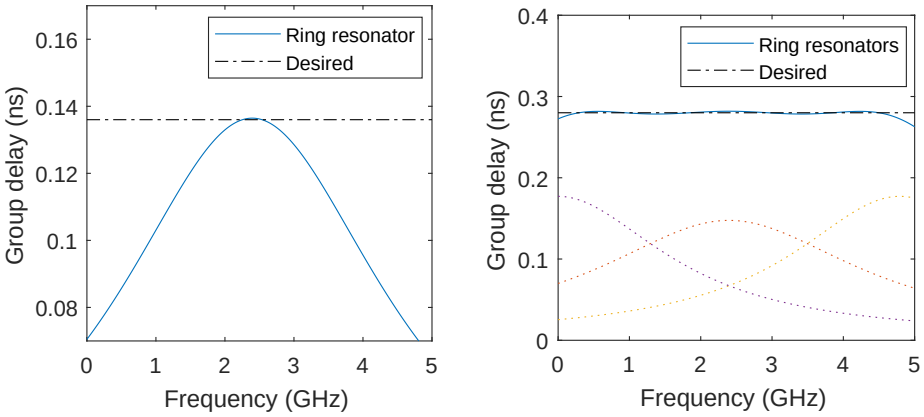


Figure 1.5: (left) Group delay response of one optical ring resonator, and a desired group delay response. (right) Group delay response of a cascade of three optical ring resonators, and a desired group delay response. The individual group delay response of each ring resonator is denoted by the dotted lines.

of each individual ring resonator i:

$$\tau_{\text{total}} = \sum_{i=1}^{n} \tau_i(f), \tag{1.6}$$

with n the total number of ring resonators. See Figure 1.5. Not only does this method achieve a larger bandwidth, also larger delays can be achieved when compared with an individual ring resonator.

The ring resonators need to be tuned correctly in order to achieve the desired group delay response. This is done by adjusting two heater actuators per ring: one that controls the phase φ, and one that controls the coupling constant κ. Changing φ gives a frequency shift to the group delay response, while changing κ changes the width and height of the group delay response. Since a perfectly constant group delay response cannot be achieved, the goal in tuning the optical ring resonators is to minimize the difference between the ideal constant group delay response τ^* and the actual group delay response:

$$\min_{\kappa_1,\ldots,\kappa_n,\varphi_1,\ldots,\varphi_n} \int_{f_{\min}}^{f_{\max}} L\left(\tau_{\text{total}}(f,\kappa_1,\ldots,\kappa_n,\varphi_1,\ldots,\varphi_n) - \tau^*\right) df, \tag{1.7}$$

for a certain bandwidth $[f_{\min}, f_{\max}]$ and for a certain loss function L, such as a quadratic loss function.

1.1.4. CHALLENGES IN PHOTONIC BEAMFORMING

Though the minimization problem (1.7) is a well-defined nonlinear optimization problem, convergence to the global optimum cannot be guaranteed by most standard solvers. This is because the objective function is not a convex function of the variables κ and

ϕ. Nevertheless, good results have been obtained with standard nonlinear optimization solvers in the past [16, 17]. The other standard method of tuning this particular photonic beamformer is a manual tuning method, where the group delay is measured with a vector network analyser, and the heater voltages are tuned by hand one at a time until the desired group delay is achieved [4, 11]. There are however several challenges that prevent both the manual tuning method and the nonlinear optimization method from being used in practice.

HEATER CROSSTALK

First of all, the variables κ and ϕ are controlled by heater actuators. However, there is a quadratic relation between these variables and their heater voltages, which needs to be taken into account [16, Sec. 3.6]. Even more importantly, the heaters influence each other by means of electrical and thermal crosstalk [16, App. B]. This means that adjusting the voltage of one heater does not only result in a change in the corresponding variable κ or ϕ, but also in the variables of other optical ring resonators. In order for the nonlinear optimization method to work, both the quadratic relation and the crosstalk can be compensated for by determining the exact crosstalk effects and all the required heater model parameters, but this is a time-consuming procedure. On the other hand, none of this is necessary for the manual tuning method, but since only one heater is tuned at a time with this method, the tuning process is severely hampered by the crosstalk.

PARAMETER SENSITIVITY

Even when the heater crosstalk is included in the system model, this can only be done up to a certain precision. And not only the relation between the different heater voltages and the variables κ and ϕ needs to be modeled, there are also model parameters for each optical ring resonator such as the round-trip time T and the loss parameter r in (1.1) that need to be modeled accurately. These last two parameters are generally given by the manufacturer of the optical ring resonator, but may be slightly inaccurate due to fabrication errors or material inhomogenieties. While the precision is generally high enough for all practical purposes of the optical ring resonator, these small inaccuracies can have a large influence as they propagate throughout the optimization procedure (1.7).

SCALAR OBJECTIVE AND MEASUREMENT NOISE

Finally, it should be noted that the objective function in (1.7) will most likely not be used in a final application. Typical applications use measures like the signal power or signal-to-noise ratio as the objective, as these can be measured more easily than frequency-dependent objectives. Just like the objective in (1.7), this gives one scalar value for each setting of heater voltages. This is very different from the manual tuning method, where the measured group delay is used, which is a function of the frequency rather than a scalar value. A benefit of the manual tuning method compared to the nonlinear optimization method is that it takes measurement noise into account. By averaging the group delay measurements over time, the noise is reduced, making the tuning process slower (since it takes time to average over several measurements) but more accurate. Measurement noise is not considered in (1.7) since no physical measurements are used.

1.1.5. CRITERIA FOR AN AUTOMATIC TUNING METHOD FOR PHOTONIC BEAMFORMERS

This thesis aims to develop a novel automatic tuning method for optical ring resonator-based photonic beamformers. By looking at the challenges in the previous subsection, we can see that the method should satisfy the following criteria:

- The method should take heater crosstalk into account.
- The method should not be sensitive to model parameters.
- If feedback from measurements will be used, the method should be able to operate with scalar-valued measurements and not be sensitive to measurement noise. Furthermore, the number of measurements used should be as low as possible to prevent the method from being too slow.

Besides these criteria, the method should operate in real time. In this case, this means that the time it takes to find the optimal heater voltages should be within the same order of magnitude as the time it takes to check how well the system is tuned if the heaters are set to those voltages.

There are also criteria that depend on the exact application and beamforming system that is used. An example of such a criterion is that the phase response of the beamformer should be accurate up to 11.25° [11] in the bandwidth of interest.

1.2. DATA-DRIVEN APPROACH TO PHOTONIC BEAMFORMING

Because both the manual tuning method and the nonlinear optimization method above have several drawbacks, a new method is developed in this thesis for the automatic tuning of a ring resonator-based photonic beamformer. Ideally, this new method will have the advantages but not the disadvantages of the manual tuning method and the nonlinear optimization method. Of course, the new method should also overcome the challenges and satisfy the criteria given in the previous section. The advantage of the manual tuning method is that feedback from measurements is used to keep tuning the heater actuators until the system is tuned correctly, but the main disadvantage of this method is that it cannot be used in a real application since it is not automatic. The advantage of the nonlinear optimization method is that an algorithm is used to automatically tune the system, but with no feedback from measurements the algorithm requires a perfect model in order to get good results as will be shown in this thesis.

The core idea in this thesis is to use a so-called surrogate model for the relation between the system parameters (heater voltages) and the performance of the system (e.g., signal power). This surrogate model is continually improved using feedback from measurements instead of just relying on physical models. Nonlinear optimization methods are applied to the surrogate model instead of the original objective to update the heater voltages. This procedure results in a data-driven automatic tuning method.

The scientific literature is full of optimization techniques where a surrogate model is used instead of the original objective [18–20]. This class of optimization algorithms often works better than most other classes of optimization algorithms in this data-driven setting. For example, traditional derivative-based methods like gradient descent or quasi-Newton methods [21] require a derivative of the objective function. If this derivative is

not available, they can approximate the derivatives from the given data points, but this approach is very sensitive to noise in the data. On the other hand, derivative-free methods [22] such as the Nelder-Mead method or genetic algorithms [23] are less susceptible to noise, but require a high number of data points in order to generate good results. Surrogate modeling methods typically also provide estimates for the derivative of the objective function, without the need for additional measurements.They are generally designed in such a way that they are able to deal with noise while not requiring too many data points. However, existing surrogate modeling methods like Bayesian optimization or sequential Kriging optimization [24–27] suffer from one drawback that makes them unfit for the application considered in this thesis: they become slower as the number of data points increases. A real-time automatic tuning method for a photonic beamformer will have to somehow circumvent this drawback.

1.2.1. RELATED RECENT WORK

Besides the methods mentioned in this chapter so far, during this research project a number of relevant studies have emerged independently and simultaneously elsewhere. Since these studies had not yet been published at the time of this project, they have not been taken into account in this thesis. However, a short discussion about these studies is given in Chapter 6.

In [28], a photonic beamformer based on Mach-Zehnder interferometers rather than optical ring resonators was investigated. This system was also automatically tuned using a data-driven approach: the output signal of the system was measured, and the delays of the beamformer were adjusted using a derivative-free optimization algorithm.

In [29], the drawback of Bayesian optimization techniques, namely their computation time becoming slower over time, is overcome by using a combination of random features and Thompson sampling. The method was applied to a materials science application, namely determining the atomic structure of a crystalline interface.

Other techniques that solve the same problem in Bayesian optimization, for example those based on sparsity, either do not solve the problem completely or introduce other disadvantages [30].

1.3. OUTLINE OF THIS THESIS

In this thesis, a data-driven automatic tuning method for photonic beamforming is developed. The method is compared with the two methods mentioned in this chapter, namely the manual tuning method and the nonlinear optimization method, as well as with state-of-the-art surrogate modeling methods. The latter comparison is made not just for the application of photonic beamforming, but also for other applications.

CHAPTER 2

In this chapter, first principles modeling is compared to surrogate modeling on a simulation of a photonic beamformer. The former uses the nonlinear optimization procedure as explained in this introduction. The latter uses the proposed data-driven procedure, where the relation between the κ and ϕ variables from Sec. 1.1.3 and the mean square error between the group delay response and the desired delay is approximated. This function approximation is done using a surrogate model. A nonlinear solver is then used on

this surrogate model to find the κ and ϕ variables that minimize the mean square error.

This chapter is based on the following publication:

L. Bliek, M. Verhaegen and S. Wahls, *Data-driven Minimization with Random Feature Expansions for Optical Beam Forming Network Tuning*, 16th IFAC Workshop on Control Applications of Optimization (CAO'2015) **48**, 166 (2015).

CHAPTER 3
In this chapter, the data-driven approach from Chapter 2 is adapted to develop an online optimization algorithm. The surrogate model used in this algorithm is updated every time a new measurement becomes available. This makes it possible to converge towards the minimum of the original objective function: the mean square error between the group delay response and the desired delay. The algorithm is applied to a simulation of a photonic beamformer like the one described in this introduction, to a toy example, and to two different applications: optical coherence tomography and robot arm control. The algorithm is compared to similar state-of-the-art surrogate modeling algorithms, and theoretical results are given that provide insight in how to configure the algorithm in practice.

This chapter is based on a joint work with H.R.G.W. Verstraete, with an equal contribution from both parties, and also appears in:

H.R.G.W. Verstraete, *Optimization-based adaptive optics for optical coherence tomography*, Ph.D. thesis, Delft University of Technology (2017).

This chapter is based on the following publication:

L. Bliek, H. R. G. W. Verstraete, M. Verhaegen and S. Wahls, *Online Optimization With Costly and Noisy Measurements Using Random Fourier Expansions*, IEEE Transactions on Neural Networks and Learning Systems **29**, 167 (2018).

In this publication, H.R.G.W. Verstraete focused more on the programming and practical use and on the OCT application, while L. Bliek focused more on the theorems and proofs and on the beamforming application.

CHAPTER 4
In this chapter, the algorithm from Chapter 3 is applied to the photonic beamformer described in this introduction, not just on a simulation of this system. The beamformer is described as being part of a phased array transmit antenna that is fully integrated on a chip. The purpose of the described system is to provide broadband Internet connections on board an aircraft, using the K_u band.

This chapter is based on the following publication:

L. Bliek, S. Wahls, I. Visscher, C. Taddei, R. B. Timens, R. Oldenbeuving, C. Roeloffzen, M. Verhaegen, *Automatic Tuning of a Novel Ring Resonator-based Photonic Beamformer for a Transmit Phased Array Antenna*, arXiv e-prints, arXiv:1808.04814 (2018).

CHAPTER 5

In this chapter, the algorithm from Chapter 3 is adapted in such a way that the surrogate model used to approximate the original objective becomes convex and sparse. This makes it possible to get fast implementations of the algorithm, and to use convex optimization solvers on the surrogate model. Different adaptations of the same algorithm are applied to a toy example, to the problem of hyper-parameter optimization for handwritten digit classification using deep learning, and to a simulation of a photonic beamformer.

This chapter is based on the following publications:

L. Bliek, M. Verhaegen, and S. Wahls, *Online function minimization with convex random relu expansions*, 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), 2017.

APPENDIX

The appendix describes two adaptations of the algorithm from Chapter 3. One is to use a sliding window on the measurements used by the algorithm, where only the most recent measurements are used to fit the surrogate model. This makes it possible to use the algorithm on applications where the objective function changes over time, which is the case in aircraft-satellite communication. The other adaptation exploits the fact that most of the applications considered in this thesis have objective functions with a convex or pseudoconvex shape. This is done by adding a variable offset to the surrogate model. These two adaptations are applied to confocal fluorescent microscopy and compared to a hill climbing algorithm. The adaptations have also been used in Chapter 4.

REFERENCES

[1] H. Schippers, J. Verpoorte, P. Jorna, A. Hulzinga, A. Meijerink, C. Roeloffzen, L. Zhuang, D. Marpaung, W. van Etten, R. Heideman, *et al.*, *Broadband conformal phased array with optical beam forming for airborne satellite communication,* in *Aerospace Conference, 2008 IEEE* (IEEE, 2008) pp. 1–17.

[2] J. Capmany and D. Novak, *Microwave photonics combines two worlds,* Nature Photonics **1**, 319 (2007).

[3] D. Marpaung, C. Roeloffzen, R. Heideman, A. Leinse, S. Sales, and J. Capmany, *Integrated microwave photonics,* Laser & Photonics Reviews **7**, 506 (2013).

[4] A. Meijerink, C. G. Roeloffzen, R. Meijerink, L. Zhuang, D. A. Marpaung, M. J. Bentum, M. Burla, J. Verpoorte, P. Jorna, A. Hulzinga, *et al.*, *Novel ring resonator-based integrated photonic beamformer for broadband phased array receive antennas—part I: Design and performance analysis,* J. Lightwave Technol. **28**, 3 (2010).

[5] R. Rotman, M. Tur, and L. Yaron, *True time delay in phased arrays,* Proceedings of the IEEE **104**, 504 (2016).

[6] M. A. Piqueras, G. Grosskopf, B. Vidal, J. Herrera, J. M. Martínez, P. Sanchis, V. Polo, J. L. Corral, A. Marceaux, J. Galière, *et al.*, *Optically beamformed beam-switched adaptive antennas for fixed and mobile broad-band wireless access networks,* IEEE Transactions on Microwave Theory and Techniques **54**, 887 (2006).

[7] H. Zmuda, R. A. Soref, P. Payson, S. Johns, and E. N. Toughlian, *Photonic beamformer for phased array antennas using a fiber grating prism,* IEEE Photonics Technology Letters **9**, 241 (1997).

[8] J. Corral, J. Marti, J. Fuster, and R. Laming, *Dispersion-induced bandwidth limitation of variable true time delay lines based on linearly chirped fibre gratings,* Electronics Letters **34**, 209 (1998).

[9] B. Ortega, J. L. Cruz, J. Capmany, M. V. Andrés, and D. Pastor, *Variable delay line for phased-array antenna based on a chirped fiber grating,* IEEE Transactions on Microwave Theory and Techniques **48**, 1352 (2000).

[10] D. B. Hunter, M. E. Parker, and J. L. Dexter, *Demonstration of a continuously variable true-time delay beamformer using a multichannel chirped fiber grating,* IEEE Transactions on Microwave Theory and Techniques **54**, 861 (2006).

[11] L. Zhuang, C. G. Roeloffzen, A. Meijerink, M. Burla, D. A. Marpaung, A. Leinse, M. Hoekman, R. G. Heideman, and W. van Etten, *Novel ring resonator-based integrated photonic beamformer for broadband phased array receive antennas—part II: Experimental prototype,* Journal of lightwave technology **28**, 19 (2010).

[12] V. C. Duarte, M. V. Drummond, and R. N. Nogueira, *Photonic true-time delay beamforming system for a phased array antenna receiver,* in *2015 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)* (2015) pp. 1–5.

[13] V. C. Duarte, M. V. Drummond, and R. N. Nogueira, *Photonic true-time-delay beamformer for a phased array antenna receiver based on self-heterodyne detection,* Journal of Lightwave Technology **34**, 5566 (2016).

[14] D. Melati, A. Waqas, Z. Mushtaq, and A. Melloni, *Wideband integrated optical delay line based on a continuously tunable Mach-Zehnder interferometer,* IEEE Journal of Selected Topics in Quantum Electronics **24**, 1 (2018).

[15] C. Roeloffzen, L. Zhuang, R. Heideman, A. Borreman, and v. W. Etten, *Ring resonator-based tunable optical delay line in LPCVD waveguide technology,* in *Proceedings Symposium IEEE/LEOS Benelux Chapter* (IEEE, 2005) pp. 79–82.

[16] L. Zhuang, *Ring resonator-based broadband photonic beam former for phased array antennas,* Ph.D. thesis, University of Twente (2010).

[17] R. Blokpoel, A. Meijerink, L. Zhuang, C. Roeloffzen, and W. van Etten, *Staggered delay tuning algorithms for ring resonators in optical beam forming networks,* in *Proc. 12th IEEE/LEOS Symp. Benelux* (2007) pp. 243–246.

[18] A. I. Forrester and A. J. Keane, *Recent advances in surrogate-based optimization,* Progress in Aerospace Sciences **45**, 50 (2009).

[19] S. Koziel, D. E. Ciaurri, and L. Leifsson, *Surrogate-based methods,* in *Computational optimization, methods and algorithms* (Springer, 2011) pp. 33–59.

[20] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, *Taking the human out of the loop: A review of bayesian optimization,* Proceedings of the IEEE **104**, 148 (2016).

[21] J. Nocedal and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

[22] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, Vol. 8 (Siam, 2009).

[23] L. Davis, *Handbook of genetic algorithms* (CUMINCAD, 1991).

[24] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient global optimization of expensive black-box functions,* J. Global Optim. **13**, 455 (1998).

[25] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, *Algorithms for hyper-parameter optimization,* in *Adv. Neur. In.* (2011) pp. 2546–2554.

[26] J. Snoek, H. Larochelle, and R. P. Adams, *Practical Bayesian optimization of machine learning algorithms,* in *Adv. Neur. In.* (2012) pp. 2951–2959.

[27] R. Martinez-Cantin, *BayesOpt: a Bayesian optimization library for nonlinear optimization, experimental design and bandits,* J. Mach. Learn. Res. **15**, 3735 (2014).

[28] V. C. Duarte, M. V. Drummond, and R. N. Nogueira, *Coherent photonic true-time-delay beamforming system for a phased array antenna receiver,* in *2016 18th International Conference on Transparent Optical Networks (ICTON)* (2016) pp. 1–5.

[29] T. Ueno, T. D. Rhone, Z. Hou, T. Mizoguchi, and K. Tsuda, *Combo: An efficient bayesian optimization library for materials science,* Materials Discovery **4**, 18 (2016).

[30] A. Singh, N. Ahuja, and P. Moulin, *Online learning with kernels: Overcoming the growing sum problem,* in *2012 IEEE International Workshop on Machine Learning for Signal Processing* (IEEE, 2012) pp. 1–6.

2

Data-driven Minimization with Random Feature Expansions for Optical Beam Forming Network Tuning

This paper proposes a data-driven method to minimize objective functions which can be measured in practice but are difficult to model. In the proposed method, the objective is learned directly from training data using random feature expansions. On the theoretical side, it is shown that the learned objective does not suffer from artificial local minima far away from the minima of the true objective if the random basis expansions are fit well enough in the uniform sense. The method is also tested on a real-life application, the tuning of an optical beamforming network. It is found that, in the presence of small model errors, the proposed method outperforms the classical approach of modeling from first principles and then estimating the model parameters.

Parts of this chapter have been published in [1].

2.1. INTRODUCTION

THE control community can roughly be divided in two groups: a model-based group and a data-based group [2]. The former takes the classical approach of building a model from first principles, estimating system parameters from data, followed by control design or the minimization of some objective. The data-driven approach skips these first steps and immediately utilizes data for some control or optimization objective. While model-based control can be a powerful tool, some problems are difficult to model. In such cases, it can become very difficult to take model errors and uncertainties into account [3]. In the data-driven approach, control design or objective minimization is done directly after gathering data, using black-box models instead of first principles. This approach is beneficial when no first principles are available or when a system is too complex to be modeled accurately. However, data-based techniques can also be beneficial when there is a model, but some parts of the model are uncertain or unknown.

The core idea of the method proposed in this paper is to directly measure the objective that is to be minimized, instead of estimating a system model which is then plugged into an objective. The objective is approximated with random feature expansions (RFEs) [4], and this approximation of the objective is then minimized. Fast algorithms for function approximation using RFEs exist. Their strength lies in the simplicity of the algorithms: training is done with a single linear regression step, even though the approximation can still be nonlinear.

Approximating an unknown function and then minimizing this approximation, however, could be troublesome if the approximation contains artificial local minima that were not present in the true objective function. This paper shows that, with high probability, the local minima of the approximation with RFEs lie close to the local minima of the true objective function if the objective is approximated well enough.

Besides this theoretical result, the method is tested in a real-life application, the tuning of an optical beamforming network (OBFN). OBFNs are used to process signals from different antenna elements in such a way that they add up in phase, resulting in direction-sensitive signal reception [5]. Actuators on the OBFN can be used to control the signal delays. If the desired delay is known, the problem of tuning the OBFN can be written as an optimization problem [5, Appendix A]. The objective to be minimized is the difference between the delay provided by the OBFN and the desired delay. Since accurate (but complex) models are available for this problem, a model-based approach can be used to solve it. However, this paper will show that very small uncertainties in the model can have a large detrimental effect on the objective minimization, while the proposed data-based method circumvents this.

RFEs and the proposed method are explained in more detail in Section 2.2. Section 2.3 investigates whether the approximation with RFEs is fit for optimization by providing a theorem about the local minima of this approximation. Section 2.4 provides more details about the OBFN tuning problem, how the proposed method is used in this application and compared with other methods, as well as simulation the results. Conclusions are presented in Section 2.5.

2.2. RANDOM FEATURE EXPANSIONS

Many nonlinear systems can be modeled by a combination of nonlinear and linear subsystems, and several identification algorithms for such systems are available [6]. In machine learning, these subsystems are often static, and several methods for function approximation are available. As an example of a static linear subsystem that follows a nonlinearity, consider the output weights c_k in a multilayer perceptron with linear output neurons

$$\hat{f}(\mathbf{x}) = \sum_{k=1}^{D} c_k \Phi(\mathbf{w}_k^T \mathbf{x} + b_k). \tag{2.1}$$

Here, $\hat{f}$ denotes the neural network, $\mathbf{x} \in \mathbb{R}^l$ is the input, D is the number of hidden neurons, Φ is a nonlinear function like a sigmoid or a Gaussian, and the other parameters are weights. The linear weights found in kernel expansions

$$\hat{f}(\mathbf{x}) = \sum_{k=1}^{N} c_k \Phi(\mathbf{x}, \mathbf{x}_k), \tag{2.2}$$

with N the number of training samples, are another example. The weights in a multilayer perceptron are usually trained with some kind of gradient descent algorithm [7]. For kernel machines, convex optimization techniques are often used [8], but the storage and computation costs can become high when the number of training samples becomes large.

Recently, both machine learning fields (neural networks and kernel methods) have started to investigate a technique that had been used mainly as a heuristic before more thoroughly: the use of random features [4, 9, 10]. For neural networks, this technique can be interpreted as randomly initializing the weights $\mathbf{w}_k$ and biases b_k, after which the training of c_k becomes a linear least squares problem [11, 12]. For kernel methods, this can be interpreted as approximating the kernel with an inner product of randomized feature mappings [4]. No matter the interpretation, in this paper a RFE will be denoted as

$$\hat{f}(\mathbf{x}) := \sum_{k=1}^{D} c_k \Phi(\mathbf{w}_k^T \mathbf{x} + b_k) = \mathbf{c}^T \Phi(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{2.3}$$

with $\mathbf{W} \in \mathbb{R}^{D \times l}$ and $\mathbf{b} \in \mathbb{R}^{D \times 1}$ being fixed matrices drawn from suitably chosen continuous probability distributions, $\Phi : \mathbb{R}^l \to \mathbb{R}^n$ a bounded non-constant piece-wise continuous function (e.g. a sigmoid or sinusoid) that operates element-wise on a vector , $\mathbf{c} \in \mathbb{R}^D$ a vector of linear coefficients, and D the number of random features.

Although random features have been used mostly because of their practical value, more and more theoretical results are becoming available [12–14]. These results show that random features can be used to approximate any continuous function with high accuracy, without the need for a kernel trick or nonlinear optimization.

Suppose that the target function f has been sampled at randomly chosen locations $\mathbf{x}_1, \ldots \mathbf{x}_N \in [-1, 1]^l$. The corresponding noisy samples of f are denoted by

$$y_n = f(\mathbf{x}_n) + \varepsilon_n, \tag{2.4}$$

where the ε_n are, for example, realizations of white Gaussian noise. Now the function f can be fitted by solving the linear least squares problem

$$\min_{\mathbf{c}} ||\mathbf{y} - \mathbf{G}\mathbf{c}||^2 + \lambda ||\mathbf{c}||^2 \tag{2.5}$$

with $\mathbf{y} \in \mathbb{R}^N$ being the vector of samples y_n and

$$\mathbf{G} = [\Phi(\mathbf{W}\mathbf{x}_1 + \mathbf{b}) \cdots \Phi(\mathbf{W}\mathbf{x}_N + \mathbf{b})]^T. \tag{2.6}$$

The regularization parameter $\lambda > 0$ helps to avoid overfitting of the model to the data, which would impair its performance on new, previously unseen inputs, and to ensure that there is a unique solution to (2.5) . This problem has the following solution [15]:

$$\hat{\mathbf{c}} = (\mathbf{G}^T\mathbf{G} + \lambda I)^{-1}\mathbf{G}^T, \tag{2.7}$$

which leads to a direct method for fitting f with RFEs.

2.3. THEORETICAL RESULTS

After computing (2.7), the RFE model (2.3) can be used efficiently as an approximation of the target function f. However, this does not necessarily mean that it is a good surrogate for f when performing optimization. We need to investigate whether the extreme points of $\hat{f}$ are close to the extreme points of f. To show that this is not trivial, Figure 2.1 shows an approximation that increases in accuracy, but introduces many artificial extreme points.

The main result of this paper comes in the form of a theorem that claims that the extreme points of $\hat{f}$ are, with high probability, close to the extreme points of f if $\hat{f}$ is a good enough approximation of f in the uniform sense. The result is theoretical in the sense that although we do know that such an approximation exists, we have no guarantee that the method from the previous section finds it.

In this section, the weights $\mathbf{w}_k$ of the random basis expansion defined in (2.3) are assumed to be i.i.d. normally distributed, $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, while the weights b_k are assumed to be i.i.d. uniform on $[0, 2\pi]$. The nonlinearity Φ is assumed to be the cosine function, which gives the RFE the interpretation of an approximated Gaussian kernel [4].

The following is a summary of well-known results from the literature:

Corollary 1. *Assume that f is continuous and fix any $\delta \in (0,1)$ and $\varepsilon > 0$. Then, there exists a constant $D_0 = D_0(f, \delta, \varepsilon)$ such that, for any $D \geq D_0$ and randomly chosen i.i.d. weights $\mathbf{w}_1, \ldots, \mathbf{w}_D$ and biases $b_1, \ldots, b_D$,*

$$\mathscr{C} := \left\{ \mathbf{c} \in \mathbb{R}^D : \|f - \hat{f}\|_\infty \leq \varepsilon, \sup_k |c_k| \leq \gamma(f)/D \right\} \neq \emptyset \tag{2.8}$$

with probability at least $1 - \delta$. Here, $\hat{f}$ is as in (2.3), $\gamma(f)$ denotes a constant that depends only on f, and

$$\|\cdot\|_\infty := \sup_{\mathbf{x} \in \mathscr{X}} |\cdot(\mathbf{x})|. \tag{2.9}$$

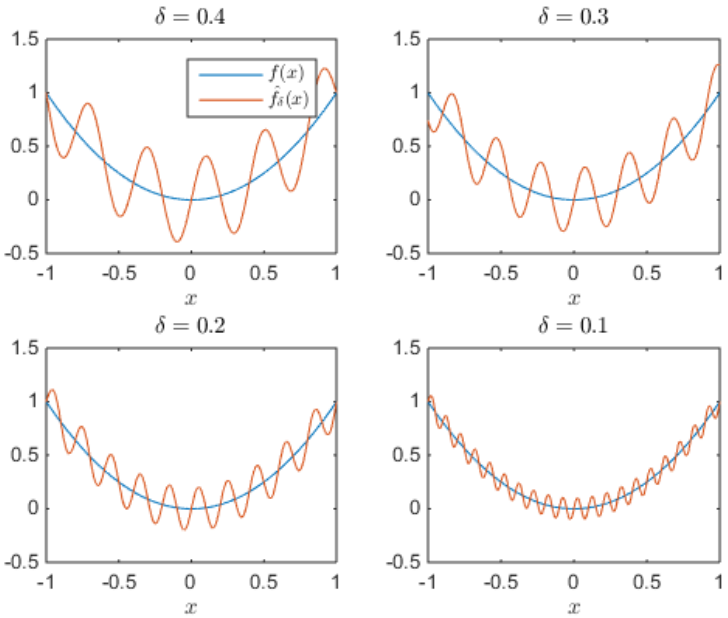


Figure 2.1: The functions $\hat{f}_\delta$ provide a better and better approximation of f for $\delta \to 0$ since $\|f - \hat{f}_\delta\|_\infty \leq \delta$, but they suffer from artificial extreme points that are distant from any extreme point of f.

This result shows that with high probability there exists a vector of weights $\mathbf{c}$ such that the RFE $\hat{f}$ approximates f up to arbitrary precision, as long as the number of features is large enough. It does not guarantee that the least squares approach of Section 2.2 results in exactly these weights. This corollary will be proved in the appendix.

Our main result is the following theorem.

Theorem 2. *Let $f : \mathscr{X} \to \mathbb{R}$, $\mathscr{X} \subset \mathbb{R}^d$ compact, be two times differentiable with continuous second derivative. Furthermore, assume that f has only finitely many critical points*

$$\{\mathbf{v}_1, \dots, \mathbf{v}_K\} := \{\mathbf{x} \in \mathscr{X} : \nabla f(\mathbf{x}) = \mathbf{0}\}$$

in $\mathscr{X}$. Then, for any $\delta \in (0,1)$ and $\varepsilon > 0$, there exists a constant $D_0 = D_0(f, \delta, \varepsilon)$ such that any random basis expansion $\hat{f}$ defined in (2.3) with coefficient vector $\mathbf{c} \in \mathscr{C}$ (see (2.8)) satisfies

$$\nabla \hat{f}(\mathbf{x}) = \mathbf{0} \quad \Longrightarrow \quad \min_{k=1,\dots,K} \|\mathbf{v}_k - \mathbf{x}\| \leq \varepsilon$$

with a probability of at least $1 - \delta$ whenever $D \geq D_0$.

The proof is given in the appendix.

2.4. APPLICATION: TUNING OF AN OPTICAL BEAM FORMING NETWORK

As a real-life application, we consider the tuning of an optical beam-forming network (OBFN) architecture proposed by [16] for applications such as aircraft-satellite communication. OBFNs are used in phased arrays, where several antenna elements are placed in an array. All antenna elements receive the same signal, but with different time delays as illustrated in Figure 2.2. The time delays between the different received signals can be calculated if the shape of the phased array and the reception angle of the incoming signal are known, as is the case in aircraft-satellite communication. OBFNs aim at improving the signal-to-noise ratio of the incoming signal. Therefore, the received signals are first aligned through proper compensation of their individual delays and then combined. OBFNs convert the incoming electric signals into the optical domain and process them using optical ring resonators, which offers several advantages such as compactness and low weight, low loss, and large bandwidth [17].

The main components of the OBFNs are optical ring resonators (ORRs) [18]. ORRs can provide a tunable time delay to signals, but only over a small frequency band. Cascades of multiple ORRs can provide a constant delay over larger bandwidths [19], but it was found that the number of required ORRs can be reduced if the ORRs are organized in tree topologies such as the one depicted in Figure 2.3 [5, Chapter 3]. In the OBFN under consideration, ORRs are combined in a binary tree topology, as illustrated in Figure 2.3, providing different constant delays for each path in the tree over a large bandwidth.

The group delay τ_i of the i-th ORR depends on the frequency ω as follows (modified from [5, p. 22]):

$$\tau_i(\omega, \kappa_i, \phi_i) = T_i \frac{r_i^2 - r_i\sqrt{1-\kappa_i}\cos(\omega T_i + \phi_i)}{r_i^2 + 1 - \kappa_i - 2r_i\sqrt{1-\kappa_i}\cos(\omega T_i + \phi_i)}$$

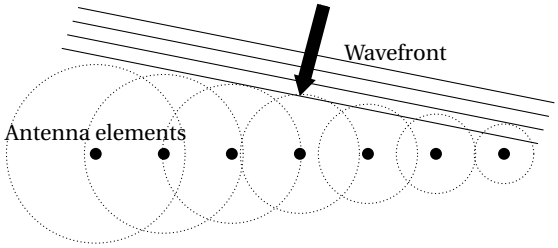


Figure 2.2: A phased array antenna. If a wave arrives at the array under an angle, each antenna element receives the same signal after a certain time delay that can be calculated if the distance between antenna elements is known.

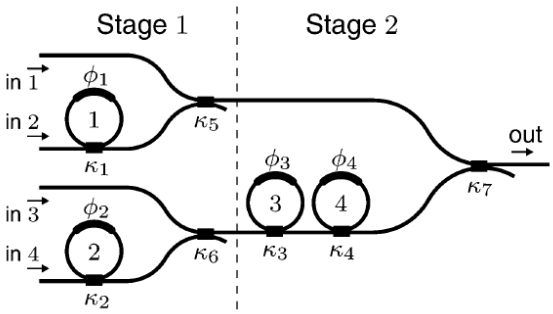


Figure 2.3: Binary tree-based 4×1 optical beamforming network (OBFN) consisting of four optical ring resonators (ORRs), from [16].

$$+ T_i \frac{r_i\sqrt{1-\kappa_i}\cos(\omega T_i + \phi_i) - r_i^2(1-\kappa_i)}{r_i^2(1-\kappa_i) + 1 - 2r_i\sqrt{1-\kappa_i}\cos(\omega T_i + \phi_i)}. \tag{2.10}$$

Here, κ_i and ϕ_i are a coupling and phase shift variable, which can be controlled with chromium heaters, and

$$r_i = \bar{r} + \Delta r_i, \; T_i = \bar{T} + \Delta T_i \tag{2.11}$$

are the loss parameter and the round-trip time of the i-th ORR respectively, centered around their averages $\bar{r}$ and $\bar{T}$. The (small) deviations Δr_i and ΔT_i are caused by fabrication errors and material inhomogeneities, and are unknown in practice.

The group delay d_j of the path connecting the j-th antenna element to the output is given by the sum of the group delays of all the ORRs in the path [19]:

$$d_j(\omega, \boldsymbol{\kappa}, \boldsymbol{\phi}) = \sum_{i=1}^{R} p_{ij}\tau_i(\omega, \kappa_i, \phi_i), \tag{2.12}$$

where $\boldsymbol{\kappa}, \boldsymbol{\phi}$ are vectors containing the κ_i and ϕ_i for the i-th ORR, R is the total number of ORRs in the OBFN, and $p_{ij} \in \{0,1\}$ indicates whether the i-th ORR appears in the j-th path (1) or not (0).

The goal is to find the values for κ_i and ϕ_i that provide the desired delays d_j^* over a set of target frequencies $\omega_1,\ldots,\omega_L$ for all OBFN paths $j=1,\ldots,P$. Since this problem has no exact solution in general, we aim at minimizing the mean-square error

$$\mathrm{MSE}(\boldsymbol{\kappa},\boldsymbol{\phi}) := \frac{1}{LP}\sum_{k=1}^{L}\sum_{j=1}^{P}\left(d_j^* - d_j(\omega_k,\boldsymbol{\kappa},\boldsymbol{\phi})\right)^2 \tag{2.13}$$

instead, where k sums over the frequencies of interest, and L is the number of frequencies considered. Although this is a non-convex problem, good results have been obtained when the mean-square error was minimized with standard black-box nonlinear optimization techniques [5, Appendix A]. However, since the exact values of the parameters (2.11) are unknown in practice, [5] assumed that

$$r_1 = \ldots = r_R = \bar{r},\ T_1 = \ldots = T_R = \bar{T}. \tag{2.14}$$

In our notation, this corresponds to the minimization of the objective function

$$\overline{\mathrm{MSE}}(\boldsymbol{\kappa},\boldsymbol{\phi}) := \frac{1}{LP}\sum_{k=1}^{L}\sum_{j=1}^{P}\left(d_j^* - \bar{d}_j(\omega_k,\boldsymbol{\kappa},\boldsymbol{\phi})\right)^2, \tag{2.15}$$

where $\bar{d}_j$ is given by (2.12) with $r_i=\bar{r}$ and $T_i=\bar{T}$ for all i. In this section it will however become clear that, even if the r_i and T_i deviate only slightly from their average values $\bar{r}$ and $\bar{T}$, this can have a large effect on the outcome of the optimization. Although parameter estimation techniques could be used to estimate these perturbations, model errors can never be eliminated completely. Therefore, this paper proposes to use the method from Section 2.2 as an alternative. This leads to a third objective function that is learned directly from training data:

$$\widehat{\mathrm{MSE}}(\boldsymbol{\kappa},\boldsymbol{\phi}) := \hat{f}(\boldsymbol{\kappa},\boldsymbol{\phi}) = \mathbf{c}\Phi\left(\mathbf{W}\begin{bmatrix}\boldsymbol{\kappa}\\ \boldsymbol{\phi}\end{bmatrix}+\mathbf{b}\right), \tag{2.16}$$

where $\mathbf{c}$, $\mathbf{W}$ and $\mathbf{b}$ are obtained by the procedure described in Section 2.2. That is, the function $\mathrm{MSE}(\boldsymbol{\kappa},\boldsymbol{\phi})$ is seen as an unknown target function that we want to approximate. Random values for $\mathbf{x}=[\boldsymbol{\kappa}^T,\boldsymbol{\phi}^T]^T$ are chosen as the input samples. Then the path group delays $\hat{d}_j$ are calculated for each $\mathbf{x}$ using (2.12), but disturbed with white Gaussian measurement noise with variance σ^2. Using these disturbed $\hat{d}_j$ in (2.13) gives noisy measurement samples y_n that can be used for finding $\mathbf{c}$. The $\mathbf{W}$ and $\mathbf{b}$ are not chosen in an optimal way, but randomly as described in Section 2.3.

In order to compare our approach with [5], the `fmincon` function from MATLAB® was used to minimize all three objective functions (2.13)-(2.16). The same box constraints for the variables $\boldsymbol{\kappa}$ and $\boldsymbol{\phi}$ were used. The number of training samples was chosen as $N=1024$, the variance of the measurement noise was chosen at $\sigma^2=1$, the basis function $\Phi(x)=\cos(x)$ was used, and the variance $\sigma_{\mathbf{W}}^2$ of the elements in $\mathbf{W}$, the Tikhonov regularization parameter λ, and the number of basis fuctions D were chosen using random hyperparameter optimization [20].

The perturbations Δr_i were chosen randomly from a uniform distribution over $[-\frac{1}{2}\sigma_{\Delta r},\frac{1}{2}\sigma_{\Delta r}]$, with 7 different interval lengths $\sigma_{\Delta r}=10^{-7},\ldots,10^{-1}$. With this scheme

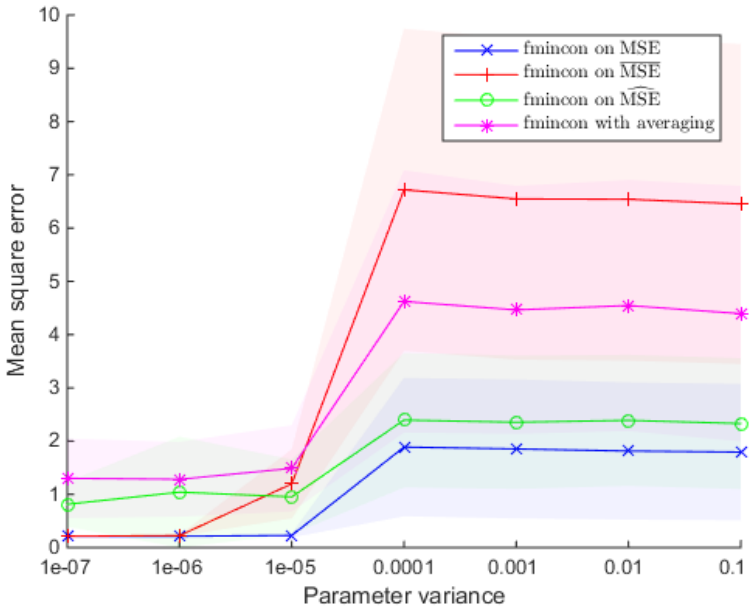


Figure 2.4: Mean square errors for the OBFN delay tuning problem with four different methods: optimization based on the true model (MSE), based on a model with averaged parameters ($\overline{\text{MSE}}$), where the accuracy of the estimates is determined by the parameter perturbations, and based on the learned objective ($\widehat{\text{MSE}}$). The fourth model uses a finite difference approach on averaged measurements. For each parameter perturbation level, the methods were repeated 1000 times, with the mean shown in the graphs and the standard deviation shown as shaded areas in the figure.

and with the estimate $\bar{r} = 0.95$, the loss parameter r_i would never go above 1, which is physically impossible for a passive OBFN system. The perturbations ΔT_i were chosen randomly from a uniform distribution over $[-\frac{1}{2}\sigma_{\Delta T}, \frac{1}{2}\sigma_{\Delta T}]$ with a varying interval length of $\sigma_{\Delta T} = 10^{-10}\sigma_{\Delta r}$, since the estimate of $\bar{T} = 1.38 \cdot 10^{-10}$ is about 10 orders of magnitude smaller than $\bar{r}$.

Figure 2.4 shows the results for minimizing the three objective functions with increasing parameter perturbations, averaged over 1000 runs. The standard deviation of the mean square errors is indicated by the shaded areas. A fourth curve shows the results of a benchmark method, where measurements are first averaged to reduce measurement noise and then minimized with a finite difference approach, using the same measurement noise with $\sigma^2 = 1$ and number of measurements $N = 1024$.

It can be seen that for parameter perturbations close to 0, the minimization of the learned error $\widehat{\text{MSE}}$ gives worse results than the minimization of $\overline{\text{MSE}}$. It also gives a larger standard deviation, showing the random nature of the method. However, as the parameter disturbance increases, the quality of the solution of minimizing $\overline{\text{MSE}}$ decreases, while the minimization of $\widehat{\text{MSE}}$ still gives results that are comparable to MSE. This change happens quickly, when the parameter disturbances are still quite small (around a variation

of 10^{-5} for r_i and 10^{-15} for T_i).

2.5. CONCLUSION

In this paper, it was proposed to use random basis expansions for the minimization of unknown objectives. Instead of deriving a model from first principles, estimating model parameters, and minimizing some objective derived from this model, the proposed method learns the objective function to be minimized directly from data. Random basis expansions were used to approximate the objective function, and it was shown that this approximation does not suffer from artificial local minima if trained ideally. The method was tested on a real life application, namely the tuning of an optical beamforming network. In the presence of model uncertainties, the proposed method outperforms the classical approach.

2.6. APPENDIX: AUXILIARY RESULTS

We start by proving Corollary 2.8, followed by some auxiliary results, followed by the proof of Theorem 2.

Proof. (Corollary 1)
Since f is continuous, we can approximate it by $\tilde{f}(\mathbf{x}) = \sum_{k=1}^{\infty} \beta_k \exp(-\sigma\|\mathbf{x}-\mathbf{x}_k\|^2)$ with $\sum_{k=1}^{\infty} |\beta_k|^2 < \infty$ such that $\|f-\tilde{f}\|_\infty < \frac{\varepsilon}{3}$ [21, Exa. 1]. Note that $\tilde{f}$ belongs to the reproducing kernel Hilbert space (RKHS) generated by the Gaussian kernel. By [14, Thm. 4.2], we can find $\check{f} \in \mathscr{F}$, where $\mathscr{F}$ is a certain dense subset of this RKHS, such that $\|\tilde{f}-\check{f}\|_\infty < \frac{\varepsilon}{3}$. Finally, by [14, Thm. 3.2], there exists an expansion (2.3) with $\|\check{f}-\hat{f}\|_\infty < \infty$ with a coefficient vector belonging to $\mathscr{C}$ with probability at least $1-\delta$ whenever $D \geq D_0$ for a suitably chosen D_0. The triangle inequality now shows that

$$\|f-\hat{f}\|_\infty \leq \|f-\tilde{f}\|_\infty + \|\tilde{f}-\check{f}\|_\infty + \|\check{f}-\hat{f}\|_\infty \leq \varepsilon.$$

The coefficients used in the proof of [14, Thm. 3.2] satisfy $\sup_k |c_k| < \gamma(f)/D$ because, in the notation of [14, proof of Thm. 3.2], $\gamma(f) = \|\check{f}\|_p < \infty$. □

Lemma 3. *Let $\delta \in (0,1)$ and let $\hat{f}$ be an approximation of f with coefficient vector $\mathbf{c} \in \mathscr{C}$ (see (2.8)). Then there exist $D_0 = D_0(f,\delta)$ and a constant $M < \infty$ such that $\left\| \frac{\partial^2 \hat{f}}{\partial x_i^2} \right\|_\infty < M$ with probability at least $1-\delta$ whenever $D \geq D_0$, for all i.*

Proof. Note that for all $D \geq D_0$ with probability $1-\delta$

$$\left\| \frac{\partial^2 \hat{f}}{\partial x_i^2} \right\|_\infty = \sup_{\mathbf{x}\in\mathscr{X}} \left| -\sum_{k=1}^{D} c_k \cos(\mathbf{w}_k^T\mathbf{x}+b_k)\mathbf{w}_{k,i}^2 \right| \leq \frac{\gamma(f)}{D} \sum_{k=1}^{D} \mathbf{w}_{k,i}^2,$$

if the coefficient vector $\mathbf{c}$ belongs to the set $\mathscr{C}$ in 2.8. Here we have used that $\|\cos\|_\infty \leq 1$ and Corollary 1. The term $\sum_{k=1}^{D} \mathbf{w}_{k,i}^2$ can be written as $\sigma^2 \sum_{k=1}^{D} z_k^2 = \sigma^2 X$, with σ^2 the variance of $\mathbf{w}_{k,i}$ and the z_k being standard normally distributed variables. Now, X is a $\chi^2(D)$-distributed random variable which has the following Chernoff bound [22, p. 3f]:

$$\mathbb{P}(X \geq M) \leq e^{-tM}(1-2t)^{-D/2}, \quad \forall t \in (0,1/2),$$

where t is free to be chosen. Choosing for example $t = 1/4$ in (2.18) below gives the following bound:

$$\mathbb{P}\left(\left\|\frac{\partial^2 \hat{f}}{\partial x_i^2}\right\|_\infty \geq M\right) \leq \mathbb{P}\left(\frac{\gamma(f)}{D}\sigma^2 X \geq M\right) \tag{2.17}$$

$$= \mathbb{P}\left(X \geq \frac{DM}{\sigma^2\gamma(f)}\right) \leq e^{-\frac{DM}{4\sigma^2\gamma(f)}}\left(\frac{1}{2}\right)^{-D/2} \tag{2.18}$$

$$= \left(\frac{1}{2}e^{\frac{M}{2\sigma^2\gamma(f)}}\right)^{-D/2} \tag{2.19}$$

If $M > 2\ln(2)\sigma^2\gamma(f)$, this last quantity will converge to 0 as $D \to \infty$. Therefore, for fixed $\delta \in (0,1)$ and $M > 2\ln(2)\sigma^2\gamma(f)$ there exists a D_0 such that

$$\mathbb{P}\left(\left\|\frac{\partial^2 \hat{f}}{\partial x_i^2}\right\|_\infty < M\right) = 1 - \mathbb{P}\left(\left\|\frac{\partial^2 \hat{f}}{\partial x_i^2}\right\|_\infty \geq M\right) \geq 1 - \delta$$

whenever $D \geq D_0$. Taking $M = 2\ln(2)\sigma^2\gamma(f)$, this concludes the proof. □

Lemma 4. *Let f be as in Theorem 2 and $\delta, \rho > 0$. Then, there exists a constant $D_0 = D_0(f, \delta, \rho)$ such that a random basis expansion $\hat{f}$ with $\mathbf{c} \in \mathscr{C}$ satisfies $\left\|\frac{\partial f}{\partial x_i} - \frac{\partial \hat{f}}{\partial x_i}\right\|_\infty \leq \rho$ with probability $1-\delta$ whenever $D \geq D_0$, for all i.*

Proof. Let $\mathbf{e}_i$ be the unit vector $[0,\ldots,0,1,0,\ldots,0]^T$. Taylor's theorem in Lagrange form [23, p.880] implies

$$\begin{aligned}
&\left\|\frac{\partial f(\mathbf{x})}{\partial x_i} - \frac{\partial \hat{f}(\mathbf{x})}{\partial x_i}\right\| \\
\leq &\left|\frac{f(\mathbf{x}+h\mathbf{e}_i) - f(\mathbf{x}) - (\hat{f}(\mathbf{x}+h\mathbf{e}_i) - \hat{f}(\mathbf{x}))}{h}\right| \\
&+ h\left(\left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_\infty + \left\|\frac{\partial^2 \hat{f}}{\partial x_i^2}\right\|_\infty\right) \\
\leq &\frac{1}{h}\left|f(\mathbf{x}+h\mathbf{e}_i) - \hat{f}(\mathbf{x}+h\mathbf{e}_i)\right| + \frac{1}{h}\left|f(\mathbf{x}) - \hat{f}(\mathbf{x})\right| \\
&+ h\left(\left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_\infty + \left\|\frac{\partial^2 \hat{f}}{\partial x_i^2}\right\|_\infty\right) \\
\leq &\frac{2\|f - \hat{f}\|_\infty}{h} + h\left(\left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_\infty + \left\|\frac{\partial^2 \hat{f}}{\partial x_i^2}\right\|_\infty\right).
\end{aligned}$$

Note that $\left\|\partial^2 f/\partial x_i^2\right\|_\infty$ is bounded since $\partial^2 f/\partial x_i^2$ is continuous by assumption and the set $\mathscr{X}$ is compact. By Lemma 3, there exists a $D_0^{(1)}$ such that

$$h\left(\left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_\infty + \left\|\frac{\partial^2 \hat{f}}{\partial x_i^2}\right\|_\infty\right) \leq \rho/2$$

with probability at least $1-\delta^{(1)}$ whenever $D \geq D_0^{(1)}$, if we choose $h = \frac{1}{4}\rho / \max\left\{M, \left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_\infty\right\}$.

By Theorem 1, for the same h, there exists $D_0^{(2)}$ such that $\frac{2\|f-\hat{f}\|_\infty}{h} \leq \rho/2$ with probability at least $1-\delta^{(2)}$ whenever $D \geq D_0^{(2)}$. Taking $D_0 = \max\{D_0^{(1)}, D_0^{(2)}\}$ and choosing $\delta^{(1)}$ and $\delta^{(2)}$ such that $(1-\delta^{(1)})(1-\delta^{(2)}) = 1-\delta$, we have

$$\begin{aligned}
&\mathbb{P}\left(\left\|\frac{\partial f(\mathbf{x})}{\partial x_i} - \frac{\partial \hat{f}(\mathbf{x})}{\partial x_i}\right\| \leq \rho\right) \\
&\geq \mathbb{P}\left(\frac{2\|f-\hat{f}\|_\infty}{h} + h\left(\left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_\infty + \left\|\frac{\partial^2 \hat{f}}{\partial x_i^2}\right\|_\infty\right) \leq \rho\right) \\
&\geq \mathbb{P}\left(\frac{2\|f-\hat{f}\|_\infty}{h} \leq \rho/2,\ h\left(\left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_\infty + \left\|\frac{\partial^2 \hat{f}}{\partial x_i^2}\right\|_\infty\right) \leq \rho/2\right) \\
&\geq 1-\delta
\end{aligned}$$

whenever $D \geq D_0$, for all i.

□

Lemma 5. *Let $g : \mathscr{X} \to \mathbb{R}$ be continuous with finitely many roots $\mathbf{r}_1, \ldots, \mathbf{r}_n$, $n \geq 1$, and fix any $\varepsilon > 0$. Then, there exists $\rho > 0$ such that*

$$|g(\mathbf{x})| < \rho \quad \Longrightarrow \quad \min_{j=1,\ldots,n} \|\mathbf{r}_j - \mathbf{x}\| < \varepsilon.$$

Proof. Define $G : \mathscr{X} \to \mathbb{R}$, $G(\mathbf{x}) := |g(\mathbf{x})|$, $\mathscr{B}_\varepsilon(\mathbf{r}) := \{\mathbf{x} \in \mathscr{X} : \|\mathbf{r}-\mathbf{x}\| < \varepsilon\}$, and $G_\varepsilon : \mathscr{X} \setminus \bigcup_{j=1}^n \mathscr{B}_\varepsilon(\mathbf{r}_j) \to \mathbb{R}$, $G_\varepsilon(\mathbf{x}) := G(\mathbf{x})$. The function G_ε is continuous and defined on a compact set. Therefore, it attains its minimum

$$\rho := \min_{\mathbf{x} \in \mathscr{X} \setminus \bigcup_{j=1}^n \mathscr{B}_\varepsilon(\mathbf{r}_j)} G_\varepsilon(\mathbf{x}) > 0.$$

Thus, whenever $|g(\mathbf{x})| = G(\mathbf{x}) < \rho$ for some $\mathbf{x} \in \mathscr{X}$, $\mathbf{x}$ cannot belong to the domain of G_ε. Instead, it is $\mathbf{x} \in \bigcup_{j=1}^n \mathscr{B}_\varepsilon(\mathbf{r}_j)$ as claimed. □

Using the lemmas above, we are ready to prove Theorem 2.

Proof. (Theorem 2)
First, choose $\rho > 0$ small enough such that

$$\left|\frac{\partial f(\mathbf{x})}{\partial x_i}\right| < \rho \ \forall i \quad \Longrightarrow \quad \min_{j=1,\ldots,n} \|\mathbf{v}_j - \mathbf{x}\| < \varepsilon \tag{2.20}$$

(possible by Lemma 5). Now, choose D_0 large enough such that

$$\left\|\frac{\partial f}{\partial x_i} - \frac{\partial \hat{f}}{\partial x_i}\right\|_\infty < \rho \ \forall i \tag{2.21}$$

with probability at least $1-\delta$ (possible by Lemma 4). Then, as claimed,

$$\nabla \hat{f}(\mathbf{x}) = \mathbf{0} \implies \forall i \left| \frac{\partial f(\mathbf{x})}{\partial x_i} \right| = \left| \frac{\partial f(\mathbf{x})}{\partial x_i} - \frac{\partial \hat{f}(\mathbf{x})}{\partial x_i} \right| \overset{(2.21)}{<} \rho$$
$$\overset{(2.20)}{\implies} \min_{j=1,\ldots,n} \|\mathbf{v}_j - \mathbf{x}\| < \varepsilon.$$

□

REFERENCES

[1] L. Bliek, M. Verhaegen, and S. Wahls, *Data-driven minimization with random feature expansions for optical beam forming network tuning,* 16th IFAC Workshop on Control Applications of Optimization (CAO'2015) **48**, 166 (2015).

[2] Z.-S. Hou and Z. Wang, *From model-based control to data-driven control: Survey, classification and perspective,* Information Sciences **235**, 3 (2013).

[3] M. Gevers, *Modelling, identification and control,* in *Iterative identification and control* (Springer, 2002) pp. 3–16.

[4] A. Rahimi and B. Recht, *Random features for large-scale kernel machines,* in *Adv. Neur. In.* (2007) pp. 1177–1184.

[5] L. Zhuang, *Ring resonator-based broadband photonic beam former for phased array antennas*, Ph.D. thesis, University of Twente (2010).

[6] E.-W. Bai, *An optimal two-stage identification algorithm for Hammerstein-–Wiener nonlinear systems,* Automatica **34**, 333 (1998).

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors,* Cognitive modeling **5**, 3 (1988).

[8] J. A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, J. Suykens, and T. Van Gestel, *Least squares support vector machines*, Vol. 4 (World Scientific, 2002).

[9] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, *An experimental unification of reservoir computing methods,* Neural Networks **20**, 391 (2007).

[10] G.-B. Huang, *An insight into extreme learning machines: random neurons, random features and kernels,* Cognitive Computation **6**, 376 (2014).

[11] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, *Feedforward neural networks with random weights,* in *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on* (IEEE, 1992) pp. 1–4.

[12] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, *Extreme learning machine: theory and applications,* Neurocomputing **70**, 489 (2006).

[13] B. Igelnik and Y.-H. Pao, *Stochastic choice of basis functions in adaptive function approximation and the functional-link net,* Neural Networks, IEEE Transactions on **6**, 1320 (1995).

[14] A. Rahimi and B. Recht, *Uniform approximation of functions with random bases,* in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on* (IEEE, 2008) pp. 555–561.

[15] G. H. Golub, P. C. Hansen, and D. P. O'Leary, *Tikhonov regularization and total least squares,* SIAM Journal on Matrix Analysis and Applications **21**, 185 (1999).

[16] A. Meijerink, C. G. Roeloffzen, R. Meijerink, L. Zhuang, D. A. Marpaung, M. J. Bentum, M. Burla, J. Verpoorte, P. Jorna, A. Hulzinga, *et al.*, *Novel ring resonator-based integrated photonic beamformer for broadband phased array receive antennas—part I: Design and performance analysis,* J. Lightwave Technol. **28**, 3 (2010).

[17] J. Capmany and D. Novak, *Microwave photonics combines two worlds,* Nature Photonics **1**, 319 (2007).

[18] G. Lenz, B. Eggleton, C. K. Madsen, and R. Slusher, *Optical delay lines based on optical filters,* Quantum Electronics, IEEE Journal of **37**, 525 (2001).

[19] C. Roeloffzen, L. Zhuang, R. Heideman, A. Borreman, and v. W. Etten, *Ring resonator-based tunable optical delay line in LPCVD waveguide technology,* in *Proceedings Symposium IEEE/LEOS Benelux Chapter* (IEEE, 2005) pp. 79–82.

[20] J. Bergstra and Y. Bengio, *Random search for hyper-parameter optimization,* The Journal of Machine Learning Research **13**, 281 (2012).

[21] I. Steinwart, *On the influence of the kernel on the consistency of support vector machines,* The Journal of Machine Learning Research **2**, 67 (2002).

[22] N. Harvey, *Lecture notes of Lecture 6 of CPSC 536N: Randomized Algorithms,* University of British Columbia (2011).

[23] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: With formulas, graphs, and mathematical tables*, 55 (Courier Corporation, 1964).

3

ONLINE OPTIMIZATION WITH COSTLY AND NOISY MEASUREMENTS USING RANDOM FOURIER EXPANSIONS

This paper analyzes DONE, an online optimization algorithm that iteratively minimizes an unknown function based on costly and noisy measurements. The algorithm maintains a surrogate of the unknown function in the form of a random Fourier expansion (RFE). The surrogate is updated whenever a new measurement is available, and then used to determine the next measurement point. The algorithm is comparable to Bayesian optimization algorithms, but its computational complexity per iteration does not depend on the number of measurements. We derive several theoretical results that provide insight on how the hyper-parameters of the algorithm should be chosen. The algorithm is compared to a Bayesian optimization algorithm for an analytic benchmark problem and three applications, namely, optical coherence tomography, optical beam-forming network tuning, and robot arm control. It is found that the DONE algorithm is significantly faster than Bayesian optimization in the discussed problems, while achieving a similar or better performance.

This chapter is based on a joint work with H.R.G.W. Verstraete, with an equal contribution from both parties, and also appears in: H.R.G.W. Verstraete, *Optimization-based adaptive optics for optical coherence tomography*, Ph.D. thesis, Delft University of Technology (2017).

Parts of this chapter have been published in [1].

3.1. INTRODUCTION

MANY optimization algorithms use the derivative of an objective function, but often this information is not available in practice. Regularly, a closed form expression for the objective function is not available and function evaluations are costly. Examples are objective functions that rely on the outcome of a simulation or an experiment. Approximating derivatives with finite differences is costly in high-dimensional problems, especially if the objective function is costly to evaluate. More efficient algorithms for derivative-free optimization (DFO) problems exist. Typically, in DFO algorithms a model is used that can be optimized without making use of the derivative of the underlying function [2, 3]. Some examples of commonly used DFO algorithms are the simplex method [4], NEWUOA [5], BOBYQA [6], and DIRECT [7]. Additionally, measurements of a practical problem are usually corrupted by noise. Several techniques have been developed to cope with a higher noise level and make better use of the expensive objective functions evaluations. Filtering and pattern search optimization algorithms such as implicit filtering [8] and SID-PSM [9] can handle local minima resulting from high frequency components. Bayesian optimization, also known as sequential Kriging optimization, deals with heteroscedastic noise and perturbations very well. One of the first and best known Bayesian optimization algorithms is EGO [10]. Bayesian optimization relies on a surrogate model that represents a probability distribution of the unknown function under noise, for example Gaussian processes or Student's-t processes [11–14]. In these processes different kernels and kernel learning methods are used for the covariance function [15, 16]. The surrogate model is used to decide where the next measurement should be taken. New measurements are used to update the surrogate model. Bayesian optimization has been successfully used in various applications, including active user modeling and reinforcement learning [17], robotics [18], hyper-parameter tuning [12], and optics [19].

Recently, the Data-based Online Nonlinear Extremum-seeker (DONE) algorithm was proposed in [20]. It is similar to Bayesian optimization, but simpler and faster. The DONE algorithm uses random Fourier expansions [21] (RFEs) as a surrogate model. The nature of the DONE algorithm makes the understanding of the hyper-parameters easier. In RFE models certain parameters are chosen randomly. In this paper, we derive a close-to-optimal probability distribution for some of these parameters. We also derive an upper bound for the regularization parameter used in the training of the RFE model.

The advantages of the DONE algorithm are illustrated in an analytic benchmark problem and three applications. We numerically compare DONE to BayesOpt [14], a Bayesian optimization library that was shown to outperform many other similar libraries in [14]. The first application is optical coherence tomography (OCT), a 3D imaging method based on interference often used to image the human retina [20, 22, 23]. The second application we consider is the tuning of an optical beam-forming network (OBFN). OBFNs are used in wireless communication systems to steer phased array antennas in the desired direction by making use of positive interference of synchronized signals [24–29]. The third application is a robot arm of which the tip has to be directed to a desired position [30].

This paper is organized as follows. Section 3.2 gives a short overview and provides new theoretical insights on random Fourier expansions, the surrogate model on which

the DONE algorithm is based. We have noticed a gap in the literature, where approximation guarantuees are given for ideal, but unknown RFE weights, while in practice RFE weights are computed via linear least squares. We investigate several properties of the ideal weights and combine these results with existing knowledge of RFEs to obtain approximation guarantees for least-square weights. Section 3.3 explains the DONE algorithm. Theoretically optimal as well as more practical ways to choose the hyperparameters of this algorithm are given in Section 3.4. In Section 3.5 the DONE algorithm and BayesOpt are compared for a benchmark problem and for the three aforementioned applications. We conclude the paper in Section 3.6.

3.2. RANDOM FOURIER EXPANSIONS

In this section, we will describe the surrogate model that we will use for optimization. There is a plethora of black-box modeling techniques to approximate a function from measurements available in the literature, with neural networks, kernel methods, and of course classic linear models probably being the most popular [31–33]. In this paper, we use random Fourier expansions (RFEs) [21] to model the unknown function because they offer a unique mix of computational efficiency, theoretical guarantees and ease of use that make them ideal for online processing. While general neural networks are more expressive than random Fourier features, they are difficult to use and come without theoretical guarantees. Standard kernel methods suffer from high computational complexity because the number of kernels equals the number of measurements. RFEs have been originally introduced to reduce the computational burden that comes with kernel methods, as will be explained next [21, 34, 35].

Assume that we are provided N scalar measurements y_i taken at measurement points $\mathbf{x}_i \in \mathbb{R}^d$ as well as a kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ that, in a certain sense, measures the closeness of two measurement points. To train the kernel expansion

$$g_{KM}(\mathbf{x}) = \sum_{i=1}^{N} a_i k(\mathbf{x}, \mathbf{x}_i), \tag{3.1}$$

a linear system involving the kernel matrix $[k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j}$ has to be solved for the coefficients a_i. The computational costs of training and evaluating (3.1) grow cubicly and linearly in the number of datapoints N, respectively. This can be prohibitive for large values of N. We now explain how RFEs can be used to reduce the complexity [21]. Assuming the kernel k is shift-invariant and has Fourier transform p, it can be normalized such that p is a probability distribution [21]. That is, we have

$$k(\mathbf{x}_i - \mathbf{x}_j) = \int_{\mathbb{R}^d} p(\omega) e^{-i\omega^T(\mathbf{x}_i - \mathbf{x}_j)} d\omega. \tag{3.2}$$

We will use several trigonometric properties and the fact that k is real to continue the derivation. This gives

$$\begin{aligned} k(\mathbf{x}_i - \mathbf{x}_j) &= \int_{\mathbb{R}^d} p(\omega) \cos(\omega^T(\mathbf{x}_i - \mathbf{x}_j)) d\omega \\ &= \int_{\mathbb{R}^d} p(\omega) \cos(\omega^T(\mathbf{x}_i - \mathbf{x}_j)) \end{aligned}$$

$$
\begin{aligned}
&+ p(\omega) \int_0^{2\pi} \cos(\omega^T(\mathbf{x}_i + \mathbf{x}_j) + 2b)\, db\, d\omega \\
&= \frac{1}{2\pi} \int_{\mathbb{R}^d} p(\omega) \int_0^{2\pi} \cos(\omega^T(\mathbf{x}_i - \mathbf{x}_j)) \\
&\quad + \cos(\omega^T(\mathbf{x}_i + \mathbf{x}_j) + 2b)\, db\, d\omega \\
&= \frac{1}{2\pi} \int_{\mathbb{R}^d} p(\omega) \int_0^{2\pi} 2\cos(\omega^T \mathbf{x}_i + b) \\
&\quad \cdot \cos(\omega^T \mathbf{x}_j + b)\, db\, d\omega \\
&= \mathbb{E}[2\cos(\boldsymbol{\Omega}^T \mathbf{x}_i + \mathbf{b}) \cos(\boldsymbol{\Omega}^T \mathbf{x}_j + \mathbf{b})] \\
&\approx \frac{2}{D} \sum_{k=1}^{D} \cos(\omega_k^T \mathbf{x}_i + b_k) \cos(\omega_k^T \mathbf{x}_j + b_k), \qquad (3.3)
\end{aligned}
$$

if ω_k are independent samples of the random variable $\boldsymbol{\Omega}$ with probability distribution function (p.d.f.) p, and $b_k \in [0, 2\pi]$ are independent samples of the random variable $\mathbf{b}$ with a uniform distribution. For $c_k = \sum_{i=1}^{N} \frac{2}{D} a_i \cos(\omega_k^T \mathbf{x}_i + b_k)$ we thus have:

$$
g_{KM}(\mathbf{x}) \approx \sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k). \qquad (3.4)
$$

Note that the number of coefficients D is now independent of the number of measurements N. This is especially advantageous in online applications where the number of measurements N keeps increasing. We use the following definition of a random Fourier expansion.

Definition 1. *A Random Fourier Expansion (RFE) is a function of the form* $g : \mathbb{R}^d \to \mathbb{R}$,

$$
g(\mathbf{x}) = \sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k), \qquad (3.5)
$$

with $D \in \mathbb{N}$, *the* b_k *being realizations of independent and identically distributed (i.i.d.) uniformly distributed random variables* $\mathbf{b}_k$ *on* $[0, 2\pi]$, *and with the* $\omega_k \in \mathbb{R}^d$ *being realizations of i.i.d. random vectors* $\boldsymbol{\Omega}_k$ *with an arbitrary continuous p.d.f.* $p_{\boldsymbol{\Omega}}$. *The* $\mathbf{b}_k$ *and the* $\boldsymbol{\Omega}_k$ *are assumed to be mutually independent.*

We finally remark that there are other approaches to reduce the complexity of kernel methods and make them suitable for online processing, which are mainly based on sparsity [36–39]. However, these are much more difficult to tune than using RFEs [35]. It is also possible to use other basis functions instead of the cosine, but the cosine was among the top performers in an exhaustive comparison with similar models [40]. Moreover, the parameters of the cosines have intuitive interpretations in terms of the Fourier transform.

3.2.1. IDEAL RFE WEIGHTS

In this section, we deal with the problem of fitting a RFE to a given function f. We derive ideal but in practice unknown weights c. We start with the case of infinitely many

samples and basis functions (see also [41, 42]), which corresponds to turning the corresponding sums into integrals.

Theorem 6. *Let $f \in L^2(\mathbb{R}^d)$ be a real-valued function and let*

$$\bar{c}(\omega, b) = \begin{cases} \frac{1}{\pi}|\hat{f}(\omega)|\cos(\angle\hat{f}(\omega) - b), & b \in [0, 2\pi], \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

Then, for all $\mathbf{x} \in \mathbb{R}^d$,

$$f(\mathbf{x}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \int_0^{2\pi} \bar{c}(\omega, b)\cos(\omega^T\mathbf{x} + b)\,db\,d\omega. \tag{3.7}$$

Here, $|\hat{f}|$ and $\angle\hat{f}$ denote the magnitude and phase of the Fourier transform $\hat{f}(\omega) = \int_{\mathbb{R}^d} f(\mathbf{x})e^{-i\omega^T\mathbf{x}}d\mathbf{x}$. The sets L^2 and L^∞ denote the space of square integrable functions and the space of all essentially bounded functions, respectively.

Proof. For $b \in [0, 2\pi]$, we have

$$\begin{aligned} \bar{c}(\omega, b) &= \frac{1}{\pi}|\hat{f}(\omega)|\cos(\angle\hat{f}(\omega) - b) \\ &= \frac{1}{\pi}\mathrm{Re}\left\{\hat{f}(\omega)e^{-ib}\right\}. \end{aligned} \tag{3.8}$$

Using that $f(\mathbf{x})$ is real, we find that

$$\begin{aligned} f(\mathbf{x}) =& \mathrm{Re}\left\{\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\hat{f}(\omega)e^{i\omega^T\mathbf{x}}d\omega\right\} \\ =& \mathrm{Re}\Big\{\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\Big(\hat{f}(\omega)e^{i\omega^T\mathbf{x}}\frac{1}{2\pi}\int_0^{2\pi}1db + \\ & \hat{f}(\omega)e^{-i\omega^T\mathbf{x}}\underbrace{\int_0^{2\pi}e^{-2ib}db}_{=0}\Big)d\omega\Big\} \\ =& \mathrm{Re}\left\{\frac{1}{\pi}\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\hat{f}(\omega)e^{-ib}\right. \\ & \left.\frac{1}{2}\left[e^{i(\omega^T\mathbf{x}+b)} + e^{-i(\omega^T\mathbf{x}+b)}\right]dbd\omega\right\} \\ =& \mathrm{Re}\left\{\frac{1}{\pi}\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\hat{f}(\omega)e^{-ib}\cos(\omega^T\mathbf{x}+b)dbd\omega\right\} \\ \overset{(3.8)}{=}& \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\omega, b)\cos(\omega^T\mathbf{x}+b)dbd\omega. \end{aligned} \tag{3.9}$$

□

For $b \in [0, 2\pi]$, we have another useful expression for the ideal weights that is used later on in this section, namely

$$\bar{c}(\omega, b) = \frac{1}{\pi}\mathrm{Re}\left\{\hat{f}(\omega)e^{-ib}\right\}$$

$$= \frac{1}{\pi}\mathrm{Re}\left\{\int_{\mathbb{R}^d} f(\mathbf{x})e^{-i(\omega^T\mathbf{x}+b)}d\mathbf{x}\right\}$$
$$= \frac{1}{\pi}\int_{\mathbb{R}^d} f(\mathbf{x})\cos(\omega^T\mathbf{x}+b)d\mathbf{x}. \tag{3.10}$$

The function $\bar{c}$ in Theorem 6 is not unique. However, of all functions c that satisfy (3.7), the given $\bar{c}$ is the one with minimum norm.

Theorem 7. *Let $\bar{c}$ be as in Theorem 6. If $\tilde{c}:\mathbb{R}^d\times[0,2\pi]\to\mathbb{R}$ satisfies*

$$f(\mathbf{x}) = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\tilde{c}(\omega,b)\cos(\omega^T\mathbf{x}+b)dbd\omega \quad \text{a.e.} \tag{3.11}$$

then $||\tilde{c}||_{L^2}^2 \geq ||\bar{c}||_{L^2}^2 = \frac{(2\pi)^d}{\pi}||f||_{L^2}^2$, with equality if and only if $\tilde{c}=\bar{c}$ in the L^2 sense.

Proof. First, using Parseval's theorem and $\int_0^{2\pi}\cos(a-b)^2db=\pi$ for any real constant a, note that

$$\begin{aligned}
||\bar{c}||_{L^2}^2 &= \int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\omega,b)^2dbd\omega \\
&\overset{(3.6)}{=} \int_{\mathbb{R}^d}\int_0^{2\pi}\frac{1}{\pi^2}|\hat{f}(\omega)|^2\cos(\angle\hat{f}(\omega)-b)^2dbd\omega \\
&= \int_{\mathbb{R}^d}\frac{1}{\pi^2}|\hat{f}(\omega)|^2\int_0^{2\pi}\cos(\angle\hat{f}(\omega)-b)^2dbd\omega \\
&= \int_{\mathbb{R}^d}\frac{1}{\pi}|\hat{f}(\omega)|^2d\omega \\
&= \frac{(2\pi)^d}{\pi}\int_{\mathbb{R}^d}f(\mathbf{x})^2d\mathbf{x} = \frac{(2\pi)^d}{\pi}||f||_{L^2}^2.
\end{aligned} \tag{3.12}$$

Assume that $\tilde{c}(\omega,b)=\bar{c}(\omega,b)+q(\omega,b)$. Then we get

$$\begin{aligned}
&\int_{\mathbb{R}^d}f(\mathbf{x})^2d\mathbf{x} \\
&\overset{(3.11)}{=} \int_{\mathbb{R}^d}f(\mathbf{x})\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\tilde{c}(\omega,b)\cos(\omega^T\mathbf{x}+b)dbd\omega d\mathbf{x} \\
&= \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\tilde{c}(\omega,b)\int_{\mathbb{R}^d}f(\mathbf{x})\cos(\omega^T\mathbf{x}+b)d\mathbf{x}dbd\omega \\
&\overset{(3.10)}{=} \frac{\pi}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\tilde{c}(\omega,b)\bar{c}(\omega,b)dbd\omega \\
&= \frac{\pi}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\omega,b)^2+\bar{c}(\omega,b)q(\omega,b)dbd\omega \\
&\overset{(3.12)}{=} \int_{\mathbb{R}^d}f(\mathbf{x})^2d\mathbf{x}+\frac{\pi}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\omega,b)q(\omega,b)dbd\omega.
\end{aligned} \tag{3.13}$$

Following the above equality we can conclude that $\int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\omega,b)q(\omega,b)dbd\omega=0$. The following now holds:

$$||\tilde{c}||_{L^2}^2 = ||\bar{c}+q||_{L^2}^2$$

$$= \int_{\mathbb{R}^d} \int_0^{2\pi} \bar{c}(\omega,b)^2 + 2\bar{c}(\omega,b)\,q(\omega,b) + q(\omega,b)^2\,db\,d\omega$$
$$= ||\bar{c}||_{L^2}^2 + ||q||_{L^2}^2 \geq ||\bar{c}||_{L^2}^2. \tag{3.14}$$

Furthermore, equality holds if and only if $||q||_{L^2} = 0$. That is, the minimum norm solution is unique in L^2.

□

These results will be used to derive ideal weights for a RFE with a finite number of basis functions as in Definition 1 by sampling the weights in (3.6). We prove unbiasedness in the following theorem, while variance properties are analyzed in Appendix 3.8.

Theorem 8. *For any continuous p.d.f. $p_{\mathbf{\Omega}}$ with $p_{\mathbf{\Omega}}(\omega) > 0$ if $|\hat{f}(\omega)| > 0$, the choice*

$$C_k = \frac{2}{D(2\pi)^d} \frac{|\hat{f}(\mathbf{\Omega}_k)|}{p_{\mathbf{\Omega}}(\mathbf{\Omega}_k)} \cos(\angle \hat{f}(\mathbf{\Omega}_k) - \mathbf{b}_k) \tag{3.15}$$

makes the (stochastic) RFE $G(\mathbf{x}) = \sum_{k=1}^D C_k \cos(\mathbf{\Omega}_k^T\mathbf{x}+\mathbf{b}_k)$ an unbiased estimator, i.e., $f(\mathbf{x}) = \mathbb{E}[G(\mathbf{x})]$ for any $\mathbf{x} \in \mathbb{R}^d$.

Proof. Using Theorem 6, we have

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \int_0^{2\pi} \bar{c}(\omega,b)\cos(\omega^T\mathbf{x}+b)\,db\,d\omega \\ &= \mathbb{E}_{\mathbf{\Omega}_1,\mathbf{b}_1}\left[\frac{1}{(2\pi)^d p_{\mathbf{b}}(\mathbf{b}_1) p_{\mathbf{\Omega}}(\mathbf{\Omega}_1)} \bar{c}(\mathbf{\Omega}_1,\mathbf{b}_1)\cos(\mathbf{\Omega}_1^T\mathbf{x}+\mathbf{b}_1)\right] \\ &= \mathbb{E}_{\mathbf{\Omega}_{1\dots D},\mathbf{b}_{1\dots D}}\left[\sum_{k=1}^D \frac{2\pi\bar{c}(\mathbf{\Omega}_k,\mathbf{b}_k)}{D(2\pi)^d p_{\mathbf{\Omega}}(\mathbf{\Omega}_k)} \cos(\mathbf{\Omega}_k^T\mathbf{x}+\mathbf{b}_k)\right] \\ &\overset{(3.6)}{=} \mathbb{E}\left[\sum_{k=1}^D \frac{2}{D(2\pi)^d} \frac{|\hat{f}(\mathbf{\Omega}_k)|}{p_{\mathbf{\Omega}}(\mathbf{\Omega}_k)} \cos(\angle\hat{f}(\mathbf{\Omega}_k) - \mathbf{b}_k)\right. \\ &\qquad \left. \cos(\mathbf{\Omega}_k^T\mathbf{x}+\mathbf{b}_k)\right] = \mathbb{E}[G(\mathbf{x})]. \end{aligned} \tag{3.16}$$

□

These ideal weights enjoy many other nice properties such as infinity norm convergence [43]. In practice, however, a least squares approach is used for a finite D. This is investigated in the next subsection.

3.2.2. CONVERGENCE OF THE LEAST SQUARES SOLUTION

The ideal weights $\bar{c}$ depend on the Fourier transform of the unknown function f that we wish to approximate. Of course, this knowledge is not available in practice. We therefore assume a finite number of measurement points $\mathbf{x}_1,\dots,\mathbf{x}_N$ that have been drawn independently from a p.d.f. $p_{\mathbf{X}}$ that is defined on a compact set $\mathcal{X} \subseteq \mathbb{R}^d$, and corresponding measurements $y_1,\dots,y_N$, with $y_n = f(\mathbf{x}_n) + \eta_n$, where $\eta_1,\dots,\eta_N$ have been drawn independently from a zero-mean normal distribution with finite variance σ_H^2. The input and

noise terms are assumed independent of each other. We determine the weights c_k by minimizing the squared error

$$J_N(\mathbf{c}) = \sum_{n=1}^{N} \left(y_n - \sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x}_n + b_k) \right)^2 + \lambda \sum_{k=1}^{D} c_k^2 = ||\mathbf{y}_N - \mathbf{A}_N \mathbf{c}||_2^2 + \lambda ||\mathbf{c}||_2^2. \tag{3.17}$$

Here,

$$\mathbf{y}_N = \begin{bmatrix} y_1 \cdots y_N \end{bmatrix}^T,$$
$$\mathbf{A}_N = \begin{bmatrix} \cos(\omega_1^T \mathbf{x}_1 + b_1) & \cdots & \cos(\omega_D^T \mathbf{x}_1 + b_D) \\ \vdots & \ddots & \vdots \\ \cos(\omega_1^T \mathbf{x}_N + b_1) & \cdots & \cos(\omega_D^T \mathbf{x}_N + b_D) \end{bmatrix}, \tag{3.18}$$

and λ is a regularization parameter added to deal with noise, over-fitting and ill-conditioning.

Since the parameters ω_k, b_k are drawn from continuous probability distributions, only the weights c_k need to be determined, making the problem a linear least squares problem. The unique minimizer of J_N is

$$\mathbf{c_N} = \left(\mathbf{A}_N^T \mathbf{A}_N + \lambda \mathbf{I}_{D \times D} \right)^{-1} \mathbf{A}_N^T \mathbf{y}_N. \tag{3.19}$$

The following theorem shows that RFEs whose coefficient vector have been obtained through a least squares fit as in (3.19) can approximate the function f arbitrarily well. Similar results were given in [41–44], but we emphasize that these convergence results did concern RFEs employing the ideal coefficient vector given earlier in Theorem 8 that is unknown in practice. Our theorem, in contrast, concerns the practically relevant case where the coefficient vector has been obtained through a least-squares fit to the data.

Theorem 9. *The difference between the function f and the RFE trained with linear least squares can become arbitrarily small if enough measurements and basis functions are used. More precisely, suppose that $f \in L^2 \cap L^\infty$ and that $\sup_{\omega \in \mathbb{R}^D, b \in [0,2\pi]} \left| \frac{\bar{c}(\omega,b)}{p_{\mathbf{\Omega}}(\omega) p_{\mathbf{b}}(b)} \right| < \infty$. Then, for every $\epsilon > 0$ and $\delta > 0$, there exist constants N_0 and D_0 such that*

$$\int_{\mathscr{X}} \left(f(\mathbf{x}) - \sum_{k=1}^{D} C_{Nk} \cos(\mathbf{\Omega}_k^T \mathbf{x} + \mathbf{b}_k) \right)^2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} < \epsilon \tag{3.20}$$

for all $N \geq N_0$, $D \geq D_0$, $0 < \lambda \leq N\Lambda$ with probability at least $1 - \delta$. Here, C_{Nk} is the k-th element of the random vector corresponding to the weight vector given in (3.19), and $\Lambda \geq 0$ is the solution to

$$\left|\left| \left(\mathbf{A}_N^T \mathbf{A}_N + N\Lambda \, \mathbf{I}_{D \times D} \right)^{-1} \mathbf{A}_N^T \mathbf{y}_N \right|\right|_2^2 = \sum_{k=1}^{D} \left(\frac{\bar{c}(\omega_k, b_k)}{(2\pi)^d D p_{\mathbf{\Omega}}(\omega_k) p_{\mathbf{b}}(b_k)} \right)^2. \tag{3.21}$$

The proof of this theorem is given in Appendix 3.7. In Section 3.4.2 we show how to obtain Λ in practice.

3.3. ONLINE OPTIMIZATION ALGORITHM

In this section, we will investigate the DONE algorithm, which locates a minimum of an unknown function f based on noisy evaluations of this function. Each evaluation, or *measurement*, is used to update a RFE model of the unknown function, based on which the next measurement point is determined. Updating this model has a constant computation time of order $O(D^2)$ per iteration, with D being the number of basis functions. We emphasize that this is in stark contrast to Bayesian optimization algorithms, where the computational cost of adding a new measurement increases with the total number of measurements so far. We also remark that the DONE algorithm operates *online* because the model is updated after each measurement. The advantage over offline methods, in which first all measurements are taken and only then processed, is that the number of required measurements is usually lower as measurement points are chosen adaptively.

3.3.1. RECURSIVE LEAST SQUARES APPROACH FOR THE WEIGHTS

In the online scenario, a new measurement y_n taken at the point $\mathbf{x}_n$ becomes available at each iteration $n = 1,2,\ldots$ These are used to update the RFE. Let $\mathbf{a}_n = [\cos(\omega_1^T\mathbf{x}_n + b_1)\cdots\cos(\omega_D^T\mathbf{x}_n + b_D)]$, then we aim to find the vector of RFE weights by minimizing the regularized mean square error

$$J_n(\mathbf{c}) = \sum_{i=1}^{n} \left(y_i - \mathbf{a}_i\mathbf{c}\right)^2 + \lambda||\mathbf{c}||_2^2. \tag{3.22}$$

Let $\mathbf{c}_n$ be the minimum of J_n,

$$\mathbf{c}_n = \underset{\mathbf{c}}{\operatorname{argmin}}\, J_n(\mathbf{c}). \tag{3.23}$$

Assuming we have found $\mathbf{c}_n$, we would like to use this information to find $\mathbf{c}_{n+1}$ without solving (3.23) again. The recursive least squares algorithm is a computationally efficient method that determines $\mathbf{c}_{n+1}$ from $\mathbf{c}_n$ as follows [45, Sec. 21]:

$$\gamma_n = 1/(1 + \mathbf{a}_n\mathbf{P}_{n-1}\mathbf{a}_n^T), \tag{3.24}$$

$$\mathbf{g}_n = \gamma_n\mathbf{P}_{n-1}\mathbf{a}_n^T, \tag{3.25}$$

$$\mathbf{c}_n = \mathbf{c}_{n-1} + \mathbf{g}_n(y_n - \mathbf{a}_n\mathbf{c}_{n-1}), \tag{3.26}$$

$$\mathbf{P}_n = \mathbf{P}_{n-1} - \mathbf{g}_n\mathbf{g}_n^T/\gamma_n, \tag{3.27}$$

with initialization $\mathbf{c}_0 = 0$, $\mathbf{P}_0 = \lambda^{-1}\mathbf{I}_{D\times D}$.

We implemented a square-root version of the above algorithm, also known as the inverse QR algorithm [45, Sec. 21], which is known to be especially numerically reliable. Instead of performing the update rules (3.24)-(3.27) explicitly, we find a rotation matrix Θ_n that lower triangularizes the upper triangular matrix in Eq. (3.28) below and generates a post-array with positive diagonal entries:

$$\begin{bmatrix} 1 & \mathbf{a}_n\mathbf{P}_{n-1}^{1/2} \\ \mathbf{0} & \mathbf{P}_{n-1}^{1/2} \end{bmatrix}\Theta_n = \begin{bmatrix} \gamma_n^{-1/2} & \mathbf{0} \\ \mathbf{g}_n\gamma_n^{-1/2} & \mathbf{P}_n^{1/2} \end{bmatrix}. \tag{3.28}$$

The rotation matrix Θ_n can be found by performing a QR decomposition of the transpose of the matrix on the left hand side of (3.28), or by the procedure explained in [45, Sec. 21]. The computational complexity of this update is $O(D^2)$ per iteration.

3.3.2. DONE ALGORITHM

We now explain the different steps of the DONE algorithm. The DONE algorithm is used to iteratively find a minimum of a function $f \in L^2$ on a compact set $\mathscr{X} \subseteq \mathbb{R}^d$ by updating a RFE $g(\mathbf{x}) = \sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k)$ at each new measurement, and using this RFE as a surrogate of f for optimization. It is assumed that the function f is unknown and only measurements perturbed by noise can be obtained: $y_n = f(\mathbf{x}_n) + \eta_n$. The algorithm consists of four steps that are repeated for each new measurement: **1)** take a new measurement, **2)** update the RFE, **3)** find a minimum of the RFE, **4)** choose a new measurement point. We now explain each step in more detail.

Initialization

Before running the algorithm, an initial starting point $\mathbf{x}_1 \in \mathscr{X}$ and the number of basis functions D have to be chosen. The parameters ω_k and b_k of the RFE expansion are drawn from continuous probability distributions as defined in Definition 1. The p.d.f. $p_{\mathbf{\Omega}}$ and the regularization parameter λ have to be chosen a priori as well. Practical ways for choosing the hyper-parameters will be discussed later in Sect. 3.4. These hyper-parameters stay fixed over the whole duration of the algorithm. Let $\mathbf{P}_0^{1/2} = \lambda^{-1/2}\mathbf{I}_{D\times D}$, and $n = 1$.

Step 1: New measurement

Unlike in Section 3.2.2, it is assumed that measurements are taken in a recursive fashion. At the start of iteration n, a new measurement $y_n = f(\mathbf{x}_n) + \eta_n$ is taken at the point $\mathbf{x}_n$.

Step 2: Update the RFE

As explained in Section 3.3.1, we update the RFE model $g(\mathbf{x}) = \sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k)$ based on the new measurement from Step 1 by using the inverse QR algorithm given in (3.24)-(3.27). Only the weights c_k are updated. The parameters ω_k and b_k stay fixed through-out the whole algorithm.

Step 3: Optimization on the RFE

After updating the RFE, an iterative optimization algorithm is used to find a (possibly local) minimum $\hat{\mathbf{x}}_n$ of the RFE. All derivatives of the RFE can easily be calculated. Using an analytic expression of the Jacobian will increase the performance of the optimization method used in this step, while not requiring extra measurements of f as in the finite difference method. For functions that are costly to evaluate, this is a big advantage. The method used in the proposed algorithm is an L-BFGS method [46, 47]. Other optimization methods can also be used. The initial guess for the optimization is the projection of the current measurement point plus a random perturbation:

$$\mathbf{x}_{\text{init}} = P_{\mathscr{X}}(\mathbf{x}_n + \zeta_n), \tag{3.29}$$

where $P_{\mathscr{X}}$ is the projection onto $\mathscr{X}$. The random perturbation prevents the optimization algorithm from starting exactly in the point where the model was trained. Increasing its value will increase the exploration capabilities of the DONE algorithm but might slow down convergence. In the proposed algorithm, ζ_n is chosen to be white Gaussian noise.

Step 4: Choose a new measurement point

The minimum found in the previous step is used to update the RFE again. A perturbation is added to the current minimum to avoid the algorithm getting trapped unnecessarily in insignificant local minima or saddle points [48]:

$$\mathbf{x}_{n+1} = P_{\mathscr{X}}(\hat{\mathbf{x}}_n + \xi_n). \tag{3.30}$$

The random perturbations can be seen as an exploration strategy and are again chosen to be white Gaussian noise. Increasing their variance σ_ξ increases the exploration capabilities of the DONE algorithm but might slow down convergence. In practice, we typically use the same distribution for ξ and ζ. Finally, the algorithm increases n and returns to Step 1.

The full algorithm is shown below in Algorithm 1 for the case $\mathscr{X} = [lb, ub]^d$.

Algorithm 1 DONE Algorithm

procedure DONE($f, \mathbf{x}_1, N, lb, ub, D, \lambda, \sigma_\zeta, \sigma_\xi$)
  Draw $\omega_1 \dots \omega_D$ from $p_{\mathbf{\Omega}}$ independently.
  Draw $b_1 \dots b_D$ from Uniform$(0, 2\pi)$ independently.
  $\mathbf{P}_0^{1/2} = \lambda^{-1/2}\mathbf{I}_{D\times D}$
  $\mathbf{c}_0 = [0 \dots 0]^T$
  $\hat{\mathbf{x}}_0 = \mathbf{x}_1$
  for $n = 1, 2, 3, \dots, N$ **do**
    $\mathbf{a}_n = [\cos(\omega_1^T \mathbf{x}_n + b_1) \cdots \cos(\omega_D^T \mathbf{x}_n + b_D)]$
    $y_n = f(\mathbf{x}_n) + \eta_n$
    $g(\mathbf{x}) = \text{updateRFE}(\mathbf{c}_{n-1}, \mathbf{P}_{n-1}^{1/2}, \mathbf{a}_n, y_n)$
    Draw ζ_n from $\mathcal{N}(0, \sigma_\zeta^2 \mathbf{I}_{d\times d})$.
    $\mathbf{x}_{\text{init}} = \max(\min(\mathbf{x}_n + \zeta_n, ub), lb)$
    $[\hat{\mathbf{x}}_n, \hat{g}_n] = \text{L-BFGS}(g(\mathbf{x}), \mathbf{x}_{\text{init}}, lb, ub)$
    Draw ξ_n from $\mathcal{N}(0, \sigma_\xi^2 \mathbf{I}_{d\times d})$.
    $\mathbf{x}_{n+1} = \max(\min(\hat{\mathbf{x}}_n + \xi_n, ub), lb)$
  return $\hat{\mathbf{x}}_n$

Algorithm 2 updateRFE

procedure UPDATERFE($\mathbf{c}_{n-1}, \mathbf{P}_{n-1}^{1/2}, \mathbf{a}_n, y_n$)
  Retrieve $\mathbf{g}_n \gamma_n^{-1/2}$, $\gamma_n^{-1/2}$ and $\mathbf{P}_n^{1/2}$ from (3.28)
  $\mathbf{c}_n = \mathbf{c}_{n-1} + \mathbf{g}_n(y_n - \mathbf{a}_n \mathbf{c}_{n-1})$
  $g(\mathbf{x}) = [\cos(\omega_1^T \mathbf{x} + b_1) \cdots \cos(\omega_D^T \mathbf{x} + b_D)]\mathbf{c}_n$
  return $g(\mathbf{x})$

3.4. CHOICE OF HYPER-PARAMETERS

In this section, we will analyze the influence of the hyper-parameters of the DONE algorithm and, based on these results, provide practical ways of choosing them. The perfor-

mance of DONE depends on the following hyper-parameters:

- number of basis functions D,
- p.d.f. $p_{\mathbf{\Omega}}$,
- regularization parameter λ,
- exploration parameters σ_ζ and σ_ξ.

The influence of D is straight-forward: increasing D will lead to a better performance (a better RFE fit) of the DONE algorithm at the cost of more computation time. Hence, D should be chosen high enough to get a good approximation, but not too high to avoid unnecessarily high computation times. It should be noted that D does not need to be very precise. Over-fitting should not be a concern for this parameter since we make use of regularization. The exploration parameters determine the trade-off between exploration and exploitation, similar to the use of the acquisition function in Bayesian optimization [16, 17]. The parameter σ_ζ influences the exploration of the RFE surrogate in Step 3 of the DONE algorithm, while σ_ξ determines exploration of the original function. Assuming both to be close to each other, σ_ζ and σ_ξ are usually chosen to be equal. If information about local optima of the RFE surrogate or of the original function is available, this could be used to determine good values for these hyper-parameters. Alternatively, similar to Bayesian optimization the expected improvement could be used for that purpose, but this remains for future work. The focus of this section will be on choosing $p_{\mathbf{\Omega}}$ and λ.

3.4.1. PROBABILITY DISTRIBUTION OF FREQUENCIES

Recall the parameters ω_k and b_k from Definition 1, which are obtained by sampling independently from the continuous probability distributions $p_{\mathbf{\Omega}}$ and $p_{\mathbf{b}} = \text{Uniform}(0, 2\pi)$, respectively. In the following, we will investigate the first and second order moments of the RFE and try to find a distribution $p_{\mathbf{\Omega}}$ that minimizes the variance of the RFE.

Unfortunately, as shown in Theorem 12 in Appendix 3.8, it turns out that the optimal p.d.f. is

$$p^*_{\mathbf{\Omega}}(\omega) = \frac{|\hat{f}(\omega)|\sqrt{\cos(2\angle\hat{f}(\omega) + 2\omega^T\mathbf{x}) + 2}}{\int_{\mathbb{R}^d} |\hat{f}(\tilde{\omega})|\sqrt{\cos(2\angle\hat{f}(\tilde{\omega}) + 2\tilde{\omega}^T\mathbf{x}) + 2}d\tilde{\omega}}. \tag{3.31}$$

This distribution depends on the input $\mathbf{x}$ and both the phase and magnitude of the Fourier transform of f. But if both $|\hat{f}|$ and $\angle\hat{f}$ were known, then the function f itself would be known, and standard optimization algorithms could be used directly. Furthermore, we would like to use a p.d.f. for ω_k that does not depend on the input $\mathbf{x}$, since the ω_k parameters are chosen independently from the input in the initialization step of the algorithm.

In calibration problems, the objective function f suffers from an unknown offset, $f(\mathbf{x}) = \tilde{f}(\mathbf{x} + \Delta)$. This unknown offset does not change the magnitude in the Fourier domain, but it does change the phase. Since the phase is thus unknown, we choose a

uniform distribution for $p_{\mathbf{b}}$ such that $b_k \in [0, 2\pi]$. However, the magnitude $|\hat{f}|$ can be measured in this case. Section 3.5.2 describes an example of such a problem. We will now derive a way to choose $p_{\mathbf{\Omega}}$ for calibration problems.

In order to get a close to optimal p.d.f. for ω_k that is independent of the input $\mathbf{x}$ and of the phase $\angle\hat{f}$ of the Fourier transform of f, we look at a complex generalization of the RFE. In this complex problem, it turns out we can circumvent the disadvantages mentioned above by using a p.d.f. that depends only on $|\hat{f}|$.

Theorem 10. *Let $\tilde{G}(\mathbf{x}) = \sum_{k=1}^{D} \tilde{C}_k e^{i\mathbf{\Omega}_k^T \mathbf{x} + \mathbf{b}_k}$, with $\mathbf{\Omega}_k$ being i.i.d. random vectors with a continuous p.d.f. $\tilde{p}_{\mathbf{\Omega}}$ over $\mathbb{R}^d$ that satisfies $\tilde{p}_{\mathbf{\Omega}_k}(\omega) > 0$ if $|\hat{f}(\omega)| > 0$, and $\mathbf{b}_k$ being random variables with uniform distribution from $[0, 2\pi]$. Then $\tilde{G}(\mathbf{x})$ is an unbiased estimator of $f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^d$ if*

$$\tilde{C}_k = \frac{\hat{f}(\mathbf{\Omega}_k) e^{-i\mathbf{b}_k}}{D(2\pi)^d \tilde{p}_{\mathbf{\Omega}}(\mathbf{\Omega}_k)}. \tag{3.32}$$

For this choice of $\tilde{C}_k$, the variance of $\tilde{G}(\mathbf{x})$ is minimal if

$$\tilde{p}_{\mathbf{\Omega}}(\omega) = \frac{|\hat{f}(\omega)|}{\int_{\mathbb{R}^d} |\hat{f}(\tilde{\omega})| d\tilde{\omega}}, \tag{3.33}$$

giving a variance of

$$\mathrm{Var}[\tilde{G}(\mathbf{x})] = \frac{1}{D(2\pi)^{2d}} \left(\int_{\mathbb{R}^d} |\hat{f}(\omega)| d\omega \right)^2 - f(\mathbf{x})^2. \tag{3.34}$$

Proof. The unbiasedness follows directly from the Fourier inversion theorem,

$$\begin{aligned}
\mathbb{E}\left[\tilde{G}(\mathbf{x})\right] &= \sum_{k=1}^{D} \int_{\mathbb{R}^d} \int_0^{2\pi} \frac{\hat{f}(\omega_k) e^{-ib_k} e^{i\omega_k^T \mathbf{x} + b_k}}{D(2\pi)^d \tilde{p}_{\mathbf{\Omega}}(\omega_k) 2\pi} db_k \tilde{p}_{\mathbf{\Omega}}(\omega_k) d\omega_k \\
&= D \int_{\mathbb{R}^d} \int_0^{2\pi} \frac{\hat{f}(\omega) e^{-ib}}{D(2\pi)^d \tilde{p}_{\mathbf{\Omega}}(\omega)} e^{i\omega^T \mathbf{x} + b} \frac{1}{2\pi} db \tilde{p}_{\mathbf{\Omega}}(\omega) d\omega \\
&= D \int_{\mathbb{R}^d} \frac{\hat{f}(\omega)}{D(2\pi)^d \tilde{p}_{\mathbf{\Omega}}(\omega)} e^{i\omega^T \mathbf{x}} \tilde{p}_{\mathbf{\Omega}}(\omega) \int_0^{2\pi} \frac{1}{2\pi} db d\omega \\
&= \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\omega) e^{i\omega^T \mathbf{x}} d\omega \\
&= f(\mathbf{x}).
\end{aligned} \tag{3.35}$$

The proof of minimum variance is similar to the proof of [49, Thm. 4.3.1]. □

Note that the coefficients $\tilde{C}_k$ can be complex in this case. Next, we show that the optimal p.d.f. for a complex RFE, $\tilde{p}_{\mathbf{\Omega}}$, is still close-to-optimal (in terms of the second moment) when used in the real RFE from Definition 1.

Theorem 11. *Let $\tilde{p}_{\mathbf{\Omega}}$ be as in (3.33) and let G with weights C_k be as in Theorem 8. Let P be the set of probability distribution functions for $\mathbf{\Omega}_k$ that are positive when $|\hat{f}(\omega)| > 0$. Then, we have*

$$\mathbb{E}_{\tilde{p}_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})^2] \leq \sqrt{3} \min_{p_{\mathbf{\Omega}} \in P} \mathbb{E}_{p_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})^2]. \tag{3.36}$$

The proof is given in Appendix 3.8. We now discuss how to choose $p_{\mathbf{\Omega}}$ in practice.

If no information of $|\hat{f}|$ is available, the standard approach of choosing $p_{\mathbf{\Omega}}$ as a zero-mean normal distribution can be used. The variance σ^2 is an important hyper-parameter in this case, and any method of hyper-parameter tuning can be used to find it. However, most hyper-parameter optimization methods are computationally expensive because they require running the whole algorithm multiple times. In the case that $|\hat{f}|$ is not exactly known, but some information about it is available (because it can be estimated or measured for example), this can be circumvented. The variance σ^2 can simply be chosen in such a way that $p_{\mathbf{\Omega}}$ most resembles the estimate for $|\hat{f}|$, using standard optimization techniques or by doing this by hand. In this approach, it is not necessary to run the algorithm at all, which is a big advantage compared to most hyper-parameter tuning methods. All of this leads to a rule of thumb for choosing $p_{\mathbf{\Omega}}$ as given in Algorithm 3.

Algorithm 3 Rule of thumb for choosing p_ω

1: **if** $|\hat{f}|$ is known exactly **then**
2: Set $p_{\mathbf{\Omega}} = |\hat{f}| / \int |\hat{f}(\omega)| d\omega$.
3: **else**
4: Measure or estimate $|\hat{f}|$.
5: Determine σ^2 for which the pdf of $\mathcal{N}(0, \sigma^2 \mathbf{I}_{d\times d})$ is close in shape to $|\hat{f}| / \int |\hat{f}(\omega)| d\omega$.
6: Set $p_{\mathbf{\Omega}} = \mathcal{N}(0, \sigma^2 \mathbf{I}_{d\times d})$.

3.4.2. UPPER BOUND ON THE REGULARIZATION PARAMETER

The regularization parameter λ in the performance criterion (3.17) is used to prevent under- or over-fitting of the RFE under noisy conditions or when dealing with few measurements. Theorem 9 guarantees the convergence of the least squares solution only if the regularization parameter satisfies $\lambda \leq N\Lambda$, where N is the total number of samples and Λ is defined in (3.21). Here we will provide a method to estimate Λ.

During the proof of Theorem 9, it was shown that the upper bound Λ corresponds to the λ that satisfies

$$\left|\left|\left(\mathbf{A}_N^T \mathbf{A}_N + N\lambda\, \mathbf{I}_{D\times D}\right)^{-1} \mathbf{A}_N^T \mathbf{y}_N\right|\right|_2^2 = \sum_{k=1}^{D} \left(\frac{\bar{c}(\omega_k, b_k)}{(2\pi)^d D\, p_{\mathbf{\Omega}}(\omega_k)\, p_{\mathbf{b}}(b_k)}\right)^2 = M^2. \tag{3.37}$$

The left-hand side in this equation is easily evaluated for different values of λ. Thus, in order to estimate Λ, all we need is an approximation of the unknown right hand M^2.

Like in Section 3.4.1, it is assumed that no information about $\angle \hat{f}$ is available, but that $|\hat{f}|$ can be measured or estimated. Under the assumptions that D is large and that $p_{\mathbf{\Omega}}$

is a good approximation of $\tilde{p}_{\mathbf{\Omega}} = |\hat{f}(\omega)| / \int_{\mathbb{R}^d} |\hat{f}(\omega)| d\omega$ as in Algorithm 3, we obtain the following approximation of M:

$$\begin{aligned} M &= \frac{2}{(2\pi)^d} \sqrt{\frac{1}{D^2} \sum_{k=1}^{D} \left(\frac{|\hat{f}(\omega_k)|}{p_{\mathbf{\Omega}}(\omega_k)} \cos(\angle \hat{f}(\omega_k) - b_k) \right)^2} \\ &\approx \frac{2}{(2\pi)^d} \sqrt{\frac{1}{D} \mathbb{E} \left[\left(\frac{|\hat{f}(\mathbf{\Omega}_1)|}{p_{\mathbf{\Omega}}(\mathbf{\Omega}_1)} \cos(\angle \hat{f}(\mathbf{\Omega}_1) - \mathbf{b}_1) \right)^2 \right]} \\ &= \frac{2}{(2\pi)^d} \sqrt{\frac{1}{2\pi D} \int_{\mathbb{R}^d} \int_0^{2\pi} \frac{|\hat{f}(\omega)|^2}{p_{\mathbf{\Omega}}(\omega)} \cos^2(\angle \hat{f}(\omega) - b) db d\omega} \\ &= \frac{\sqrt{2}}{(2\pi)^d \sqrt{D}} \sqrt{\int_{\mathbb{R}^d} \frac{|\hat{f}(\omega)|^2}{p_{\mathbf{\Omega}}(\omega)} d\omega} \\ &\approx \frac{\sqrt{2}}{(2\pi)^d \sqrt{D}} \sqrt{\int_{\mathbb{R}^d} \frac{|\hat{f}(\omega)|^2}{\tilde{p}_{\mathbf{\Omega}}(\omega)} d\omega} \\ &= \frac{\sqrt{2}}{(2\pi)^d \sqrt{D}} \int |\hat{f}(\omega)| d\omega = M_a. \end{aligned} \tag{3.38}$$

The squared cosine was removed as in Eq. (3.12). Using the exact value or an estimate of $\int_{\mathbb{R}^d} |\hat{f}(\omega)| d\omega$ as in Algorithm 3 to determine M_a, we calculate the left-hand in (3.37) for multiple values of Λ and take the value for which it is closest to M_a^2. The procedure is summarized in Algorithm 4.

Algorithm 4 Rule of thumb for finding an estimate of Λ

1: Run Algorithm 3 to get $\int_{\mathbb{R}^d} |\hat{f}(\omega)| d\omega$.
2: Take N measurements to get $\mathbf{A}_N$ and $\mathbf{y}_N$.
3: Determine Λ for which the left-hand side of (3.37) is close to $M_a^2 = \frac{2}{(2\pi)^{2d} D} \left(\int |\hat{f}(\omega)| d\omega \right)^2$.

3.5. NUMERICAL EXAMPLES

In this section, we compare the DONE algorithm to the Bayesian optimization library BayesOpt [14] in several numerical examples.

3.5.1. ANALYTIC BENCHMARK PROBLEM: CAMELBACK FUNCTION

The camelback function

$$f(\mathbf{x}) = \left(4 - 2.1 x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1 x_2 + \left(-4 + 4 x_2^2 \right) x_2^2, \tag{3.39}$$

where $\mathbf{x} = [x_1, x_2] \in [-2, 2] \times [-1, 1]$, is a standard test function with two global minima and two local minima. The locations of the global minima are approximately $(0.0898, -0.7126)$

and $(-0.0898, 0.7126)$ with an approximate function value of -1.0316. We determined the hyper-parameters for DONE on this test function as follows. First, we computed the Fourier transform of the function. We then fitted a function $h(\omega) = \frac{C}{\sigma\sqrt{2\pi}} e^{-\frac{\omega^2}{2\sigma^2}}$ to the magnitude of the Fourier transform in both directions. This was done by trial and error, giving a value of $\sigma = 10$. To validate, two RFEs were fit to the original function using a normal distribution with standard deviation $\sigma = 10$ (good fit) and $\sigma = 0.1$ (bad fit) for ω_k, using the least squares approach from Section 3.2.2. Here, we used $N = 1000$ measurements sampled uniformly from the input domain, the number of basis functions D was set to 500, and a regularization parameter of $\lambda = 10^{-10}$ was used. The small value for λ still works well in practice because the function f does not contain noise.

Let $g(\mathbf{x})$ denote the value of the trained RFE at point $\mathbf{x}$. We investigated the root mean squared error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(f(\mathbf{x}_n) - g(\mathbf{x}_n))^2}, \tag{3.40}$$

for the two stated values of σ. The good fit gave a RMSE of $5.5348 \cdot 10^{-6}$, while the bad fit gave a RMSE of 0.2321, which shows the big impact of this hyper-parameter on the least squares fit.

We also looked at the difference between using the real RFE from Definition 1 and the complex RFE from Theorem 10, for $\sigma = 10$, and for different values of D ($D \in \{10, 20, 40, 80, 160, 320, 640, 1280\}$). Fig. 3.1 shows the mean and standard deviation of the RMSE over 100 runs. We see that the real RFE indeed performs similar to the complex RFE as predicted by Theorem 13 in Appendix 3.8.

Using the hyper-parameters $\sigma = 10$ and $\lambda = 10^{-10}$, we also performed 10 runs of the DONE algorithm and compared it to reproduced results from [14, Table 1] (method "BayesOpt1"). The number of basis functions D was set to 500, one of the smallest values with a RMSE of below 10^{-5} according to Fig. 3.1, and the initial guess was chosen randomly. The exploration parameters σ_ζ and σ_ξ were set to 0.01. The resulting distance to the true minimum and the computation time in seconds (with their standard deviations) for 50 and 100 measurements can be found in Table 3.1. As in [14], the computation time for BayesOpt was only shown for 100 samples and the accuracy below 10^{-5} was not shown. It can be seen that the DONE algorithm is several orders of magnitude more accurate and about 5 times faster when compared to BayesOpt for this problem.

3.5.2. OPTICAL COHERENCE TOMOGRAPHY

Optical coherence tomography (OCT) is a low-coherence interferometry imaging technique used for making three-dimensional images of a sample. The quality and resolution of images is reduced by optical wavefront aberrations caused by the medium, e.g., the human cornea when imaging the retina. These aberrations can be removed by using active components such as deformable mirrors in combination with optimization algorithms [20, 23]. The arguments of the optimization can be the voltages of the deformable mirror or a mapping of these voltages to other coefficients such as the coefficients of Zernike polynomials. The intensity of the image at a certain depth is then maximized to

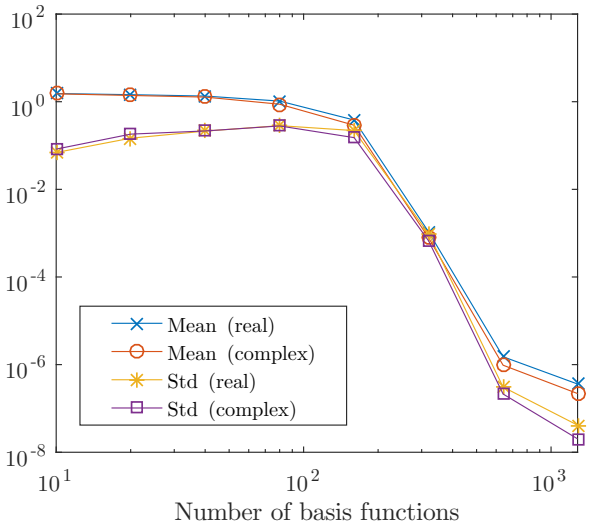


Figure 3.1: Mean and standard deviation of the root mean square error for a real and a complex RFE over 100 runs.

Table 3.1: DONE vs BayesOpt on the Camelback function

	Dist. to min. (50 samp.)	Time (50 samp.)
DONE	$2.1812 \cdot 10^{-9}$ ($8.3882 \cdot 10^{-9}$)	0.0493 (0.0015)
BayesOpt	0.0021 (0.0044)	-
	Dist. to min. (100 samp.)	Time (100 samp.)
DONE	$1.1980 \cdot 10^{-9}$ ($5.2133 \cdot 10^{-9}$)	0.0683 (0.0019)
BayesOpt	$< 1 \cdot 10^{-5}$ ($< 1 \cdot 10^{-5}$)	0.3049 (0.0563)

remove as much of the aberrations as possible. In [20] it was shown experimentally that the DONE algorithm greatly outperforms other derivative-free algorithms in final root mean square (RMS) wavefront error and image quality. Here, we numerically compare the DONE algorithm to BayesOpt [14]. The numerical results are obtained by simulating the OCT transfer function as described in [50, 51] and maximizing the OCT signal. The input dimension for this example is three. Three Zernike aberrations are considered, namely the defocus and two astigmatisms. These are generally the largest optical wavefront aberrations in the human eye. The noise of a real OCT signal is approximated by adding Gaussian white noise with a standard deviation of 0.01. The results are shown in Fig. 3.2. For the DONE algorithm the same parameters are used as described in [20], only λ is chosen to be equal to 3. The number of cosines $D = 1000$ is chosen as large as possible such that the computation time still remains around 1 ms. This is sufficiently fast to keep up with modern OCT B-scan acquisition and processing rates. The DONE algorithm is compared to BayesOpt with the default parameters and to BayesOpt with only one instead of 10 prior measurements, the latter is referred to as BayesOpt-1 init. Other values for the parameters of BayesOpt, obtained with trial and error, did not result in a significant performance increase. To use the BayesOpt algorithm, the inputs had to be normalized between 0 and 1. For each input aberration, the region -0.45 μm to 0.45 μm was scaled to the region 0 to 1. The results for BayesOpt and DONE are very similar. The mean error of the DONE algorithm is slightly lower than the BayesOpt algorithm. However, the total average computation time for the DONE algorithm was 93 ms, while the total average computation time of Bayesopt was 1019 ms.

3.5.3. TUNING OF AN OPTICAL BEAM-FORMING NETWORK

In wireless communication systems, optical beam-forming networks (OBFNs) can be used to steer the reception or transmission angle of a phased array antenna [24] in the desired direction. In the case of reception, the signals that arrive at the different antenna elements of the phased array are combined in such a way that positive interference of the signals occurs only in a specific direction. A device based on optical ring resonators [25] (ORRs) that can perform this signal processing technique in the optical domain was proposed in [26]. This OBFN can provide accurate control of the reception angle in broadband wireless receivers.

To achieve a maximal signal-to-noise ratio (SNR), the actuators in the OBFN need to be adapted according to the desired group delay of each OBFN path, which can be calculated from the desired reception angle. Each ORR is controlled by two heaters that influence its group delay, however the relation between heater voltage and group delay is nonlinear. Even if the desired group delay is available, controlling the OBFN comes down to solving a nonlinear optimization problem. Furthermore, the physical model of the OBFN can become quite complex if many ORRs are used, and the models are prone to model inaccuracies. Therefore, a black-box approach like in the DONE algorithm could help in the tuning of the OBFN. Preliminary results using RFEs in an offline fashion on this application can be found in [29]. Here, we demonstrate the advantage of online processing in terms of performance by using DONE instead of the offline algorithm in [29].

An OBFN simulation based on the same physical models as in [29] will be used in

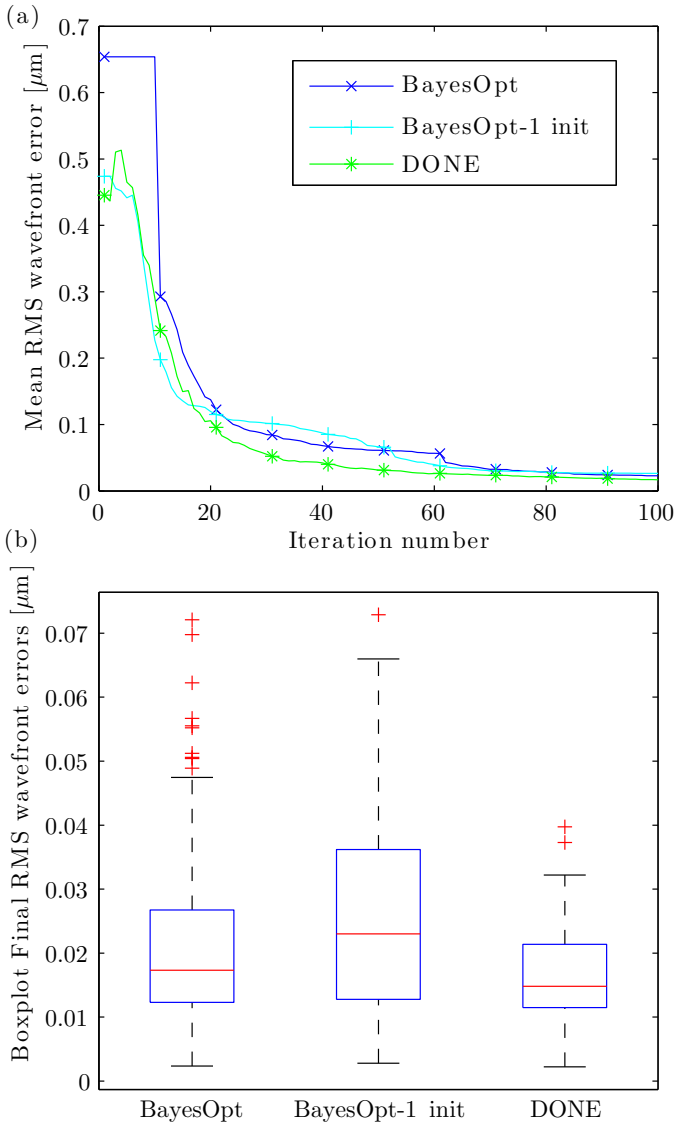


Figure 3.2: (a) The RMS wavefront error of DONE and BayesOpt averaged over 100 simulations versus the number of iterations. (b) A boxplot of 100 final RMS wavefront errors after 100 iterations for DONE and BayesOpt.On each box, the central line is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers. Outliers are plotted individually.

this section, with the following differences: 1) the implementation is done in C++; 2) ORR properties are equal for each ORR; 3) heater voltages with offset and crosstalk [28, Appendix B] have been implemented; 4) a small region outside the bandwidth of interest has a desired group delay of 0; 5) an 8×1 OBFN with 12 ORRs is considered; 6) the standard deviation of the measurement noise was set to $7.5 \cdot 10^{-3}$. The input of the simulation is the normalized heater voltage for each ORR, and the output is the corresponding mean square error of the difference between OBFN path group delays and desired delays. The simulation contains 24 heaters (two for each ORR, namely one for the phase shift and one for the coupling constant), making the problem 24-dimensional. Each heater influences the delay properties of the corresponding ORR, and together they influence the OBFN path group delays.

The DONE algorithm was used on this simulation to find the optimal heater voltages. The number of basis functions was $D = 6000$, which was the lowest number that gave an adequate performance. The p.d.f. $p_{\mathbf{\Omega}}$ was a normal distribution with variance 0.5. The regularization parameter was $\lambda = 0.1$. The exploration parameters were $\sigma_\zeta = \sigma_\xi = 0.01$. In total, 3000 measurements were taken.

Just like in the previous application, the DONE algorithm was compared to the Bayesian optimization library BayesOpt [14]. The same simulation was used in both algorithms, and BayesOpt also had 3000 function evaluations available. The other parameters for BayesOpt were set to their default values, except for the noise parameter which was set to 0.1 after calculating the influence of the measurement noise on the objective function. Also, in-between hyper-parameter optimization was turned off after noticing it did not influence the results while being very time-consuming.

The results for both algorithms are shown in Fig. 3.3. The found optimum at each iteration is shown for the two algorithms. For DONE, the mean of 10 runs is shown, while for BayesOpt only one run is shown because of the much longer computation time. The dotted line represents an offline approach: it is the average of 10 runs of a similar procedure as in [29], where a RFE with the same hyper-parameters as in DONE was fitted to 3000 random measurements and then optimized. The figure clearly shows the advantage of the online approach: because measurements are only taken in regions where the objective function is low, the RFE model can become very accurate in this region. The figure also shows that DONE outperforms BayesOpt for this application in terms of accuracy. On top of that, the total computation time shows a big improvement: one run of the DONE algorithm took less than 2 minutes, while one run of BayesOpt took 5800 minutes.

The big difference in computation time for the OBFN application can be explained by looking at the total number of measurements N. Even though the input dimension is high compared to the other problems, N is the main parameter that causes BayesOpt to slow down for a large number of measurements. This is because the models used in Bayesian optimization typically depend on the kernel matrix of all samples, which will increase in size each iteration. The runtime for one iteration of the DONE algorithm is, in contrast, independent of the number of previous measurements.

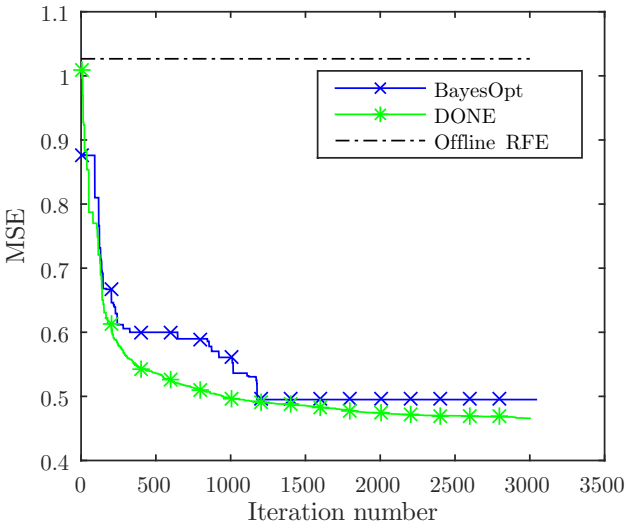


Figure 3.3: The mean square error of DONE and BayesOpt applied to the OBFN application, plotted versus the number of iterations. For DONE, the values are averaged over 10 runs. For BayesOpt only 1 run is shown. The dotted line is the result of fitting a RFE using 3000 random measurements and optimizing that RFE, averaged over 10 runs.

3.5.4. ROBOT ARM MOVEMENT

The previous two examples have illustrated how the DONE algorithm outperforms BayesOpt in terms of speed (both OCT and OFBN) and how its online processing scheme reduces the number of required measurements compared to offline processing (OFBN), respectively. The dimensions in both problems were three and 27, respectively, which is still relatively modest. To illustrate that DONE also works in higher dimensions, we will now consider a toy example from robotics. The following model of a three-link-planar robot, which has been adapted from [30], is considered:

$$a_i(k) = u_i(k) + \sin\left(\pi/180 \sum_{j=1}^{i} \alpha_j(k-1)\right) \cdot 9.8 \cdot 0.05, \tag{3.41}$$

$$v_i(k) = v_i(k-1) + a_i(k), \tag{3.42}$$

$$\alpha_i(k) = \alpha_i(k-1) + v_i(k), \tag{3.43}$$

$$x(k) = \sum_{j=1}^{3} l_j \cos\left(\pi/2 + \pi/180 \sum_{j=1}^{i} \alpha_j(k)\right), \tag{3.44}$$

$$y(k) = \sum_{j=1}^{3} l_j \sin\left(\pi/2 + \pi/180 \sum_{j=1}^{i} \alpha_j(k)\right). \tag{3.45}$$

Here, $\alpha_i(k)$ represents the angle in degrees of link i at time step k, $v_i(k)$ and $a_i(k)$ are the first and second derivative of the angles, $u_i(k) \in [-1,1]$ is the control input, $x(k)$ and $y(k)$

denote the position of the tip of the arm, and $l_1 = l_2 = 8.625$ and $l_3 = 6.125$ are the lengths of the links. The variables are initialized as $a_i(0) = v_i(0) = \alpha_i(0) = 0$ for $i = 1,2,3$. We use the DONE algorithm to design a sequence of control inputs $u_i(1),\ldots,u_i(50)$ such that the distance between the tip of the arm and a fixed target at location $(6.96, 12.66)$ at the 50-th time step is minimized. The input for the DONE algorithm is thus a vector containing $u_i(k)$ for $i = 1,2,3$ and $k = 1,\ldots,50$. This makes the problem 150-dimensional. The output is the distance between the tip and the target at the 50-th time step. The initial guess for the algorithm was set to a random control sequence with a uniform distribution over the set $[-1,1]$ for each robot arm i. We would like to stress that this example has been chosen for its high-dimensional input. We do not consider this approach a serious contender for specialized control methods in robotics.

The hyper-parameters for the DONE algorithm were chosen as follows. The number of basis functions was $D = 3000$, which was the lowest number that gave consistent results. The regularization parameter was $\lambda = 10^{-3}$. The p.d.f. $p_{\mathbf{\Omega}}$ was set to a normal distribution with variance one. The exploration parameters were set to $\sigma_\zeta = \sigma_\xi = 5\cdot 10^{-5}$. The number of measurements N was set to 10000.

No comparison with other algorithms has been made for this application. The computation time of the Bayesian optimization algorithm scales with the number of measurements and would be too long with 10000 measurements, as can be seen in Table 3.2. Algorithms like reinforcement learning use other principles, hence no comparison is given. Our main purpose with this application is to demonstrate the applicability of the DONE algorithm to high-dimensional problems. Figure 3.4 shows the distance to the target at time step 50 for different iterations of the DONE algorithm, averaged over 10 runs with different initial guesses. The control sequences converge to a sequence for which the robot arm goes to the target, i.e., DONE has successfully been applied to a problem

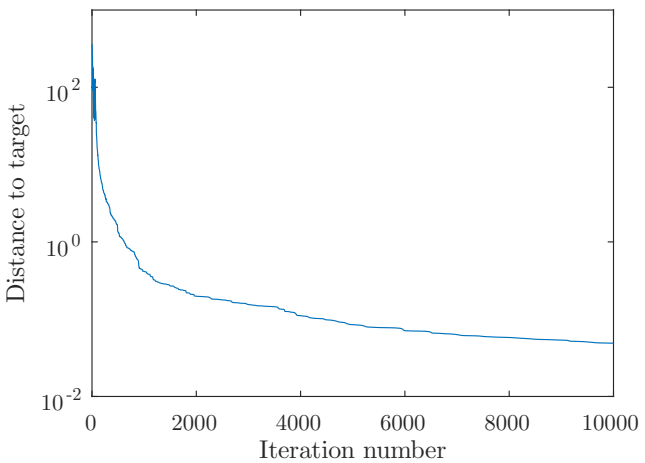


Figure 3.4: The mean distance to target for the robot arm at time step 50, after minimizing this distance with DONE, plotted versus the number of iterations, averaged over 10 runs.

with a high input dimension. The number of basis functions required did not increase when compared to the other applications in this paper, although more measurements were required. The computation time for this example and the other examples is shown in Table 3.2.

Table 3.2: Computation Time: DONE vs BayesOpt

Problem	Method	Input dim.	N	D	Time (s)
Camelback	DONE	2	100	50	0.0683
	BayesOpt	2	100	-	0.3049
OCT	DONE	3	100	1000	0.093
	BayesOpt	3	100	-	1.019
OBFN	DONE	24	3000	6000	99.7
	BayesOpt	24	3000	-	$3.48 \cdot 10^5$
Robot arm	DONE	150	10000	3000	99.1

3.6. CONCLUSIONS

We have analyzed an online optimization algorithm called DONE that is used to find the minimum of a function using measurements that are costly and corrupted by noise. DONE maintains a surrogate model in the form of a random Fourier expansion (RFE), which is updated whenever a new measurement is available, and minimizes this surrogate with standard derivative-based methods. This allows to measure only in regions of interest, reducing the overall number of measurements required. The DONE algorithm is comparable to Bayesian optimization algorithms, but it has the distinctive advantage that the computational complexity of one iteration does not grow with the number of measurements that have already been taken.

As a theoretical result, we have shown that a RFE that is trained with linear least squares can approximate square integrable functions arbitrarily well, with high probability. An upper bound on the regularization parameter used in this training procedure was given, as well as an optimal and a more practical probability distribution for the parameters that are chosen randomly. We applied the DONE algorithm to an analytic benchmark problem and to three applications: optical coherence tomography, optical beam-forming network tuning, and a robot arm. We compared the algorithm to BayesOpt, a Bayesian optimization library. The DONE algorithm gave accurate results on these applications while being faster than the Bayesian optimization algorithm, due to the fixed computational complexity per iteration.

3.7. Appendix: Proof of convergence of the least squares solution

In this section, we show that using the least squares solution in the RFE gives a function that approximates the true unknown function f. To prove this, we make use of the results in [43] and of [52, Thm. 2] and [53, Key Thm.].

Proof of Theorem 9. Let the constant $m > 0$ be given by

$$m = \left|\left|\left(\frac{1}{N}\mathbf{A}_N^T\mathbf{A}_N + \frac{\lambda}{N}\mathbf{I}_{D\times D}\right)^{-1}\frac{1}{N}\mathbf{A}_N^T\mathbf{y}_N\right|\right|_2, \tag{3.46}$$

and define the set $C_m = \{\mathbf{c} \in \mathbb{R}^D : \ ||\mathbf{c}||_2 \leq m\}$. Note that C_m is a compact set. The least squares weight vector

$$\begin{aligned}\mathbf{c_N} &= \left(\mathbf{A}_N^T\mathbf{A}_N + \lambda\mathbf{I}_{D\times D}\right)^{-1}\mathbf{A}_N^T\mathbf{y}_N \\ &= \left(\frac{1}{N}\mathbf{A}_N^T\mathbf{A}_N + \frac{\lambda}{N}\mathbf{I}_{D\times D}\right)^{-1}\frac{1}{N}\mathbf{A}_N^T\mathbf{y}_N,\end{aligned} \tag{3.47}$$

is also the solution to the constrained, but unregularized least squares problem (see [54, Sec. 12.1.3])

$$\mathbf{c_N} = \underset{\mathbf{c}\in C_m}{\operatorname{argmin}}\frac{1}{N}||\mathbf{y}_N - \mathbf{A}_N\mathbf{c}||_2^2. \tag{3.48}$$

Now, note that a decrease in λ leads to an increase in m. Since $\lambda/N \leq \Lambda$ by assumption and the upper bound Λ in Theorem 9 satisfies

$$\left|\left|\left(\frac{1}{N}\mathbf{A}_N^T\mathbf{A}_N + \Lambda\,\mathbf{I}_{D\times D}\right)^{-1}\frac{1}{N}\mathbf{A}_N^T\mathbf{y}_N\right|\right|_2 = M, \tag{3.49}$$

$$M = \sqrt{\sum_{k=1}^{D}\left(\frac{\bar{c}(\omega_k, b_k)}{(2\pi)^d D\, p_{\mathbf{\Omega}}(\omega_k)\, p_{\mathbf{b}}(b_k)}\right)^2}, \tag{3.50}$$

we have that $m \geq M$. We will need this lower bound on m to make use of the results in [43] later on in this proof.

Recall from Section 3.2.2 that the vector $\mathbf{y}_N$ depends on the function evaluations and on measurement noise η that is assumed to be zero-mean and of finite variance σ_H^2. We first consider the noiseless case, i.e. $y_n = f(\mathbf{x}_n)$. For $\mathbf{x} \in \mathcal{X}$, $\mathbf{c} \in \mathbb{R}^D$, let

$$E(\mathbf{x}, \mathbf{c}) = f(\mathbf{x}) - \sum_{k=1}^{D} c_k \cos(\omega_k^T\mathbf{x} + b_k). \tag{3.51}$$

Using the Cauchy-Schwarz inequality, we have the following bound for all $\mathbf{x} \in \mathcal{X}$, $\mathbf{c} \in C_m$:

$$E(\mathbf{x}, \mathbf{c})^2 = f(\mathbf{x})^2 + \left(\sum_{k=1}^{D} c_k \cos(\omega_k^T\mathbf{x} + b_k)\right)^2$$

$$
\begin{aligned}
&\quad -2f(\mathbf{x})\sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k) \\
&\leq f(\mathbf{x})^2 + \left(\sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k)\right)^2 \\
&\quad + 2\left|f(\mathbf{x})\right| \left|\sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k)\right| \\
&\leq f(\mathbf{x})^2 + \sum_{k=1}^{D} |c_k|^2 + 2\left|f(\mathbf{x})\right| \sqrt{\sum_{k=1}^{D} |c_k|^2} \\
&\leq f(\mathbf{x})^2 + m^2 + 2f(\mathbf{x})m \\
&\leq \left(||f||_\infty + m\right)^2. \qquad (3.52)
\end{aligned}
$$

Note that $E(\mathbf{x},\mathbf{c})$ is continuous in $\mathbf{c}$ and measurable in $\mathbf{x}$. Let now $\mathbf{X}_n$ denote i.i.d. random vectors with distribution $p_{\mathbf{X}}$. Using Theorem [52, Thm. 2] we get, with probability one,

$$
\lim_{N\to\infty} \sup_{\mathbf{c}\in C_m} \left| \frac{1}{N}\sum_{n=1}^{N} E(\mathbf{X}_n,\mathbf{c})^2 - \int_{\mathscr{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} \right| = 0. \qquad (3.53)
$$

Since almost sure convergence implies convergence in probability [55, Ch. 2], we also have:

$$
\lim_{N\to\infty} P\left(\sup_{\mathbf{c}\in C_m} \left| \frac{1}{N}\sum_{n=1}^{N} E(\mathbf{X}_n,\mathbf{c})^2 \; - \int_{\mathscr{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} \right| > \epsilon \right) = 0 \;\forall \epsilon > 0. \qquad (3.54)
$$

We will need this result when considering the case with noise. For the case with noise, i.e. $y_n = f(\mathbf{x}_n) + \eta_n$, let

$$
\begin{aligned}
\tilde{E}(\mathbf{x},\eta,\mathbf{c})^2 &= \left(f(\mathbf{x}) + \eta - \sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k)\right)^2 \\
&= E(\mathbf{x},\mathbf{c})^2 + 2\eta E(\mathbf{x},\mathbf{c}) + \eta^2. \qquad (3.55)
\end{aligned}
$$

Using the properties of the noise η with p.d.f. p_H, this gives the following mean square error:

$$
\begin{aligned}
&\int_{\mathbb{R}}\int_{\mathscr{X}} \tilde{E}(\mathbf{x},\eta,\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\, p_H(\eta)\, d\mathbf{x} d\eta \\
&= \int_{\mathscr{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x}) \left(\int_{\mathbb{R}} p_H(\eta)\, d\eta\right) d\mathbf{x} \\
&\quad + 2\int_{\mathscr{X}} E(\mathbf{x},\mathbf{c}) \left(\int_{\mathbb{R}} \eta\, p_H(\eta)\, d\eta\right) p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} \\
&\quad + \int_{\mathscr{X}} p_{\mathbf{X}}(\mathbf{x}) \left(\int_{\mathbb{R}} \eta^2 p_H(\eta)\, d\eta\right) d\mathbf{x} \\
&= \int_{\mathscr{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} + \int_{\mathscr{X}} E(\mathbf{x},\mathbf{c}) \underbrace{\mathbb{E}[H_n]}_{=0} p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} + \mathbb{E}[H_n^2]
\end{aligned}
$$

$$= \int_{\mathscr{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\,d\mathbf{x} + \sigma_H^2. \tag{3.56}$$

Here, H_n is a random variable with distribution p_H. For any choice of $\epsilon_0, \epsilon_1, \epsilon_2, \epsilon_3 > 0$ such that $\epsilon_1 + \epsilon_2 + \epsilon_3 = \epsilon_0$, we have, following a similar proof as in [56, Thm. 3.3(a)]:

$$\begin{aligned}
&P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N}\tilde{E}(\mathbf{X}_n,H_n,\mathbf{c})^2 - \int_{\mathscr{X}}\int_{\mathbb{R}}\tilde{E}(\mathbf{x},\eta,\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\,p_H(\eta)\,d\mathbf{x}d\eta\right| > \epsilon_0\right)\\
&= P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N}E(\mathbf{X}_n,\mathbf{c})^2 + \frac{2}{N}\sum_{n=1}^{N}H_n E(\mathbf{X}_n,\mathbf{c})\right.\right.\\
&\qquad\left.\left. + \frac{1}{N}\sum_{n=1}^{N}H_n^2 - \int_{\mathscr{X}}E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\,d\mathbf{x} - \sigma_H^2\right| > \epsilon_0\right)\\
&\le P\left(\sup_{\mathbf{c}\in C_m}\left\{\left|\frac{1}{N}\sum_{n=1}^{N}E(\mathbf{X}_n,\mathbf{c})^2 - \int_{\mathscr{X}}E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\,d\mathbf{x}\right|\right.\right.\\
&\quad\left.\left. + \left|\frac{2}{N}\sum_{n=1}^{N}H_n E(\mathbf{X}_n,\mathbf{c})\right| + \left|\frac{1}{N}\sum_{n=1}^{N}H_n^2 - \sigma_H^2\right|\right\} > \epsilon_0\right)\\
&\le P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N}E(\mathbf{X}_n,\mathbf{c})^2 - \int_{\mathscr{X}}E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\,d\mathbf{x}\right|\right.\\
&\quad\left. + \sup_{\mathbf{c}\in C_m}\left|\frac{2}{N}\sum_{n=1}^{N}H_n E(\mathbf{X}_n,\mathbf{c})\right| + \left|\frac{1}{N}\sum_{n=1}^{N}H_n^2 - \sigma_H^2\right| > \epsilon_0\right)\\
&\le P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N}E(\mathbf{X}_n,\mathbf{c})^2 - \int_{\mathscr{X}}E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\,d\mathbf{x}\right| > \epsilon_1\right.\\
&\qquad \text{or } \sup_{\mathbf{c}\in C_m}\left|\frac{2}{N}\sum_{n=1}^{N}H_n E(\mathbf{X}_n,\mathbf{c})\right| > \epsilon_2\\
&\qquad \left.\text{or } \left|\frac{1}{N}\sum_{n=1}^{N}H_n^2 - \sigma_H^2\right| > \epsilon_3\right)\\
&\le P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N}E(\mathbf{X}_n,\mathbf{c})^2 - \int_{\mathscr{X}}E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})\,d\mathbf{x}\right| > \epsilon_1\right)\\
&\quad + P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{2}{N}\sum_{n=1}^{N}H_n E(\mathbf{X}_n,\mathbf{c})\right| > \epsilon_2\right)\\
&\quad + P\left(\left|\frac{1}{N}\sum_{n=1}^{N}H_n^2 - \sigma_H^2\right| > \epsilon_3\right).
\end{aligned}$$

Of these last three probabilities, the first one is proven to converge to zero in (3.54), while the last one converges to zero by the weak law of large numbers. For the second probability, we can make use of Theorem [52, Thm. 2] again, noting that $\eta_n E(\mathbf{x}_n,\mathbf{c})$ is continuous in $\mathbf{c}$. We use (3.52) to get

$$\left|\eta E(\mathbf{x},\mathbf{c})\right| \le |\eta|\left(||f||_\infty + m\right)\ \forall \mathbf{x},\eta,\mathbf{c}. \tag{3.57}$$

Again, since uniform convergence implies convergence in probability, and since $\mathbb{E}[H_n E(\mathbf{X}_n, \mathbf{c})] = \mathbb{E}[H_n]\mathbb{E}[E(\mathbf{X}_n, \mathbf{c})] = 0$ for all n, using Theorem [52, Thm. 2] gives the desired convergence in probability

$$\lim_{N\to\infty} P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N} H_n E(\mathbf{X}_n, \mathbf{c})\right| > \epsilon_2\right) = 0 \ \forall \epsilon_2. \tag{3.58}$$

Together with the other two convergences and (3.57) we get:

$$\lim_{N\to\infty} P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N} \tilde{E}(\mathbf{X}_n, H_n, \mathbf{c})^2 - \int_{\mathbb{R}}\int_{\mathscr{X}} \tilde{E}(\mathbf{x}, \eta, \mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta\right| > \epsilon\right) = 0. \tag{3.59}$$

The following bound follows from (3.52) and (3.56):

$$\begin{aligned} 0 &\le \int_{\mathbb{R}}\int_{\mathscr{X}} \tilde{E}(\mathbf{x}, \eta, \mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta \\ &\le \left(||f||_\infty + m\right)^2 + \sigma_H^2. \end{aligned} \tag{3.60}$$

In light of this bound, [53, Key Thm.] now implies that the mean square error between the output of the RFE with least squares weight vector and the noisy measurements is approaching its ideal value as the number of samples increases. More precisely, for any choice of $\epsilon_4 > 0$ and $\delta_1 > 0$, there exists an N_0 such that, for all $N > N_0$,

$$\left|\int_{\mathbb{R}}\int_{\mathscr{X}} \tilde{E}(\mathbf{x}, \eta, \mathbf{C_N})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta - \int_{\mathbb{R}}\int_{\mathscr{X}} \tilde{E}(\mathbf{x}, \eta, \mathbf{C^0})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta\right| < \epsilon_4 \tag{3.61}$$

with probability at least $1 - \delta_1$. Here, $\mathbf{C_N}$ denotes the vector $\mathbf{c_N}$ as a random variable as it depends on the input and noise samples and on the samples $\omega_1, \ldots, \omega_D, b_1, \ldots, b_D$, and $\mathbf{C^0} \in C_m$ minimizes $\int_{\mathbb{R}}\int_{\mathscr{X}} \tilde{E}(\mathbf{x}, \eta, \mathbf{c}) p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta$. Next, it is shown that the same holds for the mean square error between the least-squares RFE outputs and the unknown, noise-free function values.

According to [43, Thm 3.2], for any $\delta_2 > 0$, with probability at least $1-\delta_2$ w.r.t. $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_D$ and $\mathbf{b}_1, \ldots, \mathbf{b}_D$, there exists a $\mathbf{c} \in C_m$ with the following bound[1]:

$$\int_{\mathscr{X}} \left(f(\mathbf{x}) - \sum_{k=1}^{D} c_k \cos(\mathbf{\Omega}_k^T \mathbf{x} + \mathbf{b}_k)\right)^2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} < \frac{\gamma(\delta_2)^2}{D},$$

$$\gamma(\delta_2) = \sup_{\omega, b}\left|\frac{1}{(2\pi)^d}\frac{\bar{c}(\omega, b)}{p_{\mathbf{\Omega}}(\omega) p_{\mathbf{b}}(b)}\right|\left(\sqrt{\log\frac{1}{\delta_2}} + 4r\right),$$

[1] The weights found in the proof of the cited theorem satisfy $\mathbf{c} \in C_m$ if $m \ge M$, which was shown in the beginning of this appendix. Here we also made use of the result from Theorem 6 of this paper to get what is denoted with α in [43]. We have also used, with the notation of [43], that $||f - \hat{f}||_\mu \le ||f - \hat{f}||_\infty$.

$$r = \sup_{\mathbf{x}\in\mathscr{X}} ||\mathbf{x}||_2 \sqrt{\sigma^2 d + \pi^2/3}, \tag{3.62}$$

with σ^2 denoting the variance of $p_{\mathbf{\Omega}}$. For this particular $\mathbf{c}$, (3.55), (3.56) and (3.62) imply that

$$\int_{\mathbb{R}}\int_{\mathscr{X}} \tilde{E}(\mathbf{x},\eta,\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta < \frac{\gamma(\delta_2)^2}{D} + \sigma_H^2. \tag{3.63}$$

Since $\mathbf{C^0} \in C_m$ minimizes the left-hand in the equation above by definition, we also have that

$$\int_{\mathbb{R}}\int_{\mathscr{X}} \tilde{E}(\mathbf{x},\eta,\mathbf{C^0})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta < \frac{\gamma(\delta_2)^2}{D} + \sigma_H^2 \tag{3.64}$$

with probability at least $1-\delta_2$. Since the event in (3.64) only depends on $\mathbf{\Omega}_1,\ldots,\mathbf{\Omega}_D$ and $\mathbf{b}_1,\ldots,\mathbf{b}_D$, while the event in (3.61) only depends on the input and noise samples, we can combine these two equations as follows. For any choice of $\epsilon_4 > 0$, $\delta_1 > 0$ and $\delta_2 > 0$, there exists an N_0 such that, for all $N > N_0$,

$$\int_{\mathbb{R}}\int_{\mathscr{X}} \tilde{E}(\mathbf{x},\eta,\mathbf{C_N})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta < \epsilon_4 + \frac{\gamma(\delta_2)^2}{D} + \sigma_H^2 \tag{3.65}$$

with probability at least $(1-\delta_1)(1-\delta_2)$. Using (3.56) now gives the following result. For any choice of $\epsilon_4 > 0$, $\delta_1 > 0$ and $\delta_2 > 0$, there exists an N_0 such that, for all $N > N_0$, we have

$$\int_{\mathscr{X}} E(\mathbf{x},\mathbf{C_N})^2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} < \epsilon_4 + \frac{\gamma(\delta_2)^2}{D} \tag{3.66}$$

with probability at least $(1-\delta_1)(1-\delta_2)$.

Choosing $D_0, \epsilon_4, \delta_1$ and δ_2 such that $D_0 > \gamma(\delta_2)^2/(\epsilon-\epsilon_4)$ and $(1-\delta_1)(1-\delta_2) = \delta$ concludes the proof.

□

3.8. APPENDIX: MINIMUM-VARIANCE PROPERTIES

The following theorem presents the probability density function for $\mathbf{\Omega}_k$ that minimizes the variance of a RFE at a fixed measurement location $\mathbf{x}$.

Theorem 12. *Given* $\mathbf{x}$*, the p.d.f.* $p^*_{\mathbf{\Omega}}$ *that minimizes the variance of the unbiased estimator* $G(\mathbf{x}) = \sum_{k=1}^{D} C_k \cos(\mathbf{\Omega}_k^T\mathbf{x} + \mathbf{b}_k)$ *as defined in Theorem 1, with* C_k *as defined in Theorem 8, is equal to*

$$p^*_{\mathbf{\Omega}}(\omega) = \frac{|\hat{f}(\omega)|\sqrt{\cos(2\angle\hat{f}(\omega) + 2\omega^T\mathbf{x}) + 2}}{\int_{\mathbb{R}^d} |\hat{f}(\tilde{\omega})|\sqrt{\cos(2\angle\hat{f}(\tilde{\omega}) + 2\tilde{\omega}^T\mathbf{x}) + 2}\, d\tilde{\omega}}. \tag{3.67}$$

For this choice of $p_{\mathbf{\Omega}}$*, the variance is equal to*

$$\frac{1}{2D(2\pi)^{2d}}\left(\int_{\mathbb{R}^d} |\hat{f}(\omega)|\sqrt{\cos(2\angle\hat{f}(\omega) + 2\omega^T\mathbf{x}) + 2}\, d\omega\right)^2 - f(\mathbf{x})^2. \tag{3.68}$$

Proof. The proof is similar to the proof of [49, Thm. 4.3.1]. Let $q_{\mathbf{\Omega}}$ be any p.d.f. of $\mathbf{\Omega}_k$ that satisfies $q_{\mathbf{\Omega}}(\omega) > 0$ if $|\hat{f}(\omega)| > 0$. Let $\mathrm{Var}_{q_{\mathbf{\Omega}}, p_{\mathbf{b}}}$ be the variance of $G(\mathbf{x})$ under the assumption that $p_{\mathbf{\Omega}} = q_{\mathbf{\Omega}}$, $p_{\mathbf{b}} = \mathrm{Uniform}(0, 2\pi)$, and $C_k = \frac{2}{D(2\pi)^d} \frac{|\hat{f}(\mathbf{\Omega}_k)|}{q_{\mathbf{\Omega}}(\mathbf{\Omega}_k)} \cos(\angle \hat{f}(\mathbf{\Omega}_k) - \mathbf{b}_k)$. According to Theorem 8, this choice for C_k makes sure that $G(\mathbf{x})$ is an unbiased estimator, i.e., $f(\mathbf{x}) = \mathbb{E}[G(\mathbf{x})]$. The variance of $G(\mathbf{x})$ can be computed as:

$$
\begin{aligned}
&\mathrm{Var}_{q_{\mathbf{\Omega}}, p_{\mathbf{b}}}[G(\mathbf{x})] \\
&= \mathrm{Var}_{q_{\mathbf{\Omega}}, p_{\mathbf{b}}}\left[\sum_{k=1}^{D} C_k \cos(\mathbf{\Omega}_k^T \mathbf{x} + B_k)\right] \\
&= D\, \mathrm{Var}_{q_{\mathbf{\Omega}}, p_{\mathbf{b}}}\left[C_1 \cos(\mathbf{\Omega}_1^T \mathbf{x} + B_1)\right] \\
&= \frac{D}{2\pi} \int_{\mathbb{R}^d} \int_0^{2\pi} \left(\frac{2}{D(2\pi)^d} \frac{|\hat{f}(\omega)|}{q_{\mathbf{\Omega}}(\omega)} \cos(\angle \hat{f}(\omega) - b)\right)^2 \\
&\qquad \cos(\omega^T \mathbf{x} + b)^2 q_{\mathbf{\Omega}}(\omega)\, db\, d\omega - f(\mathbf{x})^2.
\end{aligned} \tag{3.69}
$$

For the stated choice of $p^*_{\mathbf{\Omega}}$, using

$$
\begin{aligned}
&\int_0^{2\pi} \cos(\angle \hat{f}(\omega) - b)^2 \cos(\omega^T \mathbf{x} + b)^2\, db \\
&= \int_0^{2\pi} \frac{1}{4}(1 + \cos(2\angle \hat{f}(\omega) - 2b))(1 + \cos(2\omega^T \mathbf{x} + 2b))\, db \\
&= \int_0^{2\pi} \frac{1}{4} db + \frac{1}{4} \int_0^{2\pi} \cos(2\angle \hat{f}(\omega) - 2b)\, db \\
&\quad + \frac{1}{4} \int_0^{2\pi} \cos(2\omega^T \mathbf{x} + 2b)\, db \\
&\quad + \frac{1}{4} \int_0^{2\pi} \cos(2\angle \hat{f}(\omega) - 2b) \cos(2\omega^T \mathbf{x} + 2b)\, db \\
&= \frac{2\pi}{4} + \frac{1}{8} \int_0^{2\pi} \cos(2\angle \hat{f}(\omega) + 2\omega^T \mathbf{x}) \\
&\quad + \cos(2\angle \hat{f}(\omega) - 2\omega^T \mathbf{x} - 4b)\, db \\
&= \frac{2\pi}{4} + \frac{2\pi}{8} \cos(2\angle \hat{f}(\omega) + 2\omega^T \mathbf{x}) \\
&= \frac{\pi}{4}(\cos(2\angle \hat{f}(\omega) + 2\omega^T \mathbf{x}) + 2)
\end{aligned} \tag{3.70}
$$

we get:

$$
\begin{aligned}
&\mathrm{Var}_{p^*_{\mathbf{\Omega}}, p_{\mathbf{b}}}[G(\mathbf{x})] + f(\mathbf{x})^2 = \mathbb{E}_{p^*_{\mathbf{\Omega}}, p_{\mathbf{b}}}[G(\mathbf{x})^2] \\
&= \frac{D}{2\pi} \int_{\mathbb{R}^d} \int_0^{2\pi} \left(\frac{2}{D(2\pi)^d} \frac{|\hat{f}(\omega)|}{p^*_{\mathbf{\Omega}}(\omega)} \cos(\angle \hat{f}(\omega) - b)\right)^2 \\
&\qquad \cos(\omega^T \mathbf{x} + b)^2 p^*_{\mathbf{\Omega}}(\omega)\, db\, d\omega
\end{aligned}
$$

$$= \frac{D}{2\pi}\int_{\mathbb{R}^d}\frac{1}{p^*_{\mathbf{\Omega}}(\omega)}\left(\frac{2}{D(2\pi)^d}\right)^2|\hat{f}(\omega)|^2 \int_0^{2\pi}\cos(\angle\hat{f}(\omega)-b)^2\cos(\omega^T\mathbf{x}+b)^2\,db\,d\omega$$

$$= \frac{D}{2\pi}\int_{\mathbb{R}^d}\frac{1}{p^*_{\mathbf{\Omega}}(\omega)}\left(\frac{2}{D(2\pi)^d}\right)^2|\hat{f}(\omega)|^2 \frac{\pi}{4}(\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)\,d\omega \tag{3.71}$$

$$\overset{(3.67)}{=} \frac{D}{2\pi}\left(\frac{2}{D(2\pi)^d}\right)^2\left(\int_{\mathbb{R}^d}|\hat{f}(\omega)|\sqrt{\frac{\pi}{4}(\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)}\,d\omega\right)^2$$

$$= \frac{1}{2D(2\pi)^{2d}}\left(\int_{\mathbb{R}^d}|\hat{f}(\omega)|\sqrt{(\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)}\,d\omega\right)^2 \tag{3.72}$$

This gives the value of the optimal variance. To show that the variance is indeed optimal, compare it with any arbitrary p.d.f. $q_{\mathbf{\Omega}}$ using Jensen's inequality:

$$\begin{aligned}
&\mathrm{Var}_{p^*_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})]+f(\mathbf{x})^2\\
&= \frac{D}{2\pi}\left(\frac{2}{D(2\pi)^d}\right)^2\left(\int_{\mathbb{R}^d}\frac{|\hat{f}(\omega)|}{q_{\mathbf{\Omega}}(\omega)}\sqrt{\frac{\pi}{4}(\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)}\,q_{\mathbf{\Omega}}(\omega)\,d\omega\right)^2\\
&\overset{\text{Jensen}}{\leq} \frac{D}{2\pi}\left(\frac{2}{D(2\pi)^d}\right)^2\int_{\mathbb{R}^d}\frac{|\hat{f}(\omega)|^2}{q_{\mathbf{\Omega}}(\omega)^2}\frac{\pi}{4}(\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)\,q_{\mathbf{\Omega}}(\omega)\,d\omega\\
&\overset{(3.70)}{=} \frac{D}{2\pi}\int_{\mathbb{R}^d}\int_0^{2\pi}\left(\frac{2}{D(2\pi)^d}\frac{|\hat{f}(\omega)|}{q_{\mathbf{\Omega}}(\omega)}\cos(\angle\hat{f}(\omega)-b)\right)^2\cos(\omega^T\mathbf{x}+b)^2\,q_{\mathbf{\Omega}}(\omega)\,db\,d\omega\\
&\overset{(3.69)}{=} \mathrm{Var}_{q_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})]+f(\mathbf{x})^2.
\end{aligned} \tag{3.73}$$

This shows that the chosen p.d.f. $p^*_{\mathbf{\Omega}}$ gives the minimum variance.

□

The following theorem compares the second moments in real and complex RFEs for different probability distributions.

Theorem 13. *Let $\tilde{p}_{\mathbf{\Omega}}$, $p^*_{\mathbf{\Omega}}$, $\tilde{G}$ and G be as in Theorems 10 and 12. Then*

$$\frac{1}{\sqrt{3}}\mathbb{E}_{p^*_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})^2] \leq \mathbb{E}_{\tilde{p}_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})^2] \leq \sqrt{3}\,\mathbb{E}_{p^*_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})^2], \tag{3.74}$$

$$\frac{1}{2}\mathbb{E}_{\tilde{p}_{\mathbf{\Omega}},p_{\mathbf{b}}}[\tilde{G}(\mathbf{x})^2] \leq \mathbb{E}_{\tilde{p}_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})^2] \leq \frac{3}{2}\,\mathbb{E}_{\tilde{p}_{\mathbf{\Omega}},p_{\mathbf{b}}}[\tilde{G}(\mathbf{x})^2]. \tag{3.75}$$

Proof. From

$$1 \leq \sqrt{(\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)} \leq \sqrt{3}, \tag{3.76}$$

and from (3.67) and (3.33) it follows that

$$\frac{1}{\sqrt{3}}p^*_{\mathbf{\Omega}}(\omega) \leq \tilde{p}_{\mathbf{\Omega}}(\omega) \leq \sqrt{3}p^*_{\mathbf{\Omega}}(\omega),$$

$$\frac{1}{\sqrt{3}}\frac{1}{p^*_{\mathbf{\Omega}}(\omega)} \leq \frac{1}{\tilde{p}_{\mathbf{\Omega}}(\omega)} \leq \sqrt{3}\frac{1}{p^*_{\mathbf{\Omega}}(\omega)}. \tag{3.77}$$

Combining the above with (3.71) yields:

$$\begin{aligned}
&\frac{1}{\sqrt{3}}\mathbb{E}_{p^*_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})^2] \\
&= \frac{1}{\sqrt{3}}\frac{1}{2D(2\pi)^{2d}}\int_{\mathbb{R}^d}\frac{1}{p^*_{\mathbf{\Omega}}(\omega)}|\hat{f}(\omega)|^2(\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)d\omega \\
&\leq \frac{1}{2D(2\pi)^{2d}}\int_{\mathbb{R}^d}\frac{1}{\tilde{p}_{\mathbf{\Omega}}(\omega)}|\hat{f}(\omega)|^2(\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)d\omega \\
&= \mathbb{E}_{\tilde{p}_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})^2] \\
&\leq \sqrt{3}\frac{1}{2D(2\pi)^{2d}}\int_{\mathbb{R}^d}\frac{1}{p^*_{\mathbf{\Omega}}(\omega)}|\hat{f}(\omega)|^2(\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)d\omega \\
&= \sqrt{3}\,\mathbb{E}_{p^*_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})].
\end{aligned} \tag{3.78}$$

Combining (3.76) with (3.34) yields:

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}_{\tilde{p}_{\mathbf{\Omega}}}[\tilde{G}(\mathbf{x})^2] \\
&= \frac{1}{2D(2\pi)^{2d}}\int_{\mathbb{R}^d}\frac{1}{\tilde{p}_{\mathbf{\Omega}}(\omega)}|\hat{f}(\omega)|^2 d\omega \\
&\leq \frac{1}{2D(2\pi)^{2d}}\int_{\mathbb{R}^d}\frac{1}{\tilde{p}_{\mathbf{\Omega}}(\omega)}|\hat{f}(\omega)|^2 \\
&\qquad (\cos(2\angle\hat{f}(\omega)+2\omega^T\mathbf{x})+2)d\omega \\
&= \mathbb{E}_{\tilde{p}_{\mathbf{\Omega}},p_{\mathbf{b}}}[G(\mathbf{x})^2] \\
&\leq \frac{3}{2D(2\pi)^{2d}}\int_{\mathbb{R}^d}\frac{1}{\tilde{p}_{\mathbf{\Omega}}(\omega)}|\hat{f}(\omega)|^2 d\omega \\
&= \frac{3}{2}\mathbb{E}_{\tilde{p}_{\mathbf{\Omega}}}[\tilde{G}(\mathbf{x})^2].
\end{aligned} \tag{3.79}$$

□

REFERENCES

[1] L. Bliek, H. R. G. W. Verstraete, M. Verhaegen, and S. Wahls, *Online optimization with costly and noisy measurements using random Fourier expansions,* IEEE Transactions on Neural Networks and Learning Systems **29**, 167 (2018).

[2] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, Vol. 8 (Siam, 2009).

[3] L. M. Rios and N. V. Sahinidis, *Derivative-free optimization: a review of algorithms and comparison of software implementations,* J. Global Optim. **56**, 1247 (2013).

[4] J. A. Nelder and R. Mead, *A simplex method for function minimization,* Comput. J. **7**, 308 (1965).

[5] M. J. Powell, *The NEWUOA software for unconstrained optimization without derivatives,* in *Large-scale nonlinear optimization* (Springer, 2006) pp. 255–297.

[6] M. J. Powell, *The BOBYQA algorithm for bound constrained optimization without derivatives,* Cambridge NA Report NA2009/06, University of Cambridge, Cambridge , 26 (2009).

[7] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, *Lipschitzian optimization without the Lipschitz constant,* J. Optimiz. Theory App. **79**, 157 (1993).

[8] P. Gilmore and C. T. Kelley, *An implicit filtering algorithm for optimization of functions with many local minima,* SIAM J. Optimiz. **5**, 269 (1995).

[9] A. L. Custódio and L. N. Vicente, *Using sampling and simplex derivatives in pattern search methods,* SIAM J. Optimiz. **18**, 537 (2007).

[10] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient global optimization of expensive black-box functions,* J. Global Optim. **13**, 455 (1998).

[11] D. Kbiob, *A statistical approach to some basic mine valuation problems on the Witwatersrand,* Journal of Chemical, Metallurgical, and Mining Society of South Africa (1951).

[12] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, *Algorithms for hyper-parameter optimization,* in *Adv. Neur. In.* (2011) pp. 2546–2554.

[13] F. Hutter, H. H. Hoos, and K. Leyton-Brown, *Sequential model-based optimization for general algorithm configuration,* in *LION* (Springer, 2011) pp. 507–523.

[14] R. Martinez-Cantin, *BayesOpt: a Bayesian optimization library for nonlinear optimization, experimental design and bandits,* J. Mach. Learn. Res. **15**, 3735 (2014).

[15] O. Roustant, D. Ginsbourger, and Y. Deville, *Dicekriging, Diceoptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization,* Journal of Statistical Software **51** (2012).

[16] J. Snoek, H. Larochelle, and R. P. Adams, *Practical Bayesian optimization of machine learning algorithms,* in *Adv. Neur. In.* (2012) pp. 2951–2959.

[17] E. Brochu, V. M. Cora, and N. de Freitas, *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning,* arXiv e-prints , arXiv:1012.2599 (2010), arXiv:1012.2599 [cs.LG] .

[18] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, *A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot,* Autonomous Robots **27**, 93 (2009).

[19] S. ur Rehman and M. Langelaar, *Efficient global robust optimization of unconstrained problems affected by parametric uncertainties,* Struct. Multidiscip. O. , 1 (2015).

[20] H. R. G. W. Verstraete, S. Wahls, J. Kalkman, and M. Verhaegen, *Model-based sensorless wavefront aberration correction in optical coherence tomography,* Opt. Lett. **40**, 5722 (2015).

[21] A. Rahimi and B. Recht, *Random features for large-scale kernel machines,* in *Adv. Neur. In.* (2007) pp. 1177–1184.

[22] M.-R. Nasiri-Avanaki, S. Hojjatoleslami, H. Paun, S. Tuohy, A. Meadway, G. Dobre, and A. Podoleanu, *Optical coherence tomography system optimization using simulated annealing algorithm,* Proce. of Math. Meth. and Appl. Comp.,(WSEAS, 2009) , 669 (2009).

[23] S. Bonora and R. Zawadzki, *Wavefront sensorless modal deformable mirror correction in adaptive optics: optical coherence tomography,* Opt. Lett. **38**, 4801 (2013).

[24] R. C. Hansen, *Phased array antennas*, Vol. 213 (John Wiley & Sons, 2009).

[25] C. Roeloffzen, L. Zhuang, R. Heideman, A. Borreman, and v. W. Etten, *Ring resonator-based tunable optical delay line in LPCVD waveguide technology,* in *Proceedings Symposium IEEE/LEOS Benelux Chapter* (IEEE, 2005) pp. 79–82.

[26] A. Meijerink, C. G. Roeloffzen, R. Meijerink, L. Zhuang, D. A. Marpaung, M. J. Bentum, M. Burla, J. Verpoorte, P. Jorna, A. Hulzinga, *et al., Novel ring resonator-based integrated photonic beamformer for broadband phased array receive antennas—part I: Design and performance analysis,* J. Lightwave Technol. **28**, 3 (2010).

[27] L. Zhuang, C. Roeloffzen, R. Heideman, A. Borreman, A. Meijerink, and W. Van Etten, *Single-chip optical beam forming network in LPCVD waveguide technology based on optical ring resonators,* in *International Topical Meeting on Microwave Photonics, 2006. MWP'06.* (IEEE, 2006) pp. 1–4.

[28] L. Zhuang, *Ring resonator-based broadband photonic beam former for phased array antennas*, Ph.D. thesis, University of Twente (2010).

[29] L. Bliek, M. Verhaegen, and S. Wahls, *Data-driven minimization with random feature expansions for optical beam forming network tuning,* 16th IFAC Workshop on Control Applications of Optimization (CAO'2015) **48**, 166 (2015).

[30] J. de Lope, M. Santos, *et al., A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers,* Neurocomputing **72**, 2806 (2009).

[31] T. Hofmann, B. Schölkopf, and A. J. Smola, *Kernel methods in machine learning,* Ann. Stat. , 1171 (2008).

[32] J. A. Suykens and J. P. Vandewalle, *Nonlinear Modeling: advanced black-box techniques* (Springer Science & Business Media, 2012).

[33] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective* (Academic Press, 2015).

[34] A. Rahimi and B. Recht, *Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning,* in *Adv. Neur. In.* (2009) pp. 1313–1320.

[35] A. Singh, N. Ahuja, and P. Moulin, *Online learning with kernels: Overcoming the growing sum problem,* in *2012 IEEE International Workshop on Machine Learning for Signal Processing* (IEEE, 2012) pp. 1–6.

[36] C. J. Burges *et al.*, *Simplified support vector decision rules,* in *ICML*, Vol. 96 (Citeseer, 1996) pp. 71–77.

[37] D. Schölkopf, *Sampling techniques for kernel methods,* in *Adv. Neur. In.*, Vol. 1 (MIT Press, 2002) p. 335.

[38] J. Quinonero-Candela and C. E. Rasmussen, *A unifying view of sparse approximate Gaussian process regression,* J. Mach. Learn. Res. **6**, 1939 (2005).

[39] B. Chen, S. Zhao, P. Zhu, and J. C. Principe, *Quantized kernel recursive least squares algorithm,* IEEE Trans. Neural Netw. Learn. Syst. **24**, 1484 (2013).

[40] L. Zhang and P. Suganthan, *A comprehensive evaluation of random vector functional link networks,* Information Sciences (2015).

[41] F. Girosi and G. Anzellotti, *Convergence rates of approximation by translates*, Tech. Rep. (DTIC Document, 1992).

[42] A. R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function,* IEEE Trans. Inf. Theory **39**, 930 (1993).

[43] A. Rahimi and B. Recht, *Uniform approximation of functions with random bases,* in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on* (IEEE, 2008) pp. 555–561.

[44] L. K. Jones, *A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training,* Ann. Stat. , 608 (1992).

[45] A. H. Sayed and T. Kailath, *Recursive least-squares adaptive filters,* Digit. Signal Process. Handbook , 21 (1998).

[46] J. Nocedal, *Updating quasi-Newton matrices with limited storage,* Math. Comp. **35**, 773 (1980).

[47] J. Nocedal and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

[48] M. Pogu and J. S. De Cursi, *Global optimization by random perturbation of the gradient method with a fixed parameter,* J. of Global Optim. **5**, 159 (1994).

[49] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*, Vol. 707 (John Wiley & Sons, 2011).

[50] H. R. G. W. Verstraete, S. Wahls, J. Kalkman, and M. Verhaegen, *Numerical evaluation of advanced optimization algorithms for wavefront aberration correction in OCT,* in *Imaging and Applied Optics 2015* (OSA, 2015) p. AOM3F.3.

[51] H. R. G. W. Verstraete, B. Cense, R. Bilderbeek, M. Verhaegen, and J. Kalkman, *Towards model-based adaptive optics optical coherence tomography,* Opt. Express **22**, 32406 (2014).

[52] R. I. Jennrich, *Asymptotic properties of non-linear least squares estimators,* Ann. Math. Stat. , 633 (1969).

[53] V. N. Vapnik, *An overview of statistical learning theory,* IEEE Trans. Neural Netw. **10**, 988 (1999).

[54] G. H. Golub and C. F. Van Loan, *Matrix computations*, Vol. 3 (JHU Press, 2012).

[55] A. W. Van der Vaart, *Asymptotic statistics*, Vol. 3 (Cambridge university press, 2000).

[56] A. Beitollahi and P. Azhdari, *Convergence in probability and almost surely convergence in probabilistic normed spaces,* Math. Sci. **6**, 1 (2012).

4

AUTOMATIC TUNING OF A RING RESONATOR-BASED PHOTONIC BEAMFORMER FOR A PHASED ARRAY TRANSMIT ANTENNA

We present a novel photonic beamformer for a fully integrated transmit phased array antenna, together with an automatic procedure for tuning this system. Such an automatic tuning procedure is required because the large number of actuators makes manual tuning practically impossible. The antenna system is designed for the purpose of broadband aircraft-satellite communication in the K_u*-band to provide satellite Internet connections on board the aircraft. The goal of the beamformer is to automatically steer the transmit antenna electronically in the direction of the satellite. This is done using a mix of phase shifters and tunable optical delay lines, which are all integrated on a chip and laid out in a tree structure.*

The K_u*-band has a bandwidth of* 0.5 *GHz. We show how an optical delay line is automatically configured over this bandwidth, providing a delay of approximately* 0.4 *ns. The tuning algorithm calculates the best actuator voltages based on past measurements. This is the first time that such an automatic tuning scheme is used on a photonic beamformer for this type of transmit phased array antenna. We show that the proposed method is able to provide accurate beamforming (*$< 11.25°$ *phase error over the whole bandwidth) for two different delay settings.*

Parts of this chapter have been published in [1].

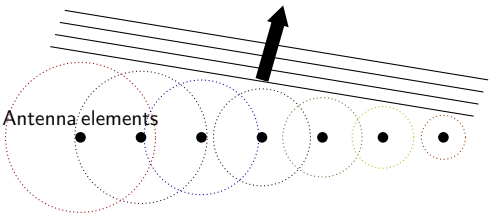


Figure 4.1: Beamforming for a phased array antenna explained. Each antenna element transmits the same signal after a certain time delay. This delay can be chosen in such a way that constructive interference occurs in a certain direction, making it possible to transmit a highly directional signal with a high gain and a focused beam.

4.1. INTRODUCTION

BEAMFORMERS are used to steer the beam of a phased array antenna by controlling the phase added to the signal at each antenna element (AE) [2]. This can be done by providing either a phase shift or a time delay to the signal. When the phase or delay of the signal corresponding to each AE has been set correctly according to the desired beam angle, signals are received or transmitted in the desired direction, while other directions are suppressed. This results in a highly directional antenna system for which the beam angle can be adapted without any mechanical movement. See Figure 4.1 for the case of a transmit phased array antenna. A receive antenna uses the same principle, but in this work we only consider a transmit antenna.

In order to get the same signal to each AE with a certain delay, a beamformer for a transmit phased array antenna consists of splitters and delay elements. These can be arranged in a tree structure to reduce the required number of delay elements [3–5]. The beamformer considered in this work is part of a transmit phased array antenna system with 1536 AEs. It uses a tree structure with four different splitting stages, as shown on the top of Figure 4.2. This system is designed to be used for aircraft-satellite communication in order to satisfy the ever-increasing demand of high-speed Internet on board of aircrafts. This large number of AEs is required to make sure that the signals have enough power when they arrive at the satellite [6, 7]. The system establishes an uplink with the satellite in the K_u-band (14.0 to 14.5 GHz frequency range). This is the range used to provide satellite Internet connections on board the aircraft. Therefore, the system will operate under a bandwidth of 0.5 GHz.

The delay that needs to be provided for one path of the tree structure depends on the distance between the AEs, which has been chosen to be 1.03 cm in this work as explained later. AEs that are close together correspond to only a small delay difference. This is why the middle of Figure 4.2 contains different types of beamformers: radio frequency (RF) beamformers are better suited for very small delay values, while photonic beamformers (explained in the Results section) are better suited for larger delay values [3]. The bottom of Figure 4.2 shows a top view of how the AEs are situated on the full antenna system that consists of 24 transmit tiles. The transmit tiles are 8.24 by 8.24 cm in size and contain 8×8 AEs each, resulting in a 1×1536 transmit scheme with 1536 AEs in total. Both the photonic and RF beamformers are integrated in the transmit tile, giving rise to a modular

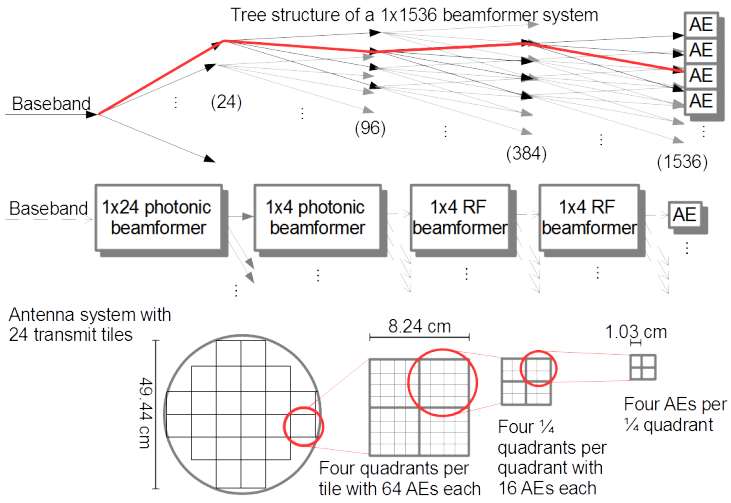


Figure 4.2: From top to bottom: tree structure of the full beamformer system with 1536 antenna elements (the highlighted path is explained in the remainder of the figure). Photonic and RF beamformer systems, where dashed arrows indicate signals in RF domain and solid arrows indicate signals in the optical domain (the photonic beamformers are shown in more detail in Figure 4.3). Full antenna system architecture with 24 transmit tiles containing 64 antenna elements each.

system.

The antenna system is an adapted version of an earlier proposed phased array receiver [7] that is to be used in transmit mode. In order to tune the beamformer actuators of these earlier phased array antenna systems, either a manual tuning procedure was used [3, 4], which is only possible for a small number of actuators, or a nonlinear optimization algorithm based on a physical model of the system was used [5, Sec. 6]. However, the approach that uses a nonlinear optimization algorithm is very sensitive to model errors [8], requires a labor-intensive measurement procedure for certain parts of the physical model [5, App. B], and can not be used while the system is already running. In this work we describe the beamformer for the transmit phased array antenna from Figure 4.2 together with its requirements, and we show how to tune this system with a different optimization algorithm. This online algorithm is not sensitive to model errors and can be used while the system is running.

4.2. FULLY INTEGRATED TRANSMIT PHASED ARRAY ANTENNA

4.2.1. BEAMFORMER REQUIREMENTS

In order to determine the requirements of the beamformer, the delay between the AEs needs to be calculated. This depends on two factors: the angle of the beam, and the distance between the AEs. For rectangular array grids, the latter should be equal to half the wavelength corresponding to the highest frequency used in the application [3, Sec. V],

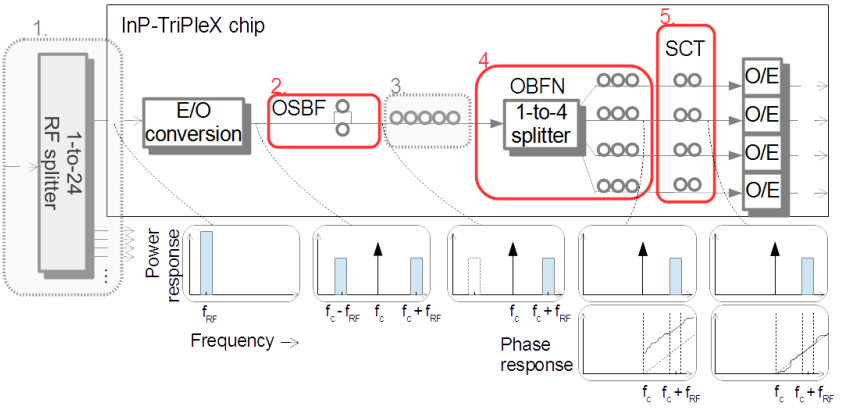


Figure 4.3: Photonic beamformer design with several highlighted subsystems. Subsystems 1. and 3. belong to the 1×24 photonic beamformer in Figure 4.2, while the other three subsystems belong to the 1 × 4 photonic beamformer. Dashed arrows indicate signals in RF domain and solid arrows indicate signals in the optical domain. The small circles indicate the presence of on-chip optical ring resonators. Power and phase responses of the signal at several locations on the chip are shown on the bottom, where the desired phase response is indicated by a dashed line.

which is 14.5 GHz in this case. This leads to a distance of 1.03 cm between two consecutive AEs. As for the beam angle, this is chosen to be no more than 60 degrees compared to the normal (i.e. a beam sent straight upwards). The maximum delay $\tau_{\max}$ that needs to be provided by each beamformer in Figure 4.2 can now be calculated as follows:

$$\tau_{\max} = \sin(60^\circ) d_{\max} / c_0, \tag{4.1}$$

where $d_{\max}$ is the maximum distance between the centers of the different elements of the beamformer and c_0 is the speed of light. For the rightmost RF beamformer in Figure 4.2, $d_{\max} = \sqrt{2} \cdot 0.0103$ m (distance between the centers of two diagonally intersecting AEs), for the other RF beamformer $d_{\max} = \sqrt{2} \cdot 2 \cdot 0.0103$ m, for the 1×4 photonic beamformer $d_{\max} = \sqrt{2} \cdot 4 \cdot 0.0103$ m, and for the 1×24 photonic beamformer $d_{\max} = \sqrt{5^2 + 1^2} \cdot 0.0824$ m. This leads to the following maximum delays for each beamformer in Figure 4.2: $\tau_{\max} = 42$ ps for the rightmost RF beamformer, $\tau_{\max} = 84$ ps for the other RF beamformer, $\tau_{\max} = 168$ ps for the 1×4 photonic beamformer, and $\tau_{\max} = 1.214$ ns for the other photonic beamformer.

After calculating the maximum delays for each beamformer subsystem, the requirements for the phase can be calculated. The phase response of each beamformer should be a linear function of the frequency, with a slope equal to $-2\pi\tau$, where τ is the required delay. Traditionally, phase shifters are used for beamformer subsystems, which provide a phase shift that is a constant function of the frequency. In other words, phase shifters provide a linear phase response with a slope of 0. For very small delays (e.g. $\tau_{\max} < 100$ ps in this case), using such a constant phase response with a slope of 0 instead of $-2\pi\tau$

gives a negligible slope mismatch, but for larger delays this mismatch will have a negative influence on the antenna beam. This problem is called beam squint.

The effects of beam squint are negligible in the two RF beamformer subsystems in Figure 4.2. Therefore, these beamformers are based on phase shifters that change the phase of the RF signal. The photonic beamformers, however, need to deal with larger delays. Therefore, they make use of optical delay lines rather than phase shifters. The key principle here is that the RF signals are converted to the optical domain, where their phase is adjusted accordingly, and then they are converted back to RF domain. This combination of RF and photonic signal processing has many advantages such as low loss, low weight, and large bandwidth, especially when implemented on a chip [9]. In this work, each transmit tile from Figure 4.2 is connected to a corresponding chip, as explained later on in this section.

The way the photonic beamformers provide a linear phase response with the correct slope is by using a cascade of optical ring resonators as tunable optical delay lines, arranged in a tree structure [3, 4, 10]. These ring resonators control the phase response of the signal by using thermo-optic heater actuators. Though some fluctuations in the phase response are allowed, these fluctuations should not be larger than 11.25° [4].

4.2.2. PHOTONIC BEAMFORMER CHIP DESIGN

Figure 4.3 shows the chip design of the photonic beamformers. The chip contains optical ring resonators that each belong to different subsystems. The effect of the different subsystems on the frequency response of the signal is shown as well. An RF splitter (subsystem 1. in Figure 4.3) makes sure that the same baseband signal arrives at all of the 24 transmit tiles shown in Figure 4.2. Each transmit tile is connected to one chip containing photonic and RF beamformer systems. The photonic beamformers use a combination of TriPleX™ waveguide technology [11] and indium phosphide (InP). TriPleX™ is a silicon nitride planar waveguide technology developed by LioniX International.

The corresponding power response of the signal entering the chip in Figure 4.3 is that of a broadband signal centered around the frequency f_{RF} = 14.25 GHz in the K_u-band. This signal is modulated on an optical carrier with a Mach-Zehnder modulator (electrical-to-optical conversion), after which the optical signal consists of the carrier f_c in the THz range and two sidebands. One of these sidebands is suppressed by the optical sideband filter (OSBF, subsystem 2. in Figure 4.3). This is done to reduce the required bandwidth[3]. After this stage the signal goes through a cascade of five ring resonators (subsystem 3. in Figure 4.3) that supply the required delay for one branch of the 1×24 beamformer. As mentioned earlier, the maximum delay that needs to be provided by the 1×24 beamformer is $\tau_{\max} = 1.214$ ns. This should be no problem for a cascade of five ring resonators, as beamforming with a cascade of only three ring resonators has been illustrated for delays of more than 1.2 ns with a bandwidth of 0.5 GHz using a manual tuning approach [12].

The signal then arrives at a 1×4 optical beamforming network (OBFN, susbystem 4. in Figure 4.3), consisting of an optical splitter and more ring resonators. These ring resonators provide a linear phase response for a maximum delay of 168 ps, as calculated earlier in this section. As seen in the phase response in Figure 4.3, the linear phase response is only provided in the region of interest, the sideband around frequency $f_c + f_{RF}$.

However, the OBFN only makes sure that this linear phase response has the correct slope, not the correct absolute phase. The last subsystem (subsystem 5. in Figure 4.3) provides an additional phase shift so that the correct phase response is provided at the sideband and also at the carrier frequency f_c. This is again done with optical ring resonators, using a principle called separate carrier tuning (SCT) [13]. Finally, the signal is converted back to RF domain with photodetectors (optical-to-electrical conversion).

The ring resonators in the beamformer subsystems are all actuated with two heaters: one for the phase, and one for the tunable coupling [12, 14]. By changing the voltages of these two heater actuators for several ring resonators, the magnitude and phase response of the subsystems can be altered. There are 27 ring resonators shown in Figure 4.3, resulting in 54 heaters. Three other heaters in the OSBF subsystem and three heaters in the 1-to-4 splitter are not shown in the figure but increase the number of heaters to 60. So the total number of heaters in the two photonic beamformer stages is $24 \times 60 = 1440$. One of the main reasons for investigating automatic procedures is that tuning all these heaters correctly by hand would not be practical.

4.3. AUTOMATIC TUNING RESULTS

4.3.1. AUTOMATIC TUNING METHOD

In this paper we look at an automatic tuning procedure for one branch of the OBFN subsystem. The traditional approach to automatically tune one branch of the photonic beamformer is to use a nonlinear optimization algorithm that minimizes a performance metric [5, Sec. 6]. An example performance metric is the mean square error (MSE) between the desired and actual phase response of the system ($\phi_D(f,V)$ and $\phi_P(f,V)$ respectively):

$$\min_V \sum_k \big(\phi_D(f_k,V) - \phi_P(f_k,V)\big)^2. \tag{4.2}$$

Here, V represents a vector of actuator voltages while the f_k represent the different frequencies that are relevant to the system. The actual phase response $\phi_P(f,V)$ can be calculated from physical models of the heaters and optical ring resonators.

The procedure described above has several disadvantages. First of all, we have shown in earlier work that even very small errors in the physical model can have a large detrimental effect on the accuracy of the procedure [8]. Such model errors can never completely be avoided. Second, although physical models are available for each system component, the heater actuators influence each other by means of electrical and thermal crosstalk. This crosstalk can also be modeled, but this requires a number of measurements that is at least equal to the square of the number of heater actuators [5, App. B]. This becomes a problem when the number of heater actuators is large, and it gives even more room for model errors. Finally, the procedure has to be performed before actually running the system, and any changes to the system are not automatically taken into account.

To avoid the drawbacks mentioned above, recently automatic tuning algorithms based on machine learning techniques have been derived [8, 15, 16]. These algorithms use system measurements in the performance metric. Instead of calculating the phase response

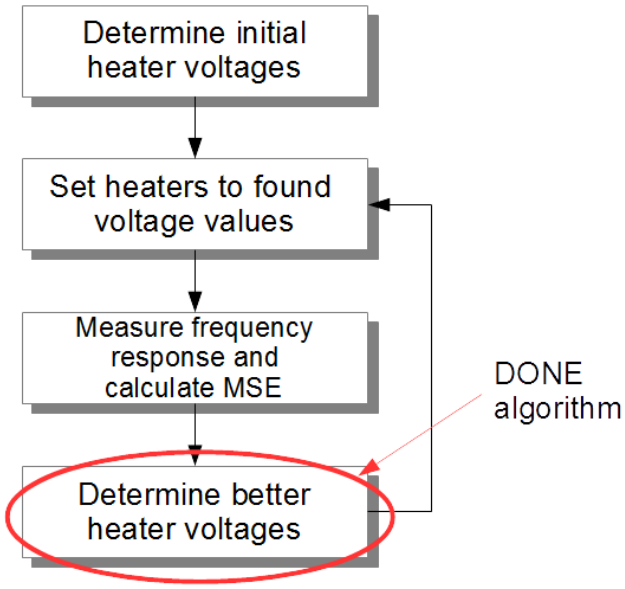


Figure 4.4: Schematic for the automatic tuning method.

from physical models, the phase response is measured directly:

$$\min_{V} \sum_{k} \left(\phi_D(f_k, V) - \phi_M(f_k, V)\right)^2. \tag{4.3}$$

Here, $\phi_M(f, V)$ denotes the measured phase response. This data-based algorithm learns the relation between the actuator voltages and the performance metric in (4.3) and chooses those voltages that optimize this metric. After several iterations, the algorithm converges to a setting of actuator voltages V with the best performance according to the data-based performance metric (4.3). Here, each iteration consists of updating the performance metric with the measured phase response, finding the optimal values for V, and setting the actuator voltages to these values and performing a new measurement. In this work this is done with the data-based online nonlinear extremum-seeker (DONE) algorithm [15]. See Figure 4.4. This algorithm is especially designed for systems where the measurements take some time to be performed, where measurement noise is present, and where the relation between the MSE and the actuators is too complicated to model accurately. The beamformer system described in this paper is an example of such a system. Using such an automatic tuning procedure on this system with over a thousand actuators is unavoidable if it is ever to be used in a real life application. Besides this, the DONE algorithm only uses the MSE in (4.3), which is a scalar value. This makes it easy to be used in practice, where it is more realistic to use a scalar value as the objective, for example by measuring the signal-to-noise ratio or output power of the system

rather than the phase response. The traditional approach with the nonlinear optimization method also uses a scalar-valued objective, but it becomes very slow and struggles with local minima when the output power is used as the objective [17, Sec. 4.4.2].

Unlike the traditional approach (4.2), the data-based approach (4.3) does not require a physical model (including a model for the heater crosstalk) and is therefore not sensitive to model errors. It is also an online algorithm, which means that it can be used while the system is running, and it will continually search for better configurations. This can also easily be adapted to allow the procedure to work for systems that change over time, which is crucial when applied to antenna systems on moving vehicles such as aircrafts. On moving vehicles, the beam angle changes over time, and therefore the objective function also changes over time.

4.3.2. OPTICAL SIDEBAND FILTER TUNING

Earlier in this paper, several subsystems of the photonic beamformer were described. See Figure 4.3. The OSBF and SCT subsystems drastically reduce the required bandwidth of the system [3, 13]. The goal of the OSBF is to filter out one of the sidebands resulting from the modulation scheme used for the E/O conversion, as shown in the frequency responses of Figure 4.3. The result is an optical single sideband full carrier modulation that can be used by the OBFN. With one sideband filtered out, beamforming would need to be performed over the sideband, the optical carrier frequency, and the frequencies in between, giving a total of 14.5 GHz instead of the whole region of 29 GHz. To further reduce the bandwidth, the SCT subsystem is used to ensure that the correct phase is achieved at the carrier frequency. If both the OSBF and SCT subsystems are tuned correctly, a linear phase response only needs to be provided in one sideband with a bandwidth of 0.5 GHz.

Figure 4.5 shows the power response of the full beamformer system after tuning the OSBF subsystem by hand. This was done by measuring the power response with a vector network analyzer (VNA) and tuning the OSBF heaters until the desired response was achieved. The shown frequencies are relative to the carrier frequency. The response features a stopband, that filters out one of the sidebands as shown in Figure 4.3, and a passband that keeps the other sideband. The difference between the stopband and passband is 30dB. In this work it is assumed that the RF signals are downconverted as in earlier work [4, 7], such that the sidebands are located closer to the carrier frequency than the 14.25 GHz mentioned earlier, though the results in this paper still hold for a bandwidth of 0.5 GHz.

4.3.3. AUTOMATIC OPTICAL BEAMFORMING NETWORK TUNING

The optical beamforming network (OBFN) subsystem is used to provide different delays to the signals of each antenna element in such a way that a strong signal is transmitted in one specific direction by positive interference. The correct delays can be calculated from the desired beam angle and the distance between two antenna elements, as shown in the introduction. This leads to a maximum delay of 168 ps for the 1×4 photonic beamformer, which is the one investigated in this work. Larger delay values were considered in this work to show that the maximum delays can indeed be achieved, to allow different antenna configurations or scan angles, and for better visibility of the measurements.

The desired group delay response for each OBFN path is a flat response with a value

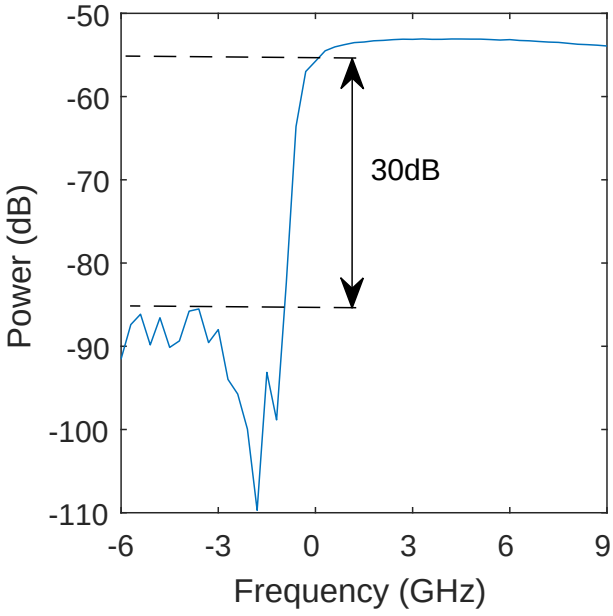


Figure 4.5: Measured power response of the full beamformer system after tuning the optical sideband filter by hand. Frequencies are shown relative to the carrier frequency.

between 0 ps and the maximum group delay value, depending on the desired beam angle. The group delay response can be made flat by using a serial configuration of ORRs for each OBFN path [12]. This flat response has to be provided over the whole frequency range of interest. However, by implementing the SCT scheme explained later in this section, the only part of the frequency domain for which a flat group delay response needs to be provided is the sideband that is not filtered by the OSBF, as shown in Fig. 4.3. This greatly reduces the required number of ORRs and therefore the complexity of the system, because there is a trade-off between the required number of ORRs, the maximum group delay, and the bandwidth [3].

The procedure for tuning the OBFN is similar to the one for tuning the OSBF: the group delay response is measured with a VNA, and the heaters of each ring resonator are tuned until the desired group delay response is achieved. Here, we made use of the automatic tuning method described in this paper. The top of Figure 4.6 shows the resulting phase response of both the manual and the automatic tuning procedures of one path of the photonic beamformer system for two different delay settings (250 ps and 408 ps). The 1-to-4 splitter shown in Figure 4.3 has been set to provide this path with 100% of the signal power. In the automatic procedure, the frequency range where a linear phase response should be achieved was set to 4.5-8.1 GHz, which corresponds to a much larger bandwidth than the desired bandwidth of 0.5 GHz. The manual tuning approach used a frequency range of the same bandwidth but no specific frequency range was given. All results were time averaged to reduce noise.

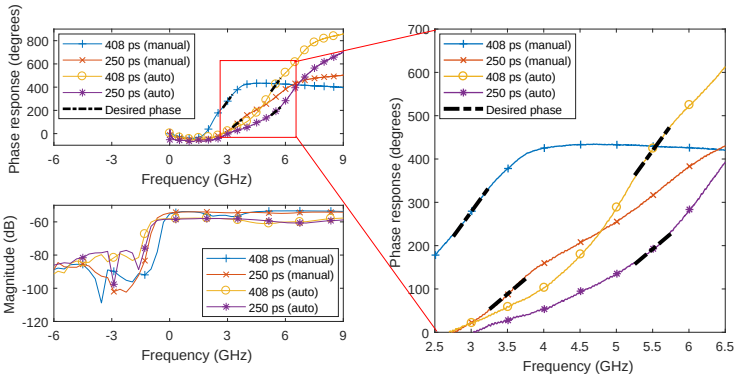


Figure 4.6: Measured phase response (top) and power response (bottom) of one path of the full beamformer system from Figure 4.3 after tuning the optical beamforming network automatically (using the DONE algorithm) and by hand for two different delay settings. The right side zooms in on the highlighted region. Desired phase responses are shown with dashed lines. The maximum difference between the measured and desired phase response for the four plots is shown in Table 4.1. Frequencies are shown relative to the carrier frequency, and the phase responses are shown relative to the phase response of the system with no delay.

The difference between the desired and the measured phase response must remain within 11.25°. The maximum errors are shown in Table 4.1. As can be seen, all phase errors remain well within the requirements. This shows that the automatic tuning procedure can be used instead of the manual tuning procedure for the photonic beamformer system.

The bottom of Figure 4.6 shows the corresponding power response. For the manual tuning method this is similar to the one shown in Figure 4.5, with some influence of the ring resonators. For the automatic tuning method, the magnitude response was tuned automatically too, using the same procedure for the OSBF as for tuning the OBFN, with the magnitude instead of the phase response. However, some improvement is possible here, as the loss in the passband could still be reduced. It seems that the MSE criterion does not work too well with the logarithmic scale. The MSE criterion is especially sensitive to data points in the stop band in this case. This was already somewhat circumvented by scaling the data points in the stopband by a factor 0.3, but other objective functions should be considered in the future.

Table 4.1: Maximum phase error in degrees for tuning the OBFN subsystem automatically (A) and by hand (M) for different delay values. All errors are well below the maximum allowed error of 11.25°.

Delay	408 ps	250 ps	408 ps	250 ps
Method	(M)	(M)	(A)	(A)
Max error	**6.2°**	**4.9°**	**5.5°**	**6.4°**

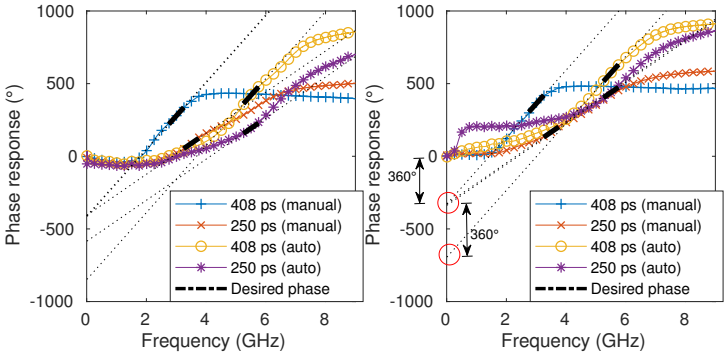


Figure 4.7: Measured phase response of the full beamformer system before (left) and after (right) manually tuning the SCT subsystem. The word 'auto' in the legend refers to the automatic tuning of the OSBF and OBFN subsystems. Using the SCT principle, the height of the phase responses is adapted in such a way that the corresponding linear responses meet at the carrier frequency, up to multiples of 360 degrees, as indicated by the red circles. These results have also been published in [18].

4.3.4. SEPARATE CARRIER TUNING

The results of the previous subsection show that both the manual and automatic tuning procedures result in a phase response with the correct slope, satisfying the requirements. However, besides the slope of the phase response, the height of the phase response also needs to be set correctly [13, 18]. Where the slope of the phase response is controlled by the OBFN subsystem, the height of the phase response is controlled by the separate carrier tuning (SCT) subsystem. This is also shown in Figure 4.3.

Like the other subsystems, the SCT subsystem makes use of optical ring resonators. Two ring resonators are tuned in such a way that they affect the phase response in the region between the carrier frequency and the sideband. This region contains no information, so it is not necessary to follow the desired linear phase response here. Tuning these two ring resonators gives an extra phase shift while leaving the shape and the slope of the phase response unaffected. This ensures that the phase response follows the desired linear phase response at the carrier frequency too. It should be noted that phase differences of multiples of 360° do not affect the system [13]. This makes it theoretically possible to use only one ring resonator for this adjustment, though in practice it is easier to use two of them. The ring resonators in the SCT subsystem were adjusted by hand after tuning their heater actuators in such a way that the ring resonators only operate in the region near the carrier frequency.

Figure 4.7 shows the effect of not using the SCT subsystem on the left. Although the phase response has the desired slope, there is a phase mismatch at the carrier frequency (at 0 GHz). Tuning the ring resonators of the SCT subsystem solves this problem. The effect of tuning the SCT subsystem is shown in the same figure on the right. This time, the phase responses have been shifted up or down in such a way that the corresponding linear phase responses are a multiple of 360 degrees off at the carrier frequency. The

accuracy can still be improved, however: due to their proximity on the photonic chip, the heaters of the SCT subsystem slightly influence the heaters of the OBFN subsystem, which leads to small changes to the slope of the phase response. This causes the desired phase response at the carrier to be about 30 degrees off, so exact multiples of 360 degrees are not yet achieved. These errors are unacceptable for practical applications. However, using the same automatic tuning procedure for this subsystem together with the other two subsystems could circumvent this problem. This remains for future work.

4.4. CONCLUSION

We have proposed a novel photonic beamformer for a transmit phased array antenna. The beamformer is based on optical ring resonators and is fully integrated on a chip. We have investigated manual and automatic procedures for tuning the photonic beamformer. Automatic procedures are essential in real-life applications where thousands of actuators are considered and where a limited number of variables can be measured, such as the signal-to-noise ratio. Both the manual and automatic procedures provided beamforming functionalities with a phase error of less than 11.25° over the whole frequency band that contains signal information. This is the first time that part of this transmit antenna tile has been tuned automatically.

The system is designed with aircraft-satellite communication in mind as the main application, providing satellite Internet connections on board the aircraft using the K_u-band. The separate carrier tuning principle greatly reduces the operating bandwidth for the photonic beamformer, from 29 GHz to 0.5 GHz. The ring resonator-based subsystems of the beamformer make squint-free beamforming possible. A fully automated transmit antenna tile remains for future work.

4.5. APPENDIX: MEASUREMENT SETUP

Figure 4.8 shows the measurement set-up for the frequency response measurements shown in this paper. A Rohde& Schwarz ZVA40 VNA generated a 50 MHz RF signal to modulate a laser with a Mach-Zehnder modulator. The modulated optical signal was coupled into the beamformer control box. The control box was connected via USB to a laptop with heater control software. This laptop also contained the automatic tuning software and software for reading the VNA measurements. The integrated photodetectors inside the control box were connected with RF cables to the second VNA port. Although integrated modulators were also available, these have not been used in order to prevent crosstalk with the integrated detectors.

The VNA measurements were sent to the laptop with the automatic tuning and heater control software via an ethernet cable. The phase-shift method [4, Sec. IV-A] was used for the group delay and power response measurements, using a function generator for the laser. With this method, the phase response shown on the VNA is actually equivalent to the group delay response of the system, so the objective (4.3) minimized by the DONE algorithm is actually equal to the mean square error between the desired and measured group delay response. In Figures 4.6 and 4.7, after the minimization procedure, the phase response was measured directly using a frequency sweep on the VNA (without the external trigger from the function generator). Here, the laser current was

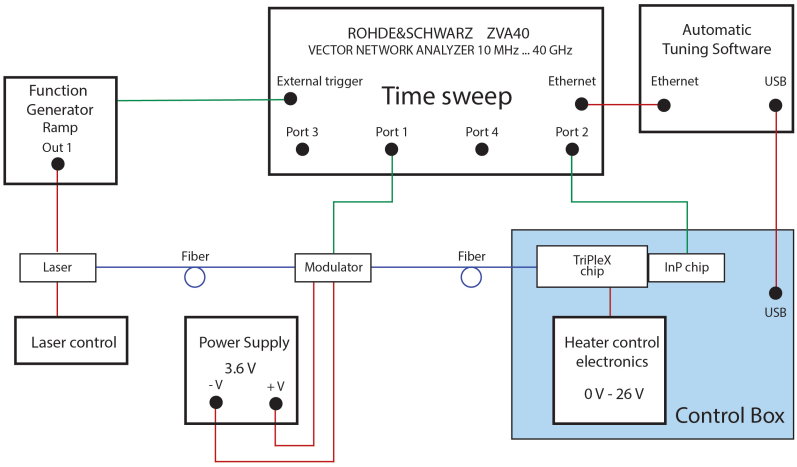


Figure 4.8: Measurement set-up for the frequency response measurements used in this paper.

Table 4.2: Values of the hyper-parameters used by the automatic tuning algorithm.

Hyper-parameter	Value (OSBF tuning)	Value (OBFN tuning)
Total num. of measurements	2000	1000
Num. of basis functions	3000	3000
Regularization parameter	0.1	0.1
Standard deviation of frequencies	1.0	1.0
Exploration parameter	0.01	0.05
Sliding window size	60	60

chosen in such a way that the optical carrier would be at 0 GHz in Figure 4.5 after tuning the OSBF.

4.6. APPENDIX: ALGORITHM SETTINGS

The automatic tuning algorithm used in this paper contains several hyper-parameters that are explained and investigated in [15]. Their values as used in this paper are shown in Table 4.2 for the OSBF and OBFN tuning results. The last value indicates the size of the sliding window: only the last 60 measurements are used in updating the model used by the algorithm, using the same adaptation of the algorithm as in [19].

REFERENCES

[1] L. Bliek, S. Wahls, I. Visscher, C. Taddei, R. B. Timens, R. Oldenbeuving, C. Roeloffzen, and M. Verhaegen, *Automatic Tuning of a Novel Ring Resonator-based Photonic Beamformer for a Transmit Phased Array Antenna,* arXiv e-prints , arXiv:1808.04814 (2018), arXiv:1808.04814 [physics.app-ph] .

[2] R. C. Hansen, *Phased array antennas*, Vol. 213 (John Wiley & Sons, 2009).

[3] A. Meijerink, C. G. Roeloffzen, R. Meijerink, L. Zhuang, D. A. Marpaung, M. J. Bentum, M. Burla, J. Verpoorte, P. Jorna, A. Hulzinga, *et al.*, *Novel ring resonator-based integrated photonic beamformer for broadband phased array receive antennas—part I: Design and performance analysis,* J. Lightwave Technol. **28**, 3 (2010).

[4] L. Zhuang, C. G. Roeloffzen, A. Meijerink, M. Burla, D. A. Marpaung, A. Leinse, M. Hoekman, R. G. Heideman, and W. van Etten, *Novel ring resonator-based integrated photonic beamformer for broadband phased array receive antennas—part II: Experimental prototype,* Journal of lightwave technology **28**, 19 (2010).

[5] L. Zhuang, *Ring resonator-based broadband photonic beam former for phased array antennas*, Ph.D. thesis, University of Twente (2010).

[6] J. Verpoorte, H. Schippers, P. Jorna, C. G. Roeloffzen, D. A. Marpaung, R. Baggen, and B. Sanadgol, *Architectures for Ku-band broadband airborne satellite communication antennas,* in *32nd EAS Antenna Workshop, 2010* (National Aerospace Laboratory NLR, 2010).

[7] M. van der Vossen, G. Voshaar, C. Roeloffzen, A. Hulzinga, and M. Iven, *Design of a highly integrated Ku-band planar broadband phased array receiver with dual polarization,* in *Microwave Conference (EuMC), 2014 44th European* (IEEE, 2014) pp. 1695–1698.

[8] L. Bliek, M. Verhaegen, and S. Wahls, *Data-driven minimization with random feature expansions for optical beam forming network tuning,* 16th IFAC Workshop on Control Applications of Optimization (CAO'2015) **48**, 166 (2015).

[9] D. Marpaung, C. Roeloffzen, R. Heideman, A. Leinse, S. Sales, and J. Capmany, *Integrated microwave photonics,* Laser & Photonics Reviews **7**, 506 (2013).

[10] L. Zhuang, C. Roeloffzen, R. Heideman, A. Borreman, A. Meijerink, and W. Van Etten, *Single-chip optical beam forming network in LPCVD waveguide technology based on optical ring resonators,* in *International Topical Meeting on Microwave Photonics, 2006. MWP'06.* (IEEE, 2006) pp. 1–4.

[11] C. G. Roeloffzen, M. Hoekman, E. J. Klein, L. S. Wevers, R. B. Timens, D. Marchenko, D. Geskus, R. Dekker, A. Alippi, R. Grootjans, *et al.*, *Low-loss si3n4 triplex optical waveguides: Technology and applications overview,* IEEE journal of selected topics in quantum electronics **24**, 1 (2018).

[12] P. Megret, C. Roeloffzen, M. Wuilpart, L. Zhuang, R. Heideman, S. Bette, A. Borreman, N. Staquet, and W. van Etten, *Ring resonator-based tunable optical delay line in LPCVD waveguide technology,* in *10th Annual Symposium of the IEEE/LEOS Benelux Chapter 2005, Mons, Belgium* (2005) pp. 79–82.

[13] M. Burla, M. R. Khan, D. A. Marpaung, L. Zhuang, C. G. Roeloffzen, A. Leinse, M. Hoekman, and R. Heideman, *Separate carrier tuning scheme for integrated optical delay lines in photonic beamformers,* in *International Topical Meeting on Microwave Photonics, 2011 & Microwave Photonics Conference, 2011 Asia-Pacific, MWP/APMP* (IEEE, 2011) pp. 65–68.

[14] L. F. Stokes, M. Chodorow, and H. J. Shaw, *All-single-mode fiber resonator,* Optics Letters **7**, 288 (1982).

[15] L. Bliek, H. R. G. W. Verstraete, M. Verhaegen, and S. Wahls, *Online optimization with costly and noisy measurements using random Fourier expansions,* IEEE Transactions on Neural Networks and Learning Systems **29**, 167 (2018).

[16] L. Bliek, H. Verstraete, S. Wahls, R. B. Timens, R. Oldenbeuving, C. Roeloffzen, and M. Verhaegen, *Automatic tuning of a ring resonator-based optical delay line for optical beamforming,* in *Symposium on Information Theory and Signal Processing in the Benelux, 2017* (IEEE, 2017) pp. 23–24.

[17] R. Blokpoel, *Staggered delay tuning algorithms for ring resonators in optical beam forming networks*, Master's thesis, University of Twente (2007).

[18] C. Roeloffzen, I. Visscher, C. Taddei, D. Geskus, R. Oldenbeuving, J. Epping, R. B. Timens, P. van Dijk, R. Heideman, M. Hoekman, R. Grootjans, L. Bliek, S. Wahls, and M. Verhaegen, *Integrated microwave photonics for 5G,* in *Conference on Lasers and Electro-Optics* (Optical Society of America, 2018) p. JTh3D.2.

[19] P. Pozzi, D. Wilding, O. Soloviev, H. Verstraete, L. Bliek, G. Vdovin, and M. Verhaegen, *High speed wavefront sensorless aberration correction in digital micromirror based confocal microscopy,* Optics Express **25**, 949 (2017).

5

ONLINE FUNCTION MINIMIZATION WITH CONVEX RANDOM RELU EXPANSIONS

We propose CDONE, a convex version of the DONE algorithm. DONE is a derivative-free online optimization algorithm that uses surrogate modeling with noisy measurements to find a minimum of objective functions that are expensive to evaluate. Inspired by their success in deep learning, CDONE makes use of rectified linear units, together with a non-negativity constraint to enforce convexity of the surrogate model. This leads to a sparse and cheap to evaluate surrogate model of the unknown optimization objective that is still accurate and that can be minimized with convex optimization algorithms. The CDONE algorithm is demonstrated on a toy example, on the problem of hyper-parameter optimization for a deep learning example on handwritten digit classification, and on the problem of photonic beamformer tuning.

Parts of this chapter have been published in [1].

5.1. INTRODUCTION

MANY practical optimization problems do not satisfy the assumptions that are present in traditional continuous optimization algorithms. Examples of these assumptions are that the derivative of the function to be optimized is known, or that there is at least a mathematical expression for the function, or that the function can be evaluated quickly and accurately. But the outcome of a simulation or algorithm, for example, can depend on many parameters and can suffer from noise. In many cases it is undesirable to try a new set of parameters and check for improvement multiple times, which is what happens in grid search or random search techniques.

Several paths have been taken in alleviating this problem. Most derivative-free optimization algorithms [2] are able to operate without the assumptions mentioned above. The ones that seem most fit to deal with noisy and expensive measurements are in the class of Bayesian optimization algorithms [3–5]. These algorithms use the available data and a prior to fit a probabilistic surrogate model and then use this model to decide where the next measurement should be taken. Hyper-parameter optimization is just one of the many examples where Bayesian optimization algorithms have shown their potential.

Another algorithm that is based on surrogate models is the DONE algorithm [6]. The surrogate model used in this algorithm is a random feature expansion (RFE) [7], which is updated every time a new measurement comes in. At each iteration of the algorithm a measurement of the objective is taken, then the surrogate model is updated, and then a new measurement location is proposed based on the minimum of the surrogate model. Using RFEs as a surrogate model makes it possible to get a fixed computational complexity per iteration by using recursive least squares updates. In comparison, Bayesian optimization algorithms become slower over time. The DONE algorithm was shown to outperform a popular Bayesian optimization algorithm on several tasks, such as the tuning of an optical beam-forming network [6].

This paper proposes an adaptation of the DONE algorithm called CDONE that has several advantages:

- There are less hyper-parameters to tune.
- The surrogate model is convex.
- The surrogate model is evaluated faster.
- The surrogate model is inherently sparse.

The DONE algorithm already had few hyper-parameters to tune, but having even less is a big advantage. The convexity allows convex optimization algorithms to be used to find the global minimum of the model, as opposed to finding a local minimum. The last two advantages make it possible to find the global minimum efficiently.

The convex model used in this paper is a combination of RFEs and rectified linear units (ReLUs) [8], which will be explained in the next section. Section 5.3 describes the CDONE algorithm. A comparison with the DONE algorithm is given in Section 5.4. Section 5.5 describes the results of both algorithms on an artificial example, on a hyper-parameter tuning problem for deep learning, and on the problem of photonic beam-former tuning. Finally, Section 5.6 contains conclusions and recommendations for future work.

5.2. RANDOM RELU EXPANSIONS

RFEs [7] have gained popularity recently due to their ability to approximate kernels with low dimensionality. They are defined as a weighted sum of basis functions with random parameters, and can be trained with conventional regularized linear least squares techniques. Since the number of basis functions and the values of the random parameters stay fixed, these models are particularly fit for problems with many data samples. Theoretical approximation guarantees for RFEs are available for the L_2 norm [9] and for the L_∞ norm [7]. Practically relevant results for the L_2 norm with models trained with regularized least squares are also available [6]. In short, any continuous function on a compact domain can be approximated arbitrarily well if the number of basis functions are large enough. The approximation error scales with the inverse square root of the number of basis functions. In practice, however, the approximation accuracy is sensitive to hyper-parameters such as the probability distribution of the random parameters. Recommendations for these hyper-parameters in the case of random cosine features are given in [6].

At the same time, ReLUs [8] have become a popular activation function in deep neural networks because of the inherent sparsity and the ability to circumvent the vanishing gradient problem. Even shallow ReLU networks can act as universal approximators [8]. In this work we use random features based on ReLUs, to make use of the advantages of both principles.

Define the ReLU $\phi : \mathbb{R} \to \mathbb{R}$ as

$$\phi(z) = \begin{cases} z, & z > 0, \\ 0, & z \leq 0. \end{cases} \tag{5.1}$$

Then, a Random ReLU expansion (RRE) is a model of the form

$$\mathrm{RRE}(\mathbf{x}) = c_D - c_{D-1} + \sum_{k=1}^{D-2} c_k \phi(\mathbf{w}_k^T \mathbf{x} + b_k), \tag{5.2}$$

with $\mathbf{w}_k \in \mathbb{R}^d$ and $b_k \in \mathbb{R}$ being realizations of i.i.d. random variables from continuous probability distributions. We assume $c_k \geq 0 \ \forall k$, so the first two terms are required for a bias that also allows the model to approximate negative values. With this assumption, the model is a convex function of $\mathbf{x}$.

In the area of neural networks, the parameters $c_k, \mathbf{w}_k$ and b_k are trained with stochastic gradient descent or similar algorithms. In the RRE model, however, $\mathbf{w}_k$ and b_k are chosen randomly, and finding the optimal parameters c_k is a convex optimization problem.

5.3. THE CDONE ALGORITHM

We present an adaptation to the DONE algorithm [6], with ReLU basis functions and a convexity constraint $c_k \geq 0 \ \forall k$, to find the minimum of an unknown function f using noisy measurements y_n. To initialize the CDONE algorithm, an initial guess $\mathbf{x}_1$ is needed, together with its corresponding measurement y_1. The random parameters $\mathbf{w}_k$ and b_k are drawn independently from their probability distributions and remain fixed for the whole

duration of the algorithm. In this paper we have used the uniform distribution on $[-1,1]$ for both $\mathbf{w}_k$ and b_k. The algorithm then repeats the following three steps:

5.3.1. FITTING THE SURROGATE MODEL

To fit the RRE to the data $(\mathbf{x}_n, y_n)$ for iterations $n = 1,\dots,n$, while imposing a convexity constraint, the following regularized nonnegative linear least squares problem needs to be solved:

$$\min_{\mathbf{c}} \sum_{n=1}^{n} \big(y_n - \mathrm{RRE}(\mathbf{x}_n;\mathbf{c})\big)^2 + \lambda||\mathbf{c}||_2^2, \tag{5.3}$$

$$\mathbf{s.t.}\ c_k \geq 0,\ k = 1,\dots,D. \tag{5.4}$$

Here, λ is a regularization parameter, which can be chosen quite small in practice (e.g. $\lambda = 10^{-8}$) because the convexity constraint already helps in preventing overfitting. Being less sensitive to this parameter is a big advantage over the DONE algorithm. The above optimization problem is a nonnegative least squares problem. This problem is convex and can be solved with, for example, an active set method [10].

5.3.2. FINDING THE MINIMUM OF THE SURROGATE MODEL

After fitting the RRE model with optimal coefficients $\mathbf{c}^*$, we find the minimum of this model:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in X} \mathrm{RRE}(\mathbf{x};\mathbf{c}^*). \tag{5.5}$$

Here, X is a convex compact set, e.g. $X = [-1,1]^d$. In the original DONE algorithm, only a local minimum of the surrogate model is found. The initial guess provided to the solver is the current measurement $\mathbf{x}_n$, plus a small perturbation to aid in exploration. However, because the RRE in the CDONE algorithm is convex, we can find the global minimum in this case with a convex optimization algorithm. There is also no need to add an extra exploration step by perturbing the initial guess.

The original DONE algorithm uses second-order optimization methods like the L-BFGS method [10]. Because of the structure of the RRE however, we propose a steepest descent method with a backtracking line search and gradient projection [10] for the CDONE algorithm. It can be seen that the RRE is a piecewise linear function, so first-order approximations are exact in a certain set, and the line search should help in taking the largest possible step within this set.

5.3.3. CHOOSE A NEW MEASUREMENT POINT

The found global optimum of the RRE is used to determine a new measurement point. Although the RRE model is convex, the original objective f might be non-convex, so a small random perturbation ζ is added to $\mathbf{x}^*$ for exploration purposes. Only a local approximation of f around its (local) minimum is needed, and the RRE becomes more accurate around this point as new measurements are added. A new measurement is taken at $\mathbf{x}_{n+1} = \mathbf{x}^* + \zeta$ (after projecting onto X), which leads to a new value y_{n+1}, and the algorithm repeats at step 1.

5.4. COMPARISON WITH THE DONE ALGORITHM

Several factors influence the computational costs of the CDONE algorithm, compared to the DONE algorithm. First of all, the basis functions used in the DONE algorithm are cosines, whereas the CDONE algorithm uses ReLU basis functions. Although this has no influence on the order of complexity, some computation time can be saved by implementing the ReLU basis function with a simple IF-statement as in (5.1). This is faster than calculating a cosine. Another time saver is the sparsity of the RRE model, which occurs in two ways: sparsity of the basis functions, and sparsity of the nonnegative weights c_k. The first case occurs if $\mathbf{w}_k^T\mathbf{x}+b_k \leq 0$, in which case a scalar multiplication and addition do not have to be computed. This inherent sparsity is one of the reasons ReLUs are used in deep learning. The second case occurs if $c_k = 0$ after fitting the model. In this case, $\phi(\mathbf{w}_k^T\mathbf{x}+b_k)$ does not have to be computed, saving a vector-vector multiplication of the same size as the input $\mathbf{x}$, a vector addition, and an IF-statement. The next section illustrates how often this happens in practice.

The convexity of the model used in the CDONE algorithm allows convex optimization algorithms to find the global minimum of the model. In the DONE algorithm, finding the global minimum is an intractable problem due to the non-convexity of the model. Furthermore, unlike the DONE algorithm, the CDONE algorithm does not include an exploration step by perturbing the initial guess of this convex optimization algorithm. For each iteration, this saves on computation time equal to the time required to draw a random vector of the same dimension as the input $\mathbf{x}$.

The only part of the CDONE algorithm that could increase its computation time when compared to the DONE algorithm, is the fitting of the surrogate model. Step 1 of the CDONE algorithm, fitting the surrogate model, is the most computationally expensive step of the algorithm. In the original DONE algorithm, a recursive least squares update was used to reduce the computation time of this step. In the CDONE algorithm, this should be changed to a recursive nonnegative least squares update. Several algorithms exist for this purpose [11–14], all with varying numerical stability, accuracy, and computational complexity. The approach in [14], based on time-, order-, and active-set-recursion, seems the most fit for this problem. With this implementation, the nonnegative least squares problem (5.3)-(5.4) can be solved recursively in $O(D^2)$, just like in the DONE algorithm, provided that the active set recursion can be carried out in $O(1)$ steps. In this paper, we did not use a recursive algorithm to solve (5.3)-(5.4), but applied the active set method directly for ease of use. We do plan to investigate a recursive implementation in the future.

In Section 5.5 we note that the active constraints between two subsequent iterations of the CDONE algorithm differ only by 2 on average in a simple test case. This implies that the average order of complexity of the fitting step of the CDONE algorithm could indeed be reduced to $O(D^2)$ in practice.

5.5. NUMERICAL EXAMPLES

In this section we test the CDONE algorithm on two numerical examples: finding the minimum of a convex function perturbed by noise, and finding the optimal hyper-parameters of a deep learning classification problem.

	DONE	DONE RELU	CDONE	CDONE ELU
Mean	0.0132	0.0150	0.0155	0.0244
Std	0.0074	0.0073	0.0091	0.0148
Time	55.735	64.266	27.991	34.427

Table 5.1: Final distance to the true minimum of the convex function, averaged over 100 runs, and average computation time in seconds.

5.5.1. MINIMIZING A NOISY CONVEX FUNCTION

As a test case, consider the function

$$f(\mathbf{x}) = \sqrt{\mathbf{x}^T\mathbf{x}} - 5, \tag{5.6}$$

with $\mathbf{x} \in \mathbb{R}^2$. We have access to this function via noisy measurements $y(\mathbf{x}) = f(\mathbf{x}) + 0.01\eta$, where η has a standard normal distribution.

The minimum of f is found with four variations of the DONE algorithm: the standard DONE algorithm, the DONE algorithm with ReLU basis functions instead of cosines (DONE RELU), the CDONE algorithm as presented in this paper, and the CDONE algorithm with exponential linear units [15] (ELUs) as basis functions (CDONE ELU). The comparison with the smoother ELUs is made to determine the effect of the smoothness of the basis functions. All algorithms used $D = 500$ basis functions and $N = 500$ measurements, with a regularization parameter of $\lambda = 10^{-2}$ for DONE and DONE RELU, and $\lambda = 10^{-8}$ for CDONE and CDONE ELU. The convexity constraints of the latter two algorithms reduce the risk of overfitting to noise, so they need less regularization as a consequence. The variance of the exploration parameter was set to 10^{-4} for all algorithms. The DONE and DONE RELU algorithms used the standard normal distribution for their respective $\mathbf{w}_k$ parameters, which is the default approach that works well in practice [6]. The experiment was repeated 100 times starting from random initial guesses in $[-1,1]^2$.

Table 5.1 shows the distance of the found minimum $\mathbf{x}^*$ to the true minimum, with the mean and standard deviation from 100 runs, as well as the average computation time in seconds. Please note that the computation time can be improved for all four algorithms, as we used (adaptations of) a slower version of the DONE algorithm available online [16]. Furthermore, the CDONE and CDONE ELU implementations do not yet use a recursive algorithm for fitting the surrogate model, as mentioned in Section 5.4. Figure 5.1 shows how the average distance progresses over time. It can be seen that the CDONE algorithm achieves a similar accuracy as the other variations, with CDONE ELU performing slightly worse. However, a larger difference between the variations can be seen in Figure 5.2. This figure shows the mean number of nonzero coefficients c_k, $k = 1,\ldots,D$. The CDONE algorithm uses only about 16 out of all 500 available basis functions. Furthermore, the set of basis functions that are used remains fairly constant as can be seen in Figure 5.3. On average, the difference between the active set of coefficients $\{k : c_k = 0\}$ for a particular iteration and the next is less than 1, although this difference can go up to around 15 in one of the earlier iterations. We conclude that the CDONE algorithm has a high accuracy compared to the number of used basis functions, and that there is potential for efficient implementations of the minimization step of this algorithm by exploiting the sparsity.

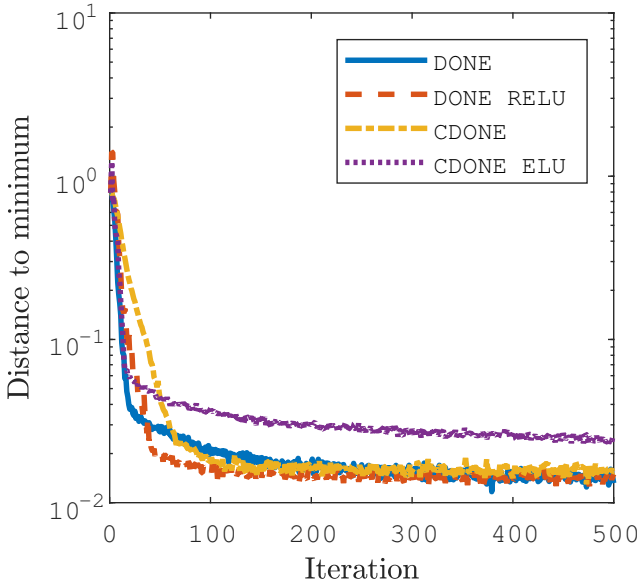


Figure 5.1: Distance to the true minimum of the convex function, averaged over 100 runs.

5.5.2. HYPER-PARAMETER OPTIMIZATION FOR DEEP LEARNING

In our second experiment we consider the problem of hyper-parameter optimization. The task we consider is a handwritten digits recognition example [17], where a deep neural network is trained to classify handwritten digits. We modify this example so that the following eight hyper-parameters are considered unknown, even though values are given in the example: height, width and stride of the filter of the convolutional layer, height, width and stride for the max pooling layer, the maximal number of epochs, and the initial learning rate. The example ends by showing the accuracy on the given test set after training.

The function we wish to minimize takes an 8-dimensional input and converts it from the set $[-1,1]^8$ to realistic values for the hyper-parameters. Hyper-parameters that should have integer values are rounded. Then we run the example with these values for the hyper-parameters, and we take -1 times the accuracy on the test data as the output to be minimized.

Figure 5.4 shows the accuracy for the same four algorithm variations as in the previous example, for 10 runs, starting from 10 different initial guesses. The initial guesses were shared by the different algorithms. Note that some initial guesses were so bad that the accuracy was precisely 0, and the algorithms had trouble getting out of this part of the hyper-parameter space. The last plot shows the best result found by each algorithm, as well as the result provided in the original example [17]. All algorithms gave better results than the results given in the example. We used the same settings as in the previous example, but the number of basis functions and measurements were changed to $D = 800$

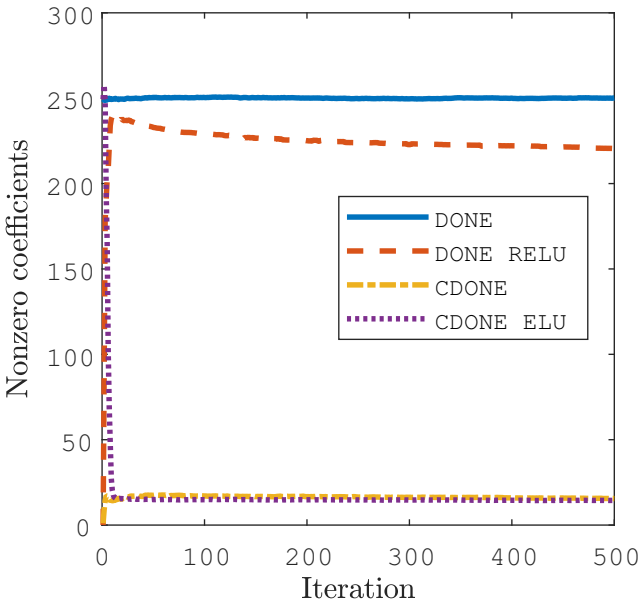


Figure 5.2: Average number of nonzero coefficients per iteration.

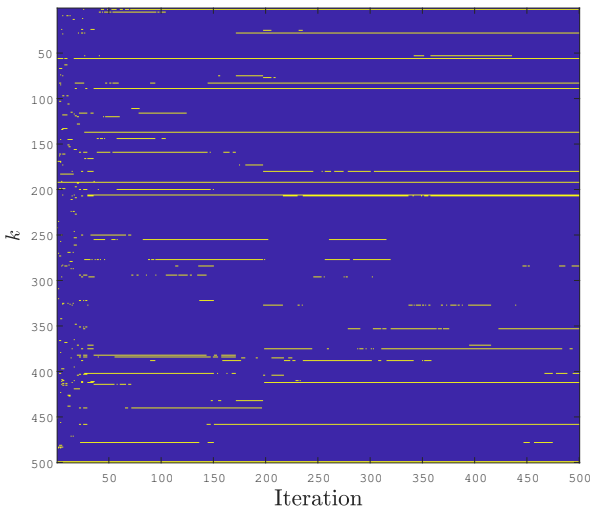
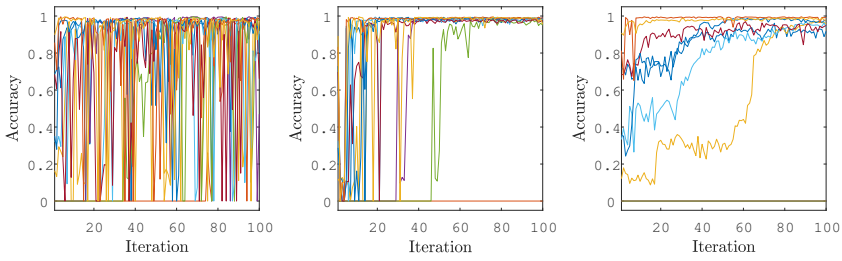
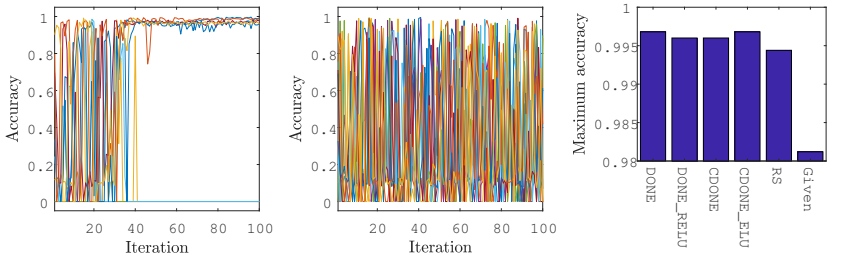


Figure 5.3: Nonzero coefficients pattern for one of the runs of CDONE. Yellow indicates $c_k > 0$ for a particular basis function k at that iteration, while blue indicates $c_k = 0$.



(a) DONE (b) DONE_RELU (c) CDONE



(d) CDONE_ELU (e) Random Search (f) Best results

Figure 5.4: Accuracy of the current iteration of various algorithms, on the test set for the deep learning example. Ten individual runs are shown. The last plot shows the accuracy of the best iteration of all the runs, where the last column shows the accuracy given in the original example.

and $N = 100$, respectively. We also make a comparison with a random search over the hyper-parameter space. The random search can provide good hyper-parameter settings in just a few iterations, and so does the DONE algorithm. However, an advantage of the CDONE algorithm is that it stays near the currently best found solution and keeps improving. This allows the user to perform the original task while the hyper-parameters are being optimized. This is important in online applications, such as aberration correction for fluorescence microscopy [18], where the quality of the solution should not deteriorate during the optimization procedure. The most stable behavior, with a clear convergence plot, is found in the CDONE algorithm, although this algorithm had trouble with the worst initial guesses.

The number of nonzero coefficients in CDONE fluctuated between 20 and 60 in this example. We again conclude that the performance of the CDONE algorithm is very high compared to the number of basis functions that are actually used. We also conclude that the CDONE algorithm can be used for non-convex optimization problems despite the convexity constraint, and that the convexity constraint gives rise to stable behavior of the algorithm.

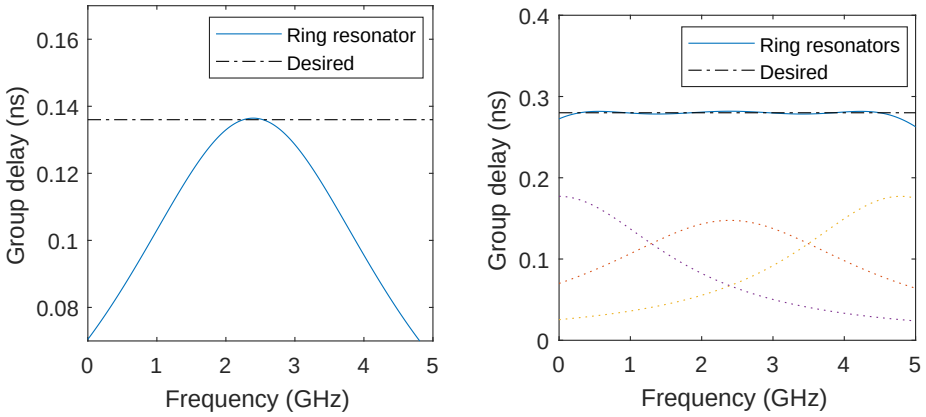


Figure 5.5: (left) Group delay response of one optical ring resonator, and a desired group delay response. (right) Group delay response of a cascade of three optical ring resonators, and a desired group delay response. The individual group delay response of each ring resonator is denoted by the dotted lines.

5

5.5.3. PHOTONIC BEAMFORMER TUNING

In this third experiment we consider the problem of tuning a photonic beamformer. Beamformers are used to electronically steer the beam angle of an array of antenna elements [19]. This is done by providing a time delay to each antenna element. By making use of the latest developments in integrated microwave photonics [20], a low loss, low weight, broadband photonic beamformer has been developed [21, 22]. This type of beamformer makes use of optical ring resonators to provide the desired time delays over a certain bandwidth. See Figure 5.5 for the group delay response of one and multiple optical ring resonators. For a cascade of multiple ring resonators, the delays are simply added up.

We consider a simulation of a cascade of five ring resonators that together need to provide a target delay τ^* in a specific frequency range. The used values correspond to one of the problems encountered in Chapter 4 of this thesis: tuning a cascade of five ring resonators over a bandwidth of 0.5 GHz to a delay of $\tau^* = 1.214$ ns. This problem was not addressed in that chapter.

The group delay response of the ring resonators is given by

$$\begin{aligned}\tau_i(\omega,\kappa_i,\phi_i) = \; & T_i \frac{r_i^2 - r_i\sqrt{1-\kappa_i}\cos(\omega T_i+\phi_i)}{r_i^2+1-\kappa_i-2r_i\sqrt{1-\kappa_i}\cos(\omega T_i+\phi_i)} \\ & + T_i \frac{r_i\sqrt{1-\kappa_i}\cos(\omega T_i+\phi_i) - r_i^2(1-\kappa_i)}{r_i^2(1-\kappa_i)+1-2r_i\sqrt{1-\kappa_i}\cos(\omega T_i+\phi_i)},\end{aligned} \tag{5.7}$$

with κ_i, ϕ_i the control variables of ring resonator i, $r_i = 0.99$, and $T_i = 3.9866 \cdot 10^{-11}$ for $i = 1,\ldots,5$. We used the DONE and the CDONE algorithms for the following minimiza-

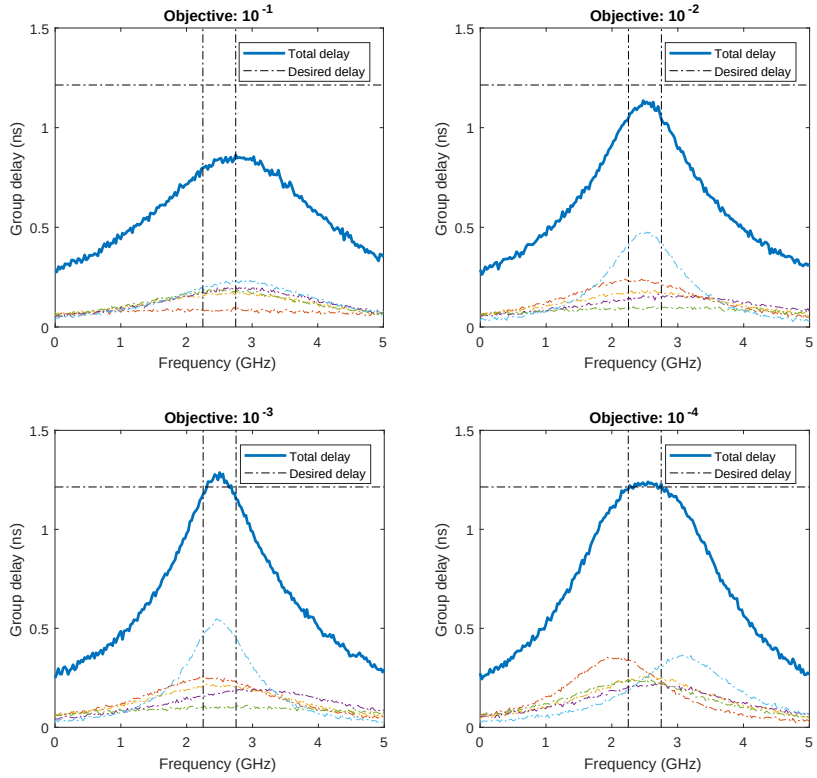


Figure 5.6: Group delay response of a cascade of five optical ring resonators at different iterations of the CDONE algorithm. The objective from the minimization problem (5.8) is shown above each figure, and the group delay responses of each individual optical ring resonator are also shown. The goal is to achieve a flat group delay response of the desired delay value in the bandwidth indicated by the vertical lines.

tion problem:

$$\begin{aligned} \min_{\kappa_1,\ldots,\kappa_5,\phi_1,\ldots,\phi_5} \quad & \frac{1}{N}\sum_{n=1}^{N}\left(\frac{\tau^* - \sum_{i=1}^{5}\left[\tau_i(\omega_n,\kappa_i,\phi_i)+\varepsilon\right]}{\tau^*}\right)^2, \\ \textbf{s.t.} \quad & 0.1 \le \kappa_i \le 0.999,\ i = 1,\ldots 5, \\ & -0.75 \le \phi_i \le -0.5,\ i = 1,\ldots 5. \end{aligned} \tag{5.8}$$

Here, ε indicates the realization of a Gaussian zero-mean white noise variable with a standard deviation of 0.005 ns. The group delay response of the cascade of ring resonators for different values of the objective function in (5.8) can be seen in Figure 5.6.

Figure 5.7 shows ten separate runs and their geometric average of the CDONE and the DONE algorithm. The variables κ_i, ϕ_i were scaled to the region $[0,1]$ before being passed to the algorithms. Unlike in the previous experiment, the behavior of both algo-

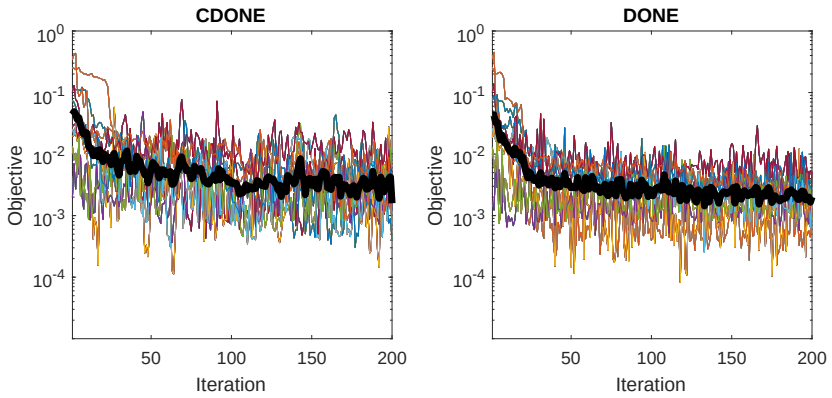


Figure 5.7: Ten runs of the CDONE and the DONE algorithm on the minimization problem (5.8). The thick black line shows the geometric mean of the ten runs.

rithms is similar. The following settings were used for the CDONE algorithm: number of basis functions $D = 1000$, number of measurements $N = 200$, regularization parameter $\lambda = 10^{-7}$, and the variance of the exploration parameter was set to $4 \cdot 10^{-4}$. The DONE algorithm used the same settings except for $\lambda = 10^{-1}$, and the $\mathbf{w}_k$ parameters were drawn from a normal distribution with standard deviation $\sigma = 1.5$.

As mentioned in Section 5.3.1, the CDONE algorithm is less sensitive to the regularization parameter λ than the DONE algorithm because it uses a convex surrogate model. We claim that this is also the case for σ, the parameter that scales the $\mathbf{w}_k$ parameters, which is equivalent to a scaling of the input. We verified this claim on the photonic beamformer application, by looking at the value of the objective function at the final iteration, averaged over ten runs, for 100 different combinations of λ and σ. These combinations were chosen randomly in such a way that the base 10 logarithms of λ and σ had a $[-8, 1]$ and a $[-3, 1]$ uniform distribution respectively.

Figure 5.8 shows the 100 results of the average objective value at the final iteration for different combinations of σ and λ for the CDONE and the DONE algorithm by means of a contour plot. Values between the 100 combinations were interpolated. Though both algorithms can achieve similar average objective values by choosing σ and λ correctly, the CDONE algorithm has a much larger viable region. The DONE algorithm needs to be tuned more precisely to get similar results, and because of the complicated shape of the plot it also seems to be much more difficult to search for the right values of λ and σ. This shows that CDONE is easier to use in practice.

5.6. CONCLUSION

The DONE algorithm, a derivative-free optimization algorithm for finding the minimum of an objective using noisy measurements, has been adapted by introducing rectified linear units and a nonnegativity constraint. The constraint makes sure that the surrogate model of the objective is convex, allowing its global minimum to be found with convex optimization algorithms. The adapted CDONE algorithm has less hyper-parameters

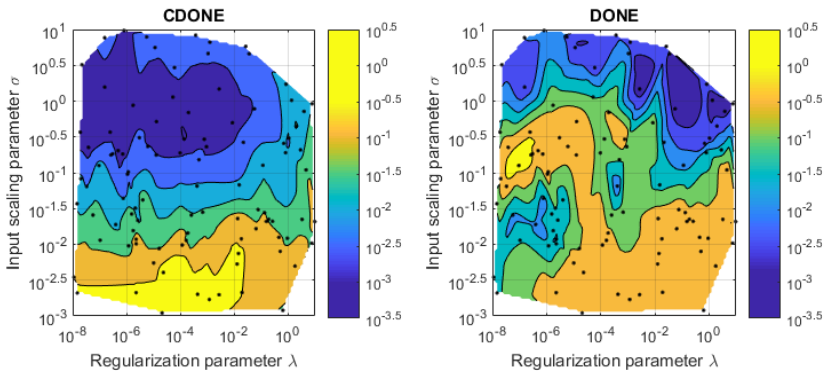


Figure 5.8: Contour plot of the effect of the λ and σ parameters on the outcome of the CDONE and DONE algorithms. The color indicates the objective value of the minimization problem 5.8 at iteration 200, averaged over 10 runs. The black dots show which random combinations of λ and σ were evaluated, and the rest of the contour plot was calculated using a cubic interpolation.

to tune, since the convexity helps in reducing the need for exploration of the surrogate model, and it is experimentally verified that it is less sensitive to the remaining hyper-parameters. Having less hyper-parameters to tune is crucial for certain tasks, especially when the algorithm is used for finding the optimal hyper-parameters of another algorithm or simulation. Furthermore, the surrogate model benefits from sparsity and can be evaluated efficiently.

The CDONE algorithm has been tested on an artificial example, on the problem of hyper-parameter optimization for a deep neural network classifier for handwritten digits, and on the problem of photonic beamformer tuning. Using a lower effective number of basis functions because of the sparsity, the CDONE algorithm still exhibited high final accuracy. In the future we will further exploit this sparsity in efficient implementations of all steps of the algorithm.

REFERENCES

[1] L. Bliek, M. Verhaegen, and S. Wahls, *Online function minimization with convex random relu expansions,* in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)* (2017) pp. 1–6.

[2] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, Vol. 8 (Siam, 2009).

[3] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient global optimization of expensive black-box functions,* J. Global Optim. **13**, 455 (1998).

[4] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, *Algorithms for hyper-parameter optimization,* in *Adv. Neur. In.* (2011) pp. 2546–2554.

[5] J. Mockus, *Bayesian approach to global optimization: theory and applications*, Vol. 37 (Springer Science & Business Media, 2012).

[6] L. Bliek, H. R. G. W. Verstraete, M. Verhaegen, and S. Wahls, *Online optimization with costly and noisy measurements using random Fourier expansions,* IEEE Transactions on Neural Networks and Learning Systems **29**, 167 (2018).

[7] A. Rahimi and B. Recht, *Uniform approximation of functions with random bases,* in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on* (IEEE, 2008) pp. 555–561.

[8] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, *Maxout networks,* in *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 28 (PMLR, Atlanta, Georgia, USA, 2013) pp. 1319–1327.

[9] A. Rahimi and B. Recht, *Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning,* in *Adv. Neur. In.* (2009) pp. 1313–1320.

[10] S. Wright and J. Nocedal, *Numerical optimization,* Springer Science **35**, 67 (1999).

[11] Y. Zhu, X. R. Li, *et al.*, *Recursive least squares with linear constraints,* Communications in Information & Systems **7**, 287 (2007).

[12] J. Chen, C. Richard, J. C. M. Bermudez, and P. Honeine, *Nonnegative least-mean-square algorithm,* IEEE Transactions on Signal Processing **59**, 5225 (2011).

[13] V. H. Nascimento and Y. V. Zakharov, *Rls adaptive filter with inequality constraints,* IEEE Signal Processing Letters **23**, 752 (2016).

[14] K. Engel and S. Engel, *Recursive least squares with linear inequality constraints,* Optimization and Engineering **16**, 1 (2015).

[15] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),* arXiv e-prints , arXiv:1511.07289 (2015), arXiv:1511.07289 [cs.LG] .

[16] *MATLAB Central File Exchange, DONE algorithm,* www.mathworks.com/matlabcentral/fileexchange/61288-done-algorithm, retrieved June 6 (2017).

[17] *The MathWorks Inc., Create simple deep learning network for classification,* www.mathworks.com/help/nnet/examples/create-simple-deep-learning-network-for-classification.html, retrieved June 6 (2017).

[18] P. Pozzi, D. Wilding, O. Soloviev, H. Verstraete, L. Bliek, G. Vdovin, and M. Verhaegen, *High speed wavefront sensorless aberration correction in digital micromirror based confocal microscopy,* Optics Express **25**, 949 (2017).

[19] R. C. Hansen, *Phased array antennas*, Vol. 213 (John Wiley & Sons, 2009).

[20] D. Marpaung, C. Roeloffzen, R. Heideman, A. Leinse, S. Sales, and J. Capmany, *Integrated microwave photonics,* Laser & Photonics Reviews **7**, 506 (2013).

[21] A. Meijerink, C. G. Roeloffzen, R. Meijerink, L. Zhuang, D. A. Marpaung, M. J. Bentum, M. Burla, J. Verpoorte, P. Jorna, A. Hulzinga, *et al., Novel ring resonator-based integrated photonic beamformer for broadband phased array receive antennas—part I: Design and performance analysis,* J. Lightwave Technol. **28**, 3 (2010).

[22] L. Zhuang, C. G. Roeloffzen, A. Meijerink, M. Burla, D. A. Marpaung, A. Leinse, M. Hoekman, R. G. Heideman, and W. van Etten, *Novel ring resonator-based integrated photonic beamformer for broadband phased array receive antennas—part II: Experimental prototype,* Journal of lightwave technology **28**, 19 (2010).

6

CONCLUSION

This final chapter gives conclusions and recommendations for the automatic tuning of photonic beamformers and for the DONE algorithm and its variants. It also describes the improvements over existing methods, and provides a short discussion of recently emerged relevant methods. This chapter ends with recommendations for future work.

6.1. IMPROVEMENTS OVER EXISTING METHODS

This thesis presented a novel data-driven approach to the problem of automatically tuning a photonic beamformer using the Data-based Online Nonlinear Extremum-seeker (DONE) algorithm. The DONE algorithm makes use of measurements from the beamformer system to approximate the relation between the system actuators and the objective to be minimized, namely the difference between the measured time delays and the desired time delays. The approximation is done with random Fourier expansions, using recursive linear least squares to update the approximation every time a new measurement comes in. The algorithm is especially designed to be able to deal with noise, slow or costly measurements, and objectives for which the derivative cannot be computed or approximated.

In Chapter 2 it was shown that earlier approaches based on physical beamformer models using nonlinear optimization techniques can be very sensitive to model errors. In the presence of model errors, performing nonlinear optimization on a model derived from data has been shown to give better results. It was also shown that no artificial local minima are introduced when using random feature expansions to approximate the data, as long as enough features are used.

In Chapter 3, the proposed data-driven approach was compared with other existing data-driven approaches like Bayesian optimization algorithms on a simulation of a photonic beamformer. Compared to Bayesian optimization algorithms, the DONE algorithm has the advantage of a fixed computational complexity per iteration, while Bayesian optimization algorithms become slower over time. For the beamformer simulation, this led to a speed-up factor of over 1000. We also provided theoretical results on the approximation capabilities of the random Fourier expansion: even when trained with linear least squares, it can approximate square integrable functions arbitrarily well if the number of basis functions is large enough. The hyper-parameters of the DONE algorithm were also investigated, and we provided both theoretical results and practical rules of thumb for them. Finally, the DONE algorithm has been applied to other applications, namely a benchmark optimization problem, optical coherence tomography, and a robot arm. This last application was mainly to test the large-scale capabilities of DONE: it was successfully applied to a problem with 150 degrees of freedom. We can conclude that DONE is a fast and powerful online optimization algorithm for those problems where function evaluations are costly and noisy and where no derivatives of the objective function are available.

In Chapter 4, the DONE algorithm was applied to a real photonic beamformer system. A full transmit phased array antenna with both radio-frequency and photonic beamformers was presented, as well as the chip design of the fully integrated photonic beamformer. Requirements for the system were also given. The goal of the system is to provide satellite Internet connections on an aircraft by sending signals through the K_u-band in the 14-14.5 GHz range. It was shown how an optical delay line based on optical ring resonators was automatically configured over a 0.5 GHz bandwidth using the DONE algorithm, providing a delay of approximately 0.4 ns. The solution found by the DONE algorithm satisfied the requirements: the corresponding phase response had a ripple of less than 11.25° over the 0.5 GHz bandwidth. This is the first time that any automatic tuning algorithm has been applied to this type of transmit antenna. Two other subsys-

tems of the beamformer system, namely an optical sideband filter and a separate carrier tuning subsystem, were also tuned, the former with the DONE algorithm and by hand, the latter only by hand. Without these two subsystems, the required bandwidth would be increased up to 29 GHz, which would severely increase the system complexity and the difficulty of tuning the system.

In Chapter 5, the DONE algorithm itself has been improved by restricting the surrogate model to be convex, giving rise to the CDONE algorithm. This is done by using rectified linear units (ReLUs) instead of cosines as basis functions and imposing a non-negativity constraint on the weights. ReLUs have seen a lot of success in the context of deep learning lately. It was shown that the CDONE algorithm could benefit from the inherent sparsity and cheap evaluation of the ReLUs, allowing for a more efficient implementation than the one used in the DONE algorithm. Furthermore, the convexity of the surrogate model gives a number of advantages, such as getting rid of certain hyper-parameters, less sensitivity to other hyper-parameters, more stable convergence behavior, and guarantees for finding the global optimum of the surrogate model. The CDONE algorithm was applied to a toy function, to the problem of hyper-parameter tuning for deep learning, and to a photonic beamformer simulation. Sparsity of the model was investigated, as well as the sensitivity to two hyper-parameters of DONE and CDONE: the regularization parameter and an input scaling parameter. It can be concluded that CDONE is easier to use than DONE, as it requires less knowledge to set up correctly.

In the Appendix A, another improvement to the DONE algorithm has been made. In order to be able to deal with time-varying objective functions, a sliding window principle has been applied, where only the latest few measurements are used for fitting the surrogate model. Furthermore, the algorithm has been extended with a variable offset. With this offset, the pseudo-convex shape that occurs in many objective functions can be exploited. The adapted algorithm is applied to a confocal fluorescent microscopy application with 18 degrees of freedom and compared to a hill climbing algorithm. Though both algorithms achieve similar accuracy in the end, the hill climbing algorithm takes measurements far away from the current best solution, while the adapted DONE algorithm remains stable, allowing the system to be used while the algorithm is running.

6.2. CRITERIA

Chapter 1 gave some criteria for the automatic tuning method used in tuning a photonic beamformer. Here, we check if the criteria are satisfied.

THE METHOD SHOULD TAKE HEATER CROSSTALK INTO ACCOUNT.

Since the proposed method automatically approximates the relation between the system actuators (the heaters) and the objective, heater crosstalk is automatically taken into account. The surrogate that was used in the approximation, namely a random Fourier expansion, is general enough to allow for all system actuators together to influence the objective in a nonlinear fashion. In Chapter 3, the DONE algorithm was successfully applied to a simulation of a photonic beamformer that included heater crosstalk, while in Chapter 4 the same algorithm was successfully applied to a real photonic beamformer where heater crosstalk was also present.

THE METHOD SHOULD NOT BE SENSITIVE TO MODEL PARAMETERS.

Since the proposed method does not make use of the physical photonic beamformer model, it is not sensitive to these kinds of parameters. However, like with most algorithms, the outcome of the DONE algorithm depends on its own hyper-parameters, such as the number of basis functions or the probability distributions of the random parameters. In Chapter 3 we investigated these hyper-parameters and gave some theoretical results as well as rules of thumb, while in Chapter 5 we developed a variant of the DONE algorithm that has less hyper-parameters to tune.

IF FEEDBACK FROM MEASUREMENTS WILL BE USED, THE METHOD SHOULD BE ABLE TO OPERATE WITH SCALAR-VALUED MEASUREMENTS AND NOT BE SENSITIVE TO MEASUREMENT NOISE.

The proposed method indeed uses system measurements to continuously improve its surrogate function. As far as noise is concerned, the DONE algorithm is especially designed to be able to deal with measurement noise by using regularized linear least squares for training the surrogate model. It already assumes scalar-valued measurements and should therefore have no problem with other scalar-valued objectives like the signal-to-noise ratio or the output power of a beamformer system.

THE NUMBER OF MEASUREMENTS USED SHOULD BE AS LOW AS POSSIBLE TO PREVENT THE METHOD FROM BEING TOO SLOW.

The DONE algorithm assumes that there is some cost, such as time, associated with each measurement. Therefore, it aims at keeping the number of measurements as low as possible. This is done by using a surrogate function which efficiently approximates multi-dimensional objective functions with a low number of measurements, and by only taking measurements near the minimum of the surrogate function rather than focusing too much on exploring the entire search space.

THE METHOD SHOULD OPERATE IN REAL TIME.

The DONE algorithm is an online algorithm, meaning every time a new measurement comes in, it updates the surrogate function and calculates a new set of heater voltages to try next. Depending on the implementation and the application, this should not take more than a few milliseconds (from Table 3.2 in Chapter 3 it can be concluded that the DONE algorithm takes about $1-30$ ms per iteration, depending on the application). This is sufficiently fast for the current application, since the time it takes to apply the heater voltages and the time it takes to perform a measurement are both in the millisecond range.

It should be noted that applying the DONE algorithm to the real photonic beamformer system described in Chapter 4 currently takes about 3 to 5 seconds per iteration, not milliseconds, mainly because of the communication between the beamformer, the computer running the DONE algorithm, and the vector network analyzer as can be seen in Figure 4.8 in the same chapter. These three systems also used different types of software. The software and the measurement set-up could be made more efficient in the future.

CRITERIA THAT DEPEND ON THE EXACT APPLICATION AND BEAMFORMING SYSTEM THAT IS USED.

Finally, Chapter 4 gave some more specific criteria for the beamformer system described there, see Section 4.2.1. These requirements were satisfied: the 1×4 beamformer was automatically tuned, using the DONE algorithm, in such a way that the corresponding phase response had a maximum error of less than 11.25° over a bandwidth of 0.5 GHz. This was satisfied for two different delay settings, namely a delay of 408 ps and a delay of 250 ps.

6.3. COMPARISON WITH RECENTLY DEVELOPED METHODS

Chapter 1 also mentioned two recently developed methods that are relevant to this thesis. These two methods were not considered in this thesis as they had not yet been published at the time the research in this thesis was conducted. To get more insight in the similarities and differences between these methods and the methods described in this thesis, this section gives a short discussion.

6.3.1. AUTOMATIC TUNING OF A MACH-ZEHNDER INTERFEROMETER-BASED PHOTONIC BEAMFORMER

In [1], a photonic beamformer based on Mach-Zehnder interferometers was tuned automatically using a derivative-free optimization algorithm. However, the type of photonic beamformer that was investigated in this thesis is based on optical ring resonators. The beamformer considered in this thesis can provide a large delay (in the order of hundreds of ps) over a large bandwidth by using a cascade of optical ring resonators as delay elements. By adding more ring resonators to the system, the maximum delay and bandwidth can be increased, and the ripple can be decreased, at the cost of increasing the system complexity. The beamformer in [1] also deals with large bandwidths, but the considered delays are much smaller (in the order of tens of ps) and there is no way to decrease the ripple as only one Mach-Zehnder interferometer is used for each antenna element. Furthermore, it is not possible to use the automatic tuning method in [1] on the system in this thesis because of the complexity of the system. On the other hand, applying the DONE algorithm to the system in [1] would probably work, but it is expected that it would only lead to improvements in the presence of model errors, as discussed in Chapter 2.

6.3.2. COMMON BAYESIAN OPTIMIZATION LIBRARY (COMBO)

In [2], a Bayesian optimization method that makes use of random features was developed. The method was applied to the problem of determining the atomic structure of a crystalline interface. As the method is similar to the DONE algorithm described in this thesis, we show some similarities and differences here. COMBO and DONE share the following similarities:

- Both methods approximate the Gaussian kernel using random Fourier features.
- Both methods use a covariance matrix based on a regularized least squares criterion.

- Both methods are online, in the sense that the regularized least squares problem is not calculated every iteration, but the covariance matrix and the weights are updated efficiently instead.
- Both methods avoid computing the predictive variance.

The following differences between the two methods were found:

- In finding the minimum of the surrogate model, COMBO assumes a finite number of potential candidate points in the search space and finds the best candidate using Thompson sampling, while DONE solves a continuous optimization problem on what would be considered the mean of the surrogate model from a Bayesian point of view. Both methods avoid the slower approach of using an acquisition function that depends on the predictive variance, which is common in Bayesian optimization algorithms.
- The efficient update of the covariance matrix in COMBO is based on Cholesky decompositions, while the efficient update of the same matrix in DONE is based on an inverse QR decomposition. Essentially, DONE keeps track of the inverse covariance matrix, while COMBO keeps track of the covariance matrix and avoids using the inverse by solving triangular systems of equations. Both approaches have the same order of complexity but might differ in numerical stability.
- DONE uses an exploration heuristic to avoid local minima, while COMBO performs global optimization on a finite set of candidate points. Note that the number of candidate points scales exponentially with the input dimension in the case of discrete variables.
- DONE provides rules of thumb in choosing the hyper-parameters, while COMBO implements an online optimization algorithm to choose the hyper-parameters automatically after initialization and every few iterations.
- COMBO never evaluates the same candidate point twice, increasing efficiency but making it unsuited for objective functions that suffer from noise. DONE does not use this restriction.
- The largest (in terms of input dimension) problem DONE has been applied to is a problem with 150 degrees of freedom (see Chapter 3), where each variable has a continuous range of possible values, while the largest problem COMBO has been applied to was a problem with 65536 candidate points [3], which corresponds to 16 degrees of freedom in case the variables are binary.

Overall, COMBO seems more fit for discrete optimization problems with a finite number of possible solutions (and a low input dimension), while DONE seems more fit for continuous optimization problems.

6.4. RECOMMENDATIONS FOR FUTURE WORK

In science, each new discovery also leads to new questions. In this section we discuss some of the possible directions for further research on the topic of automatic tuning of photonic beamformers and the DONE algorithm.

6.4.1. FULLY AUTOMATIC PHOTONIC BEAMFORMER SYSTEM

A part of the photonic beamformer system presented in Chapter 4 was tuned automatically with the DONE algorithm. In order to fully automate the whole system, some more work has to be done. First of all, the scale of the system has to be considered. The beamformer in Chapter 4 contains 1536 paths in total, of which only one was tuned automatically, while in Chapter 3 eight beamformer paths were tuned at the same time. In Chapter 3 the DONE algorithm was also applied to a problem with 150 degrees of freedom, but the full beamformer system from Chapter 4 contains thousands of actuators. One possible solution is to solve the problem in a distributed manner. If the performance of each individual beamformer in Figure 4.2 in Chapter 4 can be measured separately, for example by measuring the input and output signal of each individual beamformer, then the DONE algorithm could be applied to each beamformer separately as well. This greatly reduces the scale of the problem to that of tuning a 1×24 beamformer, which consists of a 1-to-24 splitter and five optical ring resonators for each of the 24 paths. The total number of actuators for this beamformer is at most 263, while the other beamformers contain 30 actuators or less. This is still more than the 150 degrees of freedom mentioned earlier, but should be possible with more computational power and with a more efficient variation of the DONE algorithm.

Another part of the system that should be considered is the optical sideband filter. This subsystem was also tuned automatically in Chapter 4, but it was noted that the results could be improved by choosing a different objective function. The mean square error criterion that was used does not work too well with the logarithmic scale of the measurements. In the future, different objective functions should be investigated.

Finally, the separate carrier tuning subsystem was not tuned automatically yet. The reason for this is that the measurement set-up needed to be changed slightly in order to get the required measurements (using a frequency sweep instead of a time sweep). If all measurements would be done with this set-up, including the optical beamforming network and optical sideband filter measurements, then it might be possible to automate the whole system. In order to avoid the problems that are common in multi-objective optimization, the three subsystems should not be tuned simultaneously but rather one at a time, starting with the optical sideband filter, followed by the optical beamforming network, and then the separate carrier tuning subsystem.

This fully automated approach still requires measurements using a vector network analyzer. This is not desired in real life applications. One possible solutions is to use the solution from this approach as an initial guess for a second similar automated approach that does not use a vector network analyzer. This second approach would only use measurements based on the output power or signal-to-noise ratio of the beamformer system. In order for this second approach to be applied to moving vehicles like aircrafts, the Sliding-Window DONE algorithm described in Appendix A could be used to deal with the time-varying beam angle. A big advantage of this approach is that the beam angle itself is not required: the algorithm is capable of adapting to changes in the objective function by means of the sliding window principle.

6.4.2. CDONE

The CDONE algorithm described in Chapter 5 has many advantages over the DONE algorithm, such as sparsity, a quick to evaluate surrogate model, convexity, and less hyperparameters. To make full use of these advantages, an efficient implementation is required. The current implementation of CDONE does not use a recursive least squares implementation, but several possibilities for performing recursive least squares with a nonnegativity constraint were given. The most promising one is the approach found in [4]. An efficient implementation of that approach in the CDONE algorithm that also makes use of the sparsity, and that possibly also makes use of the adaptations made in Appendix A, should lead to a very efficient, robust, adaptive and easy to use algorithm. Since Bayesian optimization is an on-going topic of research, this new algorithm should then be compared to newer types of Bayesian optimization algorithms as well, such as the COMBO algorithm which was mentioned in this chapter.

Finally, another possible line of research to consider is the use of dedicated hardware for the DONE algorithm or one of its variants. A first attempt has been made using a graphical processing unit in [5], making the algorithm approximately twice as fast. Further improvements could be made with better graphical processing units, field-programmable gate arrays, or even optical computers. Particularly interesting would be the question whether hardware based on integrated photonics, such as the examples described in [6], can be used to implement CDONE. If it can, the next question is whether this type of hardware can somehow be combined with the integrated microwave photonics technology described in this thesis, such that both the signal processing and the automatic tuning software are all realized on one chip.

REFERENCES

[1] V. C. Duarte, M. V. Drummond, and R. N. Nogueira, *Coherent photonic true-time-delay beamforming system for a phased array antenna receiver,* in *2016 18th International Conference on Transparent Optical Networks (ICTON)* (2016) pp. 1–5.

[2] T. Ueno, T. D. Rhone, Z. Hou, T. Mizoguchi, and K. Tsuda, *Combo: An efficient bayesian optimization library for materials science,* Materials Discovery **4**, 18 (2016).

[3] S. Ju, T. Shiga, L. Feng, Z. Hou, K. Tsuda, and J. Shiomi, *Designing nanostructures for phonon transport via bayesian optimization,* Physical Review X **7**, 021024 (2017).

[4] K. Engel and S. Engel, *Recursive least squares with linear inequality constraints,* Optimization and Engineering **16**, 1 (2015).

[5] J. Munnix, *Parallel Approach to Derivative-Free Optimization: Implementing the DONE Algorithm on a GPU*, Master's thesis, Delft University of Technology (2016).

[6] G. Van der Sande, D. Brunner, and M. C. Soriano, *Advances in photonic reservoir computing,* Nanophotonics **6**, 561 (2017).

A

THE SLIDING-WINDOW DONE ALGORITHM

The DONE algorithm minimizes an unknown objective that can be evaluated only with expensive and/or noisy measurements. In many practical applications, the objective also varies over time. To deal with these types of objectives we extend the DONE algorithm with a sliding window. Compared to the use of a forgetting factor, the sliding window has the advantage that the regularization remains constant. We also extend the algorithm with a variable offset, which exploits the fact that many objectives have a (pseudo)convex shape in practice. The Sliding-Window DONE algorithm (SW-DONE) is demonstrated on a confocal fluorescent microscopy application and compared to a hill climbing algorithm. Unlike the hill climbing algorithm, SW-DONE takes measurements close to the found optimum, ensuring a high quality image at all times.

This chapter is based on unpublished work that was done in collaboration with H.R.G.W. Verstraete, P. Pozzi, S. Wahls, and M. Verhaegen.

A

A.1. INTRODUCTION

In many applications, an objective that can be measured but is not known in closed-form has to be minimized. For example, in adaptive optics the aberrations of an optical system are being compensated to improve the performance of the system [1]. Measurement noise and a large measurement time can prevent the objective to be minimized with standard derivative-based optimization techniques.

Recently, the DONE algorithm has been proposed to find the minimum of these types of objectives [2, 3]. The algorithm consists of a combination of recursive least squares (RLS), random Fourier expansions (RFEs) [4], and nonlinear optimization. The algorithm was shown to outperform other derivative-free optimization (DFO) methods like NEWUOA [5] and coordinate search.

Besides noise and expensive function evaluations, another challenge arises in many practical applications. The objective function that is to be minimized might change over time due to physical or parametric changes in a system or simulation. For example, aging components in a setup might behave differently over time, and focusing on different areas of a sample in microscopy can cause different wavefront aberrations. In this case, not all available measurements should be used for training the model, but only the most recent ones. In the RLS part of the DONE algorithm, a forgetting factor could be introduced, but this can lead to the problem of estimator wind-up where the change in parameters becomes extremely large with each new measurement [6, 7]. A common way to prevent this problem is the use of directional forgetting [8], where old data is forgotten 'in the direction of the new data'. With this approach, however, it would not be clear which measurements are used and which are forgotten in the RLS estimation, making it possible that obsolete or incorrect measurements are used in the model. Another way to prevent estimator wind-up would be to make use of Tikhonov regularization and increase the regularization parameter at each iteration, to compensate for the forgetting factor. Unfortunately, to our knowledge there is no way to make the regularization parameter adaptive without resorting to inefficient (cubic complexity) or unstable algorithms or algorithms similar to directional forgetting [9].

In this paper we propose a simpler technique to deal with objective functions that change over time, based on a sliding window principle as demonstrated in Fig. A.1. At the cost of more memory storage, the problem of estimator wind-up is avoided. We also provide an extension that lets the DONE algorithm exploit the pseudo-convexity of the objective. Both extensions are demonstrated on a real time aberration correction technique in fluorescence microscopy [10] and compared to a hill climbing algorithm [11].

A.1.1. THE DONE ALGORITHM

The DONE algorithm[2, 3] tries to find the minimum of a function f by approximating it with an RFE and then applying standard optimization techniques to this RFE model. An RFE is a function g of the form

$$g(\mathbf{x}) = \sum_{k=1}^{D} c_k \cos(\omega_k^T \mathbf{x} + b_k),$$

with D the number of Fourier expansions and b_k and ω_k realizations of random variables with continuous probability distributions. Practical ways to choose the proba-

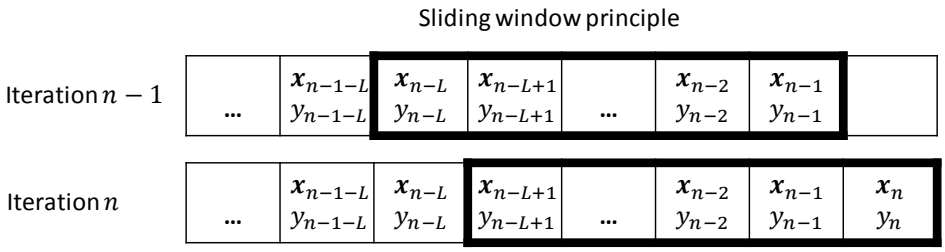


Figure A.1: At each iteration in the sliding window RLS algorithm the oldest measurement is removed before adding a new measurement.

Algorithm 5 DONE Algorithm

1: Get measurement y_n at point $\mathbf{x}_n$.
2: Update RFE coefficients with RLS.
3: Find minimum of RFE.
4: New measurement point is minimum of RFE plus small perturbation. Repeat from Step 1.

bility distributions are discussed in [3]. Using measurement y_n at point $\mathbf{x}_n$, the coefficients c_k can be calculated using an RLS algorithm on the transformed inputs $\mathbf{a}_n = [\cos(\omega_1^T \mathbf{x}_n + b_1) \cdots \cos(\omega_D^T \mathbf{x}_n + b_D)]$. We use an inverse QR [12, Sec. 21] version of the RLS algorithm as follows: find a rotation matrix Θ_n that lower triangularizes the upper triangular matrix in Eq. (A.1) below and generates a post-array with positive diagonal entries:

$$\begin{bmatrix} 1 & \mathbf{a}_n \mathbf{P}_{n-1}^{1/2} \\ \mathbf{0} & \mathbf{P}_{n-1}^{1/2} \end{bmatrix} \Theta_n = \begin{bmatrix} \gamma_n^{-1/2} & \mathbf{0} \\ \mathbf{g}_n \gamma_n^{-1/2} & \mathbf{P}_n^{1/2} \end{bmatrix}. \tag{A.1}$$

with initialization $\mathbf{P}_0 = \lambda^{-1} \mathbf{I}_{D \times D}$, a diagonal matrix. Here, $\lambda \in \mathbb{R}$ is a regularization parameter. The updated weights of the RFE model can then be found with

$$\mathbf{c}_n = \mathbf{c}_{n-1} + \mathbf{g}_n (y_n - \mathbf{a}_n \mathbf{c}_{n-1}), \tag{A.2}$$

with initialization $\mathbf{c}_0 = \mathbf{0}$. Here, $\mathbf{c}_n \in \mathbb{R}^D$ contains the coefficients c_k of the RFE model at step n of the algorithm. The rotation matrix Θ_n can be found by performing a QR decomposition of the transpose of the matrix on the left hand side of (A.1), or by the procedure explained in [12, Sec. 21].

The next measurement point $\mathbf{x}_{n+1}$ is determined by finding the minimum of the RFE model using standard nonlinear optimization methods, such as the L-BFGS method [13], and adding a small perturbation to it for exploration purposes. This is summarized in Algorithm 5.

A.2. SLIDING WINDOW DONE

When the DONE algorithm is extended with a sliding window, only a fixed maximum number L of past measurements are taken into account by the RFE model. If a newer

measurement becomes available, the oldest function evaluation in the sliding window is removed from the RFE model as demonstrated in Fig. A.1. Hence, the model will adapt to the latest L evaluations of the changing function f.

For the first L steps, the procedure is identical to the previous section. From step $n = L+1$ onward, we remove the oldest measurement from the RFE model before the model is updated. This is done with an inverse QR downdate using hyperbolic Givens rotations [14, Sec. 14.9]. This can be done by finding a $(1 \oplus -I)$ unitary matrix[1] $\check{\Theta}_n$ that triangularizes the pre-array shown below and generates a post-array with positive diagonal entries. Then the post array is given by

$$\begin{bmatrix} 1 & \mathbf{a}_{n-L}\mathbf{P}_{n-1}^{1/2} \\ \mathbf{0} & \mathbf{P}_{n-1}^{1/2} \end{bmatrix} \check{\Theta}_n = \begin{bmatrix} \check{\gamma}_{n-1}^{-1/2} & \mathbf{0} \\ \check{\mathbf{g}}_{n-1}\check{\gamma}_{n-1}^{-1/2} & \check{\mathbf{P}}_{n-1}^{1/2} \end{bmatrix}. \tag{A.3}$$

The downdated weights of the RFE model are then given by

$$\check{\mathbf{c}}_{n-1} = \mathbf{c}_{n-1} - \check{\mathbf{g}}_{n-1}(y_{n-L} - \mathbf{a}_{n-L}\mathbf{c}_{n-1}). \tag{A.4}$$

The matrix $\check{\Theta}_n$ does not need to be explicitly calculated, the triangularization can be completed by a series of hyperbolic Given's rotations [14, Sec. 14.A].

The downdated weights $\check{\mathbf{c}}_{n-1}$ and the downdated $\check{\mathbf{P}}_{n-1}^{1/2}$ can now be used in the update procedure (A.1)-(A.2) by replacing $\mathbf{c}_{n-1}$ and $\mathbf{P}_{n-1}^{1/2}$ respectively.

A.3. VARIABLE OFFSET

In many practical applications, the objective function to be minimized has a (pseudo)-convex shape. In this section we will adapt the DONE algorithm so that it exploits this shape with a solution that is in harmony with the online nature of the algorithm.

Regularization is a key aspect of the model fitting procedure in the DONE algorithm. It can help prevent overfitting and deal with noise sensitivity and ill-conditioning [15]. However, it also has the effect of pulling the weights of the basis functions towards zero. Therefore, the output of the RFE model is also pulled towards zero, especially in regions where less measurements have been taken.

To illustrate, consider the approximation of the function $f(x) = x^2 + 3$. Figure A.2a shows the RFE model fit on measurements of this function. Even after a measurement outside the set of previous measurements is introduced and is used to update the model, we see that the model is already turning towards zero near this point because of the regularization. A local maximum and minimum are introduced that do not appear in the original function. This is detrimental to step 3 of Algorithm 5, where the optimization algorithm that finds a minimum of the RFE model can get stuck in the artificial local minimum.

We propose to add a negative offset to all the measurements. If we ensure that the function lies below zero on its domain, as illustrated in Figure A.2b, we see that no artificial local maxima and minima are introduced. Although the RFE approximation is still a bad approximation outside the region where the measurements were taken, the

[1] A $(1 \oplus -I)$ unitary matrix Θ satisfies $\Theta \begin{bmatrix} 1 & \\ & I \end{bmatrix} \Theta^T = \begin{bmatrix} 1 & \\ & I \end{bmatrix}$.

shape of the RFE model is still similar to the shape of the true function. It is expected that derivative-based optimization methods on the RFE model will give similar results as using the same methods on the true function in this case. Therefore, we add an offset $v < -f_{\max}$ to all the measurements, where $f_{\max}$ is the maximum of f over the domain of interest.

A.3.1. IMPLEMENTATION OF A VARIABLE OFFSET

A potential problem with the approach of the previous section is that the maximum value of the unknown function f is not known beforehand. And if we just take the maximum value out of all previous measurements, this value will change as soon as a measurement with a higher value is encountered. Our solution is to use a variable offset that can change every iteration.

In Section A.2, only the weights $\mathbf{c}_n$ depend on the measurements y_n, so only (A.2) and (A.4) would be affected by an offset v. To change these equations, we keep track of a 'history' variable $\mathbf{h}_n$ that satisfies

$$\mathbf{h}_n = \mathbf{h}_{n-1} + \mathbf{g}_n(1 - \mathbf{a}_n\mathbf{h}_{n-1}), \tag{A.5}$$

with $\mathbf{h}_0 = \mathbf{0}$. Only a vector of the same length as $\mathbf{c}_n$ is required for this history variable. Note that the above is just a standard RLS update rule for measurements with value 1, so this can be seen as approximating a constant function alongside the original function f.

Since the offset v is added to all previous measurements, the weights $\mathbf{c}_n$ need to be adapted for all iterations by adding an unknown vector Δ_n. For the update rule (A.2) this gives:

$$\begin{aligned}\mathbf{c}_n + \Delta_n &= \mathbf{c}_{n-1} + \Delta_{n-1} + \mathbf{g}_n(y_n + v - \mathbf{a}_n\mathbf{c}_{n-1} - \mathbf{a}_n\Delta_{n-1}) \\ &= \mathbf{c}_{n-1} + \mathbf{g}_n(y_n - \mathbf{a}_n\mathbf{c}_{n-1}) \\ &\quad + \Delta_{n-1} + \mathbf{g}_n(v - \mathbf{a}_n\Delta_{n-1}).\end{aligned} \tag{A.6}$$

Using (A.2) in the above gives the rule

$$\Delta_n = \Delta_{n-1} + \mathbf{g}_n(v - \mathbf{a}_n\Delta_{n-1}), \tag{A.7}$$

or equivalently

$$\Delta_n = v\mathbf{h}_n. \tag{A.8}$$

In other words, to approximate a function $\tilde{f} = f + v$ while making use of the least square weights $\mathbf{c}_n$ used in the approximation of f, it suffices to add $v\mathbf{h}_n$ to these weights, provided $\mathbf{h}_n$ has been updated each iteration as in (A.5). This vector has length D, so this is the additional memory required to make use of a variable offset. The computational complexity of (A.5) is $O(D)$, so the order of complexity of the whole algorithm does not increase.

For the downdate, from step $n = L+1$ onward, we follow a similar procedure. Let

$$\check{\mathbf{h}}_{n-1} = \mathbf{h}_{n-1} - \check{\mathbf{g}}_{n-1}(1 - \mathbf{a}_{n-L}\mathbf{h}_{n-1}). \tag{A.9}$$

Now, the weights $\check{\mathbf{c}}$ need to be adapted for all iterations by adding an unknown vector $\check{\Delta}_n$. For the downdate rule (A.4) this gives:

$$\begin{aligned}\check{\mathbf{c}}_{n-1} + \check{\Delta}_{n-1} &= \mathbf{c}_{n-1} + \Delta_{n-1} \\ &\quad - \check{\mathbf{g}}_{n-1}(y_{n-L} + v - \mathbf{a}_{n-L}\mathbf{c}_{n-1} - \mathbf{a}_{n-L}\Delta_{n-1}) \\ &= \mathbf{c}_{n-1} - \check{\mathbf{g}}_{n-1}(y_{n-L} - \mathbf{a}_{n-L}\mathbf{c}_{n-1}) \\ &\quad + \Delta_{n-1} - \check{\mathbf{g}}_{n-1}(v - \mathbf{a}_{n-L}\Delta_{n-1}). \end{aligned} \tag{A.10}$$

Using (A.4) in the above gives the rule

$$\check{\Delta}_{n-1} = \Delta_{n-1} - \check{\mathbf{g}}_n(v - \mathbf{a}_{n-L}\Delta_{n-1}), \tag{A.11}$$

or equivalently

$$\check{\Delta}_{n-1} = v\check{\mathbf{h}}_{n-1}. \tag{A.12}$$

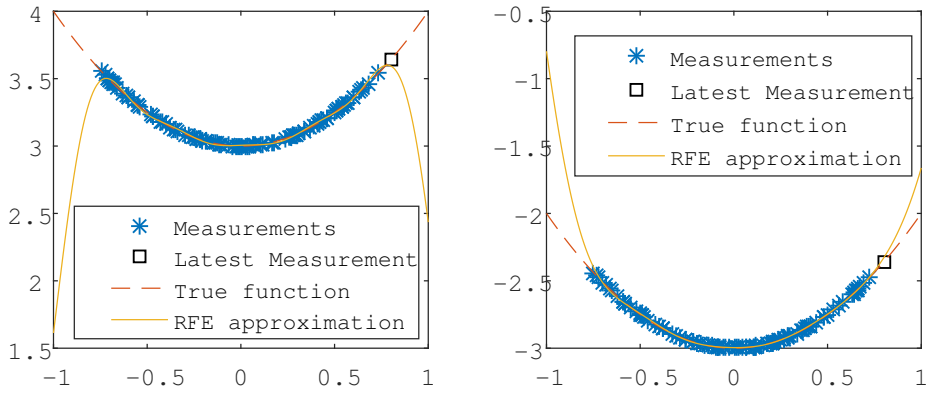
The downdated history variable $\check{\mathbf{h}}_{n-1}$ now replaces $\mathbf{h}_{n-1}$ in (A.5). Note that the update and downdate rules of the history variable are independent of the actual offset v, allowing to change the offset at any desired step of the algorithm.

To ensure that the function that is approximated stays below zero, the offset is changed as soon as a positive measurement is encountered. To prevent changing the offset too often, the offset is chosen to be -2 times the value of the positive measurement. Other values are also possible. We have chosen to change the offset between the downdate and update step. When changing the offset more than once, the difference between subsequent offsets $v_n - v_{n-1}$ is used, with $v_0 = 0$. All of this, together with the sliding window principle, is illustrated in Algorithm 6. The procedures updateRFE and downdateRFE correspond to equations (A.1)-(A.2) and (A.3)-(A.4) respectively. Using this in step 2 of Algorithm 5 leads to the SW-DONE algorithm.

A.4. ADAPTIVE OPTICS APPLICATION

The SW-DONE algorithm is demonstrated on a fluorescent microscopy application. Confocal fluorescent microscopy is an optical sectioning technique which aims to produce clear images of focal planes within a thick sample. The image quality largely depends on the aberrations present in the sample, and the ability to correct these. A real-time sensorless aberration correction technique that makes use of the SW-DONE algorithm was presented in [10]. This technique makes use of the sliding window principle and the variable offset as described in this paper, but in this paper we describe these principles in more detail and verify their effects experimentally.

The objective that is to be minimized was chosen to be the second moment of the intensity distribution of fluorescence light emitted under point like illumination. Here, the optimal value of the objective is depending on the sample used, but is empirically limited to circa $2\mu m^2$. This objective was measured at 100 Hz. The 69 actuators mirror was used to simulate combinations of the first 18 Zernike polynomials [16] (a commonly used base in aberrations space), excluding tip, tilt and defocus, therefore constraining the problem to 18-dimensional measurements $\mathbf{x}_n$. To investigate the effects of a changing objective function, three fixed artificial aberrations in the Zernike coefficients were introduced after respectively 5, 10 and 15 seconds.



(a) (b)

Figure A.2: Demonstration of a RFE model fit for (a) the original function $f(x) = x^2 + 3$ (without offset) and (b) the function $\tilde{f}(x) = f(x) - 6$ (with offset).

Algorithm 6 Sliding window with variable offset in DONE

procedure SLIDINGWINDOW($\mathbf{a}_n$, y_n, $\mathbf{a}_{n-L}$, y_{n-L}, $\mathbf{P}_{n-1}^{1/2}$, v_{n-1}, $\mathbf{h}_{n-1}$, $\mathbf{c}_{n-1}$)
  if $n \geq L+1$ **then**
    $[\check{\mathbf{P}}_{n-1}^{1/2}, \check{\mathbf{g}}_{n-1}, \check{\mathbf{c}}_{n-1}] \leftarrow$ downdateRFE($\mathbf{a}_{n-L}, \mathbf{P}_{n-1}^{1/2}, \mathbf{c}_{n-1}, y_{n-L} + v_{n-1}$)
    $\check{\mathbf{h}}_{n-1} \leftarrow \mathbf{h}_{n-1} - \check{\mathbf{g}}_{n-1}(1 - \mathbf{a}_{n-L}\mathbf{h}_{n-1})$
  else
    $\check{\mathbf{h}}_{n-1} \leftarrow \mathbf{h}_{n-1}$, $\check{\mathbf{P}}_{n-1}^{1/2} \leftarrow \mathbf{P}_{n-1}^{1/2}$, $\check{\mathbf{c}}_{n-1} \leftarrow \mathbf{c}_{n-1}$
  if $y_n + v_{n-1} > 0$ **then**
    $v_n \leftarrow -2y_n$
    $\check{\mathbf{c}}_{n-1} \leftarrow \check{\mathbf{c}}_{n-1} + (v_n - v_{n-1})\check{\mathbf{h}}_{n-1}$
  else
    $v_n \leftarrow v_{n-1}$
  $[\mathbf{P}_n, \mathbf{g}_n, \gamma_n, \mathbf{c}_n] \leftarrow$ updateRFE($\mathbf{a}_n, \check{\mathbf{P}}_{n-1}, \check{\mathbf{c}}_{n-1}, y_n + v_n$)
  $\mathbf{h}_n \leftarrow \check{\mathbf{h}}_{n-1} + \mathbf{g}_n(1 - \mathbf{a}_n\check{\mathbf{h}}_{n-1})$

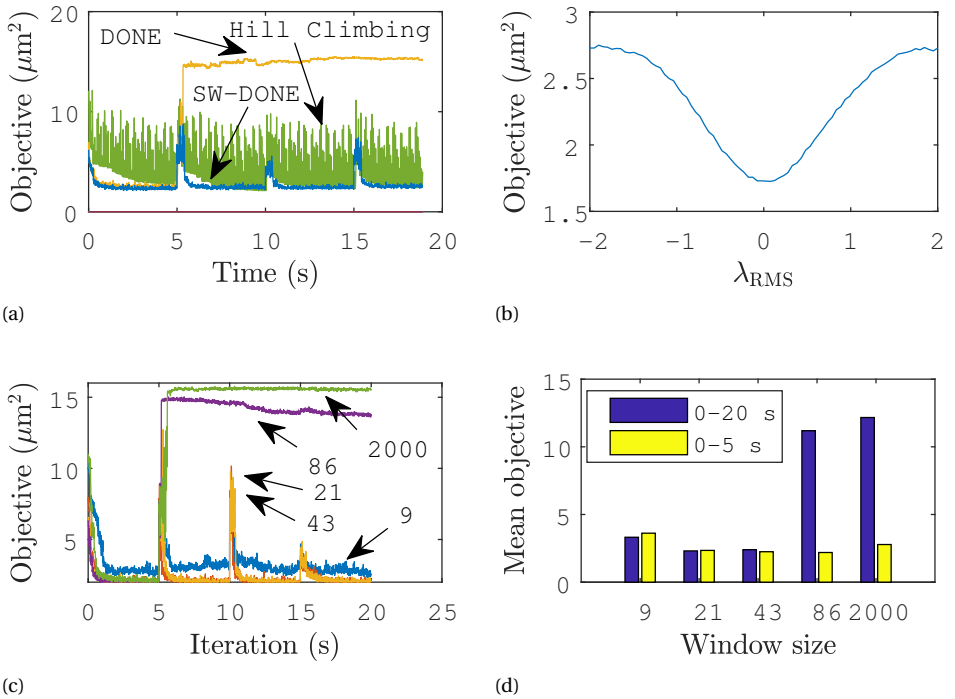


Figure A.3: (a) Comparison between DONE (adapted with a variable offset), SW-DONE and hill climbing. (b) Scan through one of the degrees of freedom (coma). (c) Objective over time for SW-DONE with window sizes varying between 9 and 2000. (d) Mean objective for different time ranges and window sizes.

Figure A.3a shows the results of various algorithms on this application. As a baseline, a hill climbing algorithm was tested using a similar implementation as in [11], with 7 measurements per input dimension. This method was compared with the DONE algorithm [3] adapted with the variable offset as described in this paper, and with the SW-DONE algorithm with a window size of $L = 43$. The DONE algorithm without the variable offset gave an objective function value of around $16\mu m^2$ for the whole duration of the experiment, hence this result has been omitted from the figure.

It can be seen that the SW-DONE algorithm and the hill climbing algorithm give similar final results. However, for the hill climbing algorithm it is required to take measurements far away from the current optimum, resulting in large fluctuations of the objective. This makes the SW-DONE algorithm more fit for real-time applications as it ensures a constant sharp image. Furthermore, unlike the DONE algorithm, the SW-DONE algorithm is able to handle sudden changes in the objective function.

Figure A.3b shows a scan of the objective value through one of the 18 degrees of freedom, namely coma aberration, measured in the root mean square deviation of phase (denoted as λ_{RMS}). The pseudoconvex shape shows the benefit of using the variable forgetting factor of Section A.3.

Figure A.3c investigates the effect of the window size L for the SW-DONE algorithm. Five different values of L were used, namely 9, 21, 43, 86 and 2000. With 2000 total measurements, this last case corresponds to having no sliding window. For the first 5 seconds, a fixed objective function is being minimized, and all window sizes give satisfying results. However, when the objective function changes due to the artificial aberrations, the algorithm fails to adapt when using the largest window sizes, because it makes use of past measurements that have become obsolete. With the smallest window sizes, quick adaptation is provided, however the accuracy deteriorates for $L = 9$ because too few measurements are used. A window size of 21 or 43 gives the best performance (similar performance for both values) and shows that the sliding window principle allows the minimization of functions that change over time. The average values for the first 5 seconds and for the whole duration are shown in Figure A.3d.

A.5. CONCLUSION

We have applied the sliding window principle to the DONE algorithm. Together with the introduction of a variable offset in the linear regression step of the algorithm, this lead to the SW-DONE algorithm. The SW-DONE algorithm can be used to find the local minimum of a function that is not only expensive to evaluate and perturbed by noise, but that also changes over time. This is shown experimentally by applying the algorithm to a confocal fluorescent microscopy application. The objective function in this application suffers from noise and changes over time, as simulated with artificial aberrations. By introducing a sliding window, and a variable offset to exploit pseudoconvexity, the minimum of this objective function was found.

REFERENCES

[1] W. T. Welford, *Aberrations of optical systems* (CRC Press, 1986).

A

[2] H. R. G. W. Verstraete, S. Wahls, J. Kalkman, and M. Verhaegen, *Model-based sensorless wavefront aberration correction in optical coherence tomography,* Opt. Lett. **40**, 5722 (2015).

[3] L. Bliek, H. R. G. W. Verstraete, M. Verhaegen, and S. Wahls, *Online optimization with costly and noisy measurements using random Fourier expansions,* IEEE Transactions on Neural Networks and Learning Systems **29**, 167 (2018).

[4] A. Rahimi and B. Recht, *Random features for large-scale kernel machines,* in *Adv. Neur. In.* (2007) pp. 1177–1184.

[5] M. J. Powell, *The NEWUOA software for unconstrained optimization without derivatives,* in *Large-scale nonlinear optimization* (Springer, 2006) pp. 255–297.

[6] B. Wittenmark and K. J. Åström, *Practical issues in the implementation of self-tuning control,* Automatica **20**, 595 (1984).

[7] D. JANECKI, *New recursive parameter estimation algorithms with varying but bounded gain matrix,* International Journal of Control **47**, 75 (1988).

[8] L. Cao and H. Schwartz, *A directional forgetting algorithm based on the decomposition of the information matrix,* Automatica **36**, 1725 (2000).

[9] S. Gunnarsson, *Combining tracking and regularization in recursive least squares identification,* in *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, Vol. 3 (IEEE, 1996) pp. 2551–2552.

[10] P. Pozzi, D. Wilding, O. Soloviev, H. Verstraete, L. Bliek, G. Vdovin, and M. Verhaegen, *High speed wavefront sensorless aberration correction in digital micromirror based confocal microscopy,* Optics Express **25**, 949 (2017).

[11] M. Skorsetz, P. Artal, and J. M. Bueno, *Performance evaluation of a sensorless adaptive optics multiphoton microscope,* Journal of microscopy **261**, 249 (2016).

[12] A. H. Sayed and T. Kailath, *Recursive least-squares adaptive filters,* Digit. Signal Process. Handbook , 21 (1998).

[13] J. Nocedal and S. Wright, *Numerical optimization* (Springer Science & Business Media, 2006).

[14] A. H. Sayed, *Fundamentals of adaptive filtering* (John Wiley & Sons, 2003).

[15] A. N. Tikhonov and V. Y. Arsenin, *Solutions of ill-posed problems* (Winston, 1977).

[16] M. Born and E. Wolf, *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light* (Elsevier, 1980).

SUMMARY

Beamforming is a signal processing technique used in highly directional antennas. An array of antenna elements transmits the same signal, but with a different time delay for each element. By providing the right time delays for each antenna element, the whole array transmits a high-powered signal in one desired direction. This technique can be used for example to provide satellite television and Internet connections on board of aircrafts.

Recently, developments in the field of integrated microwave photonics have paved the way for broadband, low-loss, and low-weight beamformer systems. These photonic beamformers convert the signals to be transmitted to the optical domain, provide the correct time delays with tunable optical delay lines, and then convert the signal back to the radio frequency domain. The main challenge here lies in tuning the actuators of the tunable optical delay lines in such a way that they provide the desired time delays. Challenges like actuator crosstalk, parameter sensitivity, noise and model errors cause complications when traditional tuning algorithms are used, such as nonlinear optimization routines. All results obtained with these photonic beamformers in the literature so far have been achieved by tuning the whole system by hand, or by applying nonlinear optimization techniques to a simplified simulation of the system rather than the actual system.

In order to find a practical way of tuning a photonic beamformer in real time, this thesis takes a data-driven approach. Instead of relying on perfectly accurate physical models, a surrogate function is used that approximates the relation between the system actuators and a cost function, namely the difference between the measured and desired time delay of each antenna element. By performing nonlinear optimization techniques on this surrogate cost function and by continuously updating the approximation as new measurements are obtained, the time delays of each antenna element should converge towards the desired values.

The Data-based Online Nonlinear Extremum-seeker (DONE) algorithm is used to update and optimize the surrogate function in real time. This algorithm is especially designed to optimize cost functions that are costly to evaluate (for example in terms of time), that contain noise, and for which derivatives cannot be easily computed or approximated. The DONE algorithm is applied to a simulation of a photonic beamformer and to the real system, as well as to several other applications. It is shown that the algorithm outperforms comparable methods on several fronts, especially computation time. Furthermore, the theory behind the algorithm is investigated, but practical results are also given, for example rules of thumb for choosing the hyper-parameters. Finally, variations to the DONE algorithm have been developed that are easier to use, can be implemented more efficiently, and can deal with time-varying objective functions.

SAMENVATTING

Bundelvorming is een signaalverwerkingstechniek die gebruikt wordt bij versterkte richtantennes. Een reeks antenne-elementen verzendt hetzelfde signaal met verschillende vertragingen voor elk element. Door elk antenne-element van de juiste vertragingen te voorzien wordt door de antennereeks een versterkt signaal in één specifieke richting verzonden. Deze techniek kan bijvoorbeeld gebruikt worden om vliegtuigen van satellietteleviesie en -internet te voorzien.

De laatste tijd hebben ontwikkelingen op het gebied van geïntegreerde microgolffotonica de weg gebaand voor breedband, verliesarme en lichtgewicht bundelvormersystemen. Deze fotonische bundelvormers zetten de te verzenden signalen om naar het optische domein, leveren de juiste vertragingen met instelbare optische vertragingslijnen, en zetten het signaal dan weer terug om naar het radiofrequentiedomein. De grootste uitdaging zit hem in het instellen van de actuatoren van de instelbare optische vertragingslijnen op zo een manier dat ze de gewenste vertragingen leveren. Uitdagingen zoals overspraak tussen de actuatoren, parametergevoeligheid, ruis en modelfouten zorgen voor complicaties wanneer traditionele instellingsalgoritmes zoals niet-lineaire-optimalisatieroutines worden gebruikt. Alle resultaten die met deze fotonische bundelvormers zijn behaald in de literatuur tot nu toe, zijn behaald door het hele systeem met de hand in te stellen, of door niet-lineaire-optimalisatietechnieken toe te passen op een vereenvoudigde simulatie van het systeem in plaats van het echte systeem.

Om een praktische manier te vinden om een fotonische bundelvormer in real time in te stellen wordt er in deze dissertatie een datagestuurde aanpak gebruikt. In plaats van te vertrouwen op perfect nauwkeurige fysische modellen wordt er een surrogaatfunctie gebruikt die de relatie tussen systeemactuatoren en een kostenfunctie, namelijk het verschil tussen de gemeten en gewenste vertraging van elk antenne-element, benadert. Door niet-lineaire-optimalisatietechnieken toe te passen op deze surrogaatfunctie en door de benadering herhaaldelijk bij te werken naarmate nieuwe metingen worden verkregen, zouden de vertragingen van elk antenne-element moeten convergeren naar de gewenste waarden.

Het datagestuurde online niet-lineaire extremumzoekeralgoritme (Data-based Online Nonlinear Extremum-seeker, DONE) wordt gebruikt om de surrogaatfunctie in realtime bij te werken en te optimaliseren. Dit algoritme is speciaal ontworpen om kostenfuncties te optimaliseren die prijzig zijn om te evalueren (bijvoorbeeld wat tijd betreft), die ruis bevatten, en waarvoor de afgeleides niet makkelijk uitgerekend of benaderd kunnen worden. Het DONE-algoritme wordt toegepast op een simulatie van een fotonische bundelvormer en op het echte systeem, alsmede op verscheidene andere toepassingen. Er wordt aangetoond dat het algoritme op verschillende vlakken beter presteert dan vergelijkbare algoritmes, vooral wat de rekentijd betreft. Daarnaast wordt de achterliggende theorie van het algoritme onderzocht, maar worden er ook praktische resultaten geleverd, zoals vuistregels voor het kiezen van de hyperparameters. Tenslotte zijn er varian-

ten op het DONE-algoritme ontwikkeld die makkelijker te gebruiken zijn, die efficiënter geïmplementeerd kunnen worden, en die met tijdsvariërende doelfuncties om kunnen gaan.

LIST OF PUBLICATIONS

JOURNAL PAPERS

[1] L. Bliek, H. R. G. W. Verstraete, M. Verhaegen, and S. Wahls, *Online optimization with costly and noisy measurements using random Fourier expansions,* IEEE Transactions on Neural Networks and Learning Systems **29**, 167 (2018).

[2] P. Pozzi, D. Wilding, O. Soloviev, H. Verstraete, L. Bliek, G. Vdovin, and M. Verhaegen, *High speed wavefront sensorless aberration correction in digital micromirror based confocal microscopy,* Optics Express **25**, 949 (2017).

[3] H. R. Verstraete, M. Heisler, M. J. Ju, D. Wahl, L. Bliek, J. Kalkman, S. Bonora, Y. Jian, M. Verhaegen, and M. V. Sarunic, *Wavefront sensorless adaptive optics OCT with the DONE algorithm for in vivo human retinal imaging,* Biomedical optics express **8**, 2261 (2017).

[4] L. Bliek, S. Wahls, I. Visscher, C. Taddei, R. B. Timens, R. Oldenbeuving, C. Roeloffzen, and M. Verhaegen, *Automatic Tuning of a Novel Ring Resonator-based Photonic Beamformer for a Transmit Phased Array Antenna,* arXiv e-prints , arXiv:1808.04814 (2018), arXiv:1808.04814 [physics.app-ph] .

In [1], the first two authors contributed equally.

CONFERENCE PAPERS

[1] L. Bliek, M. Verhaegen, and S. Wahls, *Data-driven minimization with random feature expansions for optical beam forming network tuning,* 16th IFAC Workshop on Control Applications of Optimization (CAO'2015) **48**, 166 (2015).

[2] Y. Jian, H. R. Verstraete, M. Heisler, M. J. Ju, D. J. Wahl, L. Bliek, J. Kalkman, S. Bonora, M. Verhaegen, and M. V. Sarunic, *Data-based online nonlinear extremum-seeker for wavefront sensorless adaptive optics oct (conference presentation),* in *Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XXI*, Vol. 10053 (International Society for Optics and Photonics, 2017) p. 1005327.

[3] H. R. Verstraete, M. Heisler, M. J. Ju, D. J. Wahl, L. Bliek, J. Kalkman, S. Bonora, M. V. Sarunic, M. Verhaegen, and Y. Jian, *Real time optimization algorithm for wavefront sensorless adaptive optics oct (conference presentation),* in *Adaptive Optics and Wavefront Control for Biological Systems III*, Vol. 10073 (International Society for Optics and Photonics, 2017) p. 100731B.

[4] L. Bliek, H. Verstraete, S. Wahls, R. B. Timens, R. Oldenbeuving, C. Roeloffzen, and M. Verhaegen, *Automatic tuning of a ring resonator-based optical delay line for optical beamforming,* in *Symposium on Information Theory and Signal Processing in the Benelux, 2017* (IEEE, 2017) pp. 23–24.

[5] L. Bliek, M. Verhaegen, and S. Wahls, *Online function minimization with convex random relu expansions,* in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)* (2017) pp. 1–6.

[6] C. Roeloffzen, I. Visscher, C. Taddei, D. Geskus, R. Oldenbeuving, J. Epping, R. B. Timens, P. van Dijk, R. Heideman, M. Hoekman, R. Grootjans, L. Bliek, S. Wahls, and M. Verhaegen, *Integrated microwave photonics for 5G,* in *Conference on Lasers and Electro-Optics* (Optical Society of America, 2018) p. JTh3D.2.

CURRICULUM VITAE

Laurens Bliek was born in Amsterdam, the Netherlands, on the 4th of December 1989. After graduating from VWO (secondary education) at the Zeldenrust-Steelantcollege in Terneuzen, the Netherlands, in 2007, he went on to study Applied Mathematics at the Delft University of Technology. He obtained his Bachelor of Science degree in 2011 on the topic "Memristors: one step closer towards electrical brains", and his Master of Science degree - also in Applied Mathematics at the Delft University of Technology - in 2013, on the topic "Nonlinear System Identification and Control for Autonomous Robots". In 2013 he was also a student assistant at the Delft Institute of Applied Mathematics, as part of the Umbrella project, a European project for the optimization of electrical power systems.

In 2014 he started his PhD at the Delft Center for Systems and Control on the SCOPAS project (Smart Control of OBFN-based Phased Array Systems), under the supervision of prof. dr. ir. Michel Verhaegen and dr. ir. Sander Wahls. The goal of the project was to provide smart control algorithms for an optical beamforming network (OBFN). The results of the project are presented in this thesis. During his PhD, he supervised a total of six master students and assisted in the B.Sc. course "Stochastische Signaalanalyse". He also performed measurements for the SCOPAS project at LioniX International, and participated in various conferences. He continued as a post-doctoral researcher on the project "Real time data-driven maintenance logistics" at the Algorithmics group and at the Cybersecurity group of Delft University of Technology.