TU Delft

Delft University of Technology

Online optimization with costly and noisy measurements using random Fourier expansions

Bliek, Laurens; Verstraete, Hans; Verhaegen, Michel; Wahls, Sander

DOI
10.1109/TNNLS.2016.2615134

Publication date
2016

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Neural Networks and Learning Systems

Citation (APA)
Bliek, L., Verstraete, H., Verhaegen, M., & Wahls, S. (2016). Online optimization with costly and noisy measurements using random Fourier expansions. *IEEE Transactions on Neural Networks and Learning Systems*, *29 (2018)*(1), 167-182. https://doi.org/10.1109/TNNLS.2016.2615134

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Aceepted Author Manuscript

Online Optimization with Costly and Noisy Measurements using Random Fourier Expansions

Laurens Bliek*, Hans R. G. W. Verstraete*, Michel Verhaegen, *Member, IEEE* and Sander Wahls, *Member, IEEE*

***Abstract*—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.**

***Index Terms*—derivative-free optimization, Bayesian optimization, surrogate model, learning systems, adaptive optics**

I. 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 [1], [2]. Some examples of commonly used DFO algorithms are the simplex method [3], NEWUOA [4], BOBYQA [5], and DIRECT [6]. 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 [7] and SID-PSM [8] 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 [9]. 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 [10]–[13]. In these processes different kernels and kernel learning methods are used for the covariance function [14], [15]. 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 [16], robotics [17], hyper-parameter tuning [11], and optics [18].

Recently, the Data-based Online Nonlinear Extremum-seeker (DONE) algorithm was proposed in [19]. It is similar to Bayesian optimization, but simpler and faster. The DONE algorithm uses random Fourier expansions [20] (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 [13], a Bayesian optimization library that was shown to outperform many other similar libraries in [13]. The first application is optical coherence tomography (OCT), a 3D imaging method based on interference often used to image the human retina [19], [21], [22]. 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 [23]–[28]. The third application is a robot arm of which the tip has to be directed to a desired position [29].

This paper is organized as follows. Section II 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 III explains the DONE algorithm. Theoretically optimal as well as more practical ways to choose the hyper-parameters of this algorithm are given in Section IV. In Section V the DONE algorithm and

*Both authors contributed equally to this work. Corresponding authors: l.bliek@tudelft.nl, h.r.g.w.verstraete@tudelft.nl.

All authors are with the Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, Netherlands.

BayesOpt are compared for a benchmark problem and for the three aforementioned applications. We conclude the paper in Section VI.

II. 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 [30]–[32]. In this paper, we use random Fourier expansions (RFEs) [20] 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 [20], [33], [34].

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

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

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 (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 [20]. Assuming the kernel k is shift-invariant and has Fourier transform p, it can be normalized such that p is a probability distribution [20]. That is, we have

$$k(\mathbf{x}_i - \mathbf{x}_j) = \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) e^{-i\boldsymbol{\omega}^T(\mathbf{x}_i - \mathbf{x}_j)} d\boldsymbol{\omega}. \tag{2}$$

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

$$\begin{aligned}
k(\mathbf{x}_i - \mathbf{x}_j) &= \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) \cos(\boldsymbol{\omega}^T(\mathbf{x}_i - \mathbf{x}_j)) d\boldsymbol{\omega} \\
&= \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) \cos(\boldsymbol{\omega}^T(\mathbf{x}_i - \mathbf{x}_j)) \\
&\quad + p(\boldsymbol{\omega}) \int_0^{2\pi} \cos(\boldsymbol{\omega}^T(\mathbf{x}_i + \mathbf{x}_j) + 2b) db d\boldsymbol{\omega} \\
&= \frac{1}{2\pi} \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) \int_0^{2\pi} \cos(\boldsymbol{\omega}^T(\mathbf{x}_i - \mathbf{x}_j)) \\
&\quad + \cos(\boldsymbol{\omega}^T(\mathbf{x}_i + \mathbf{x}_j) + 2b) db d\boldsymbol{\omega} \\
&= \frac{1}{2\pi} \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) \int_0^{2\pi} 2\cos(\boldsymbol{\omega}^T\mathbf{x}_i + b) \\
&\quad \cdot \cos(\boldsymbol{\omega}^T\mathbf{x}_j + b) db d\boldsymbol{\omega} \\
&= \mathbb{E}[2\cos(\boldsymbol{\Omega}^T\mathbf{x}_i + B)\cos(\boldsymbol{\Omega}^T\mathbf{x}_j + B)] \\
&\approx \frac{2}{D} \sum_{k=1}^{D} \cos(\boldsymbol{\omega}_k^T\mathbf{x}_i + b_k)\cos(\boldsymbol{\omega}_k^T\mathbf{x}_j + b_k), \quad (3)
\end{aligned}$$

if $\boldsymbol{\omega}_k$ 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 B with a uniform distribution. For $c_k = \sum_{i=1}^{N} \frac{2}{D} a_i \cos(\boldsymbol{\omega}_k^T\mathbf{x}_i + b_k)$ we thus have:

$$g_{KM}(\mathbf{x}) \approx \sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k). \tag{4}$$

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 $g : \mathbb{R}^d \to \mathbb{R}$,*

$$g(\mathbf{x}) = \sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k), \tag{5}$$

with $D \in \mathbb{N}$, the b_k being realizations of independent and identically distributed (i.i.d.) uniformly distributed random variables B_k on $[0, 2\pi]$, and with the $\boldsymbol{\omega}_k \in \mathbb{R}^d$ being realizations of i.i.d. random vectors $\boldsymbol{\Omega}_k$ with an arbitrary continuous p.d.f. $p_{\boldsymbol{\Omega}}$. The 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 [35]–[38]. However, these are much more difficult to tune than using RFEs [34]. 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 [39]. Moreover, the parameters of the cosines have intuitive interpretations in terms of the Fourier transform.

A. 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 [40], [41]), which corresponds to turning the corresponding sums into integrals.

Theorem 1. *Let $f \in L^2(\mathbb{R}^d)$ be a real-valued function and let*

$$\bar{c}(\boldsymbol{\omega}, b) = \begin{cases} \frac{1}{\pi}|\hat{f}(\boldsymbol{\omega})|\cos(\angle\hat{f}(\boldsymbol{\omega}) - b), & b \in [0, 2\pi], \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

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

$$f(\mathbf{x}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \int_0^{2\pi} \bar{c}(\boldsymbol{\omega}, b)\cos(\boldsymbol{\omega}^T\mathbf{x} + b) db d\boldsymbol{\omega}. \tag{7}$$

Here, $|\hat{f}|$ and $\angle\hat{f}$ denote the magnitude and phase of the Fourier transform $\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} f(\mathbf{x})e^{-i\boldsymbol{\omega}^T\mathbf{x}}d\mathbf{x}$. 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

$$\begin{aligned}\bar{c}(\boldsymbol{\omega}, b) &= \frac{1}{\pi}|\hat{f}(\boldsymbol{\omega})|\cos(\angle\hat{f}(\boldsymbol{\omega}) - b)\\ &= \frac{1}{\pi}\mathrm{Re}\left\{\hat{f}(\boldsymbol{\omega})e^{-ib}\right\}. \end{aligned} \tag{8}$$

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

$$\begin{aligned}f(\mathbf{x}) =&\mathrm{Re}\left\{\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\hat{f}(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^T\mathbf{x}}d\boldsymbol{\omega}\right\}\\ =&\mathrm{Re}\Bigg\{\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\Big(\hat{f}(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^T\mathbf{x}}\frac{1}{2\pi}\int_0^{2\pi}1db+\\ &\hat{f}(\boldsymbol{\omega})e^{-i\boldsymbol{\omega}^T\mathbf{x}}\underbrace{\int_0^{2\pi}e^{-2ib}db}_{=0}\Big)d\boldsymbol{\omega}\Bigg\}\\ =&\mathrm{Re}\left\{\frac{1}{\pi}\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\hat{f}(\boldsymbol{\omega})e^{-ib}\right.\\ &\left.\frac{1}{2}\left[e^{i(\boldsymbol{\omega}^T\mathbf{x}+b)} + e^{-i(\boldsymbol{\omega}^T\mathbf{x}+b)}\right]dbd\boldsymbol{\omega}\right\}\\ =&\mathrm{Re}\left\{\frac{1}{\pi}\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\hat{f}(\boldsymbol{\omega})e^{-ib}\cos(\boldsymbol{\omega}^T\mathbf{x}+b)dbd\boldsymbol{\omega}\right\}\\ \overset{(8)}{=}&\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\boldsymbol{\omega}, b)\cos(\boldsymbol{\omega}^T\mathbf{x}+b)dbd\boldsymbol{\omega}. \end{aligned} \tag{9}$$

□

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

$$\begin{aligned}\bar{c}(\boldsymbol{\omega}, b) &= \frac{1}{\pi}\mathrm{Re}\left\{\hat{f}(\boldsymbol{\omega})e^{-ib}\right\}\\ &= \frac{1}{\pi}\mathrm{Re}\left\{\int_{\mathbb{R}^d}f(\mathbf{x})e^{-i(\boldsymbol{\omega}^T\mathbf{x}+b)}d\mathbf{x}\right\}\\ &= \frac{1}{\pi}\int_{\mathbb{R}^d}f(\mathbf{x})\cos(\boldsymbol{\omega}^T\mathbf{x}+b)d\mathbf{x}. \end{aligned} \tag{10}$$

The function $\bar{c}$ in Theorem 1 is not unique. However, of all functions c that satisfy (7), the given $\bar{c}$ is the one with minimum norm.

Theorem 2. *Let $\bar{c}$ be as in Theorem 1. If $\tilde{c} : \mathbb{R}^d \times [0, 2\pi] \to \mathbb{R}$ satisfies*

$$f(\mathbf{x}) = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\tilde{c}(\boldsymbol{\omega}, b)\cos(\boldsymbol{\omega}^T\mathbf{x}+b)dbd\boldsymbol{\omega} \quad \text{a.e.} \tag{11}$$

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

$$\begin{aligned}||\bar{c}||_{L^2}^2 &= \int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\boldsymbol{\omega}, b)^2dbd\boldsymbol{\omega}\\ &\overset{(6)}{=}\int_{\mathbb{R}^d}\int_0^{2\pi}\frac{1}{\pi^2}|\hat{f}(\boldsymbol{\omega})|^2\cos(\angle\hat{f}(\boldsymbol{\omega}) - b)^2dbd\boldsymbol{\omega}\\ &= \int_{\mathbb{R}^d}\frac{1}{\pi^2}|\hat{f}(\boldsymbol{\omega})|^2\int_0^{2\pi}\cos(\angle\hat{f}(\boldsymbol{\omega}) - b)^2dbd\boldsymbol{\omega}\\ &= \int_{\mathbb{R}^d}\frac{1}{\pi}|\hat{f}(\boldsymbol{\omega})|^2d\boldsymbol{\omega}\\ &= \frac{(2\pi)^d}{\pi}\int_{\mathbb{R}^d}f(\mathbf{x})^2d\mathbf{x} = \frac{(2\pi)^d}{\pi}||f||_{L^2}^2. \end{aligned} \tag{12}$$

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

$$\begin{aligned}&\int_{\mathbb{R}^d}f(\mathbf{x})^2d\mathbf{x}\\ &\overset{(11)}{=}\int_{\mathbb{R}^d}f(\mathbf{x})\frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\tilde{c}(\boldsymbol{\omega}, b)\cos(\boldsymbol{\omega}^T\mathbf{x}+b)dbd\boldsymbol{\omega}d\mathbf{x}\\ &= \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\tilde{c}(\boldsymbol{\omega}, b)\int_{\mathbb{R}^d}f(\mathbf{x})\cos(\boldsymbol{\omega}^T\mathbf{x}+b)d\mathbf{x}dbd\boldsymbol{\omega}\\ &\overset{(10)}{=}\frac{\pi}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\tilde{c}(\boldsymbol{\omega}, b)\bar{c}(\boldsymbol{\omega}, b)dbd\boldsymbol{\omega}\\ &= \frac{\pi}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\boldsymbol{\omega}, b)^2 + \bar{c}(\boldsymbol{\omega}, b)q(\boldsymbol{\omega}, b)dbd\boldsymbol{\omega}\\ &\overset{(12)}{=}\int_{\mathbb{R}^d}f(\mathbf{x})^2d\mathbf{x} + \frac{\pi}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\boldsymbol{\omega}, b)q(\boldsymbol{\omega}, b)dbd\boldsymbol{\omega}. \end{aligned} \tag{13}$$

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

$$\begin{aligned}||\tilde{c}||_{L^2}^2 &= ||\bar{c} + q||_{L^2}^2\\ &= \int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\boldsymbol{\omega}, b)^2 + 2\bar{c}(\boldsymbol{\omega}, b)q(\boldsymbol{\omega}, b) + q(\boldsymbol{\omega}, b)^2dbd\boldsymbol{\omega}\\ &= ||\bar{c}||_{L^2}^2 + ||q||_{L^2}^2 \geq ||\bar{c}||_{L^2}^2. \end{aligned} \tag{14}$$

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 (6). We prove unbiasedness in the following theorem, while variance properties are analyzed in Appendix B.

Theorem 3. *For any continuous p.d.f. $p_{\boldsymbol{\Omega}}$ with $p_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) > 0$ if $|\hat{f}(\boldsymbol{\omega})| > 0$, the choice*

$$C_k = \frac{2}{D(2\pi)^d}\frac{|\hat{f}(\boldsymbol{\Omega}_k)|}{p_{\boldsymbol{\Omega}}(\boldsymbol{\Omega}_k)}\cos(\angle\hat{f}(\boldsymbol{\Omega}_k) - B_k) \tag{15}$$

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

Proof. Using Theorem 1, we have

$$\begin{aligned}f(\mathbf{x}) &= \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\int_0^{2\pi}\bar{c}(\boldsymbol{\omega}, b)\cos(\boldsymbol{\omega}^T\mathbf{x}+b)dbd\boldsymbol{\omega}\\ &= \mathbb{E}_{\boldsymbol{\Omega}_1, B_1}\left[\frac{1}{(2\pi)^d p_B(B_1)p_{\boldsymbol{\Omega}}(\boldsymbol{\Omega}_1)}\bar{c}(\boldsymbol{\Omega}_1, B_1)\cos(\boldsymbol{\Omega}_1^T\mathbf{x}+B_1)\right]\\ &= \mathbb{E}_{\boldsymbol{\Omega}_{1\dots D}, B_{1\dots D}}\left[\sum_{k=1}^D\frac{2\pi\bar{c}(\boldsymbol{\Omega}_k, B_k)}{D(2\pi)^d p_{\boldsymbol{\Omega}}(\boldsymbol{\Omega}_k)}\cos(\boldsymbol{\Omega}_k^T\mathbf{x}+B_k)\right] \end{aligned}$$

$$\overset{(6)}{=} \mathbb{E}\left[\sum_{k=1}^{D} \frac{2}{D(2\pi)^d} \frac{|\hat{f}(\boldsymbol{\Omega}_k)|}{p_{\boldsymbol{\Omega}}(\boldsymbol{\Omega}_k)} \cos(\angle \hat{f}(\boldsymbol{\Omega}_k) - B_k) \cos(\boldsymbol{\Omega}_k^T \mathbf{x} + B_k)\right] = \mathbb{E}\left[G(\mathbf{x})\right]. \tag{16}$$

□

These ideal weights enjoy many other nice properties such as infinity norm convergence [42]. In practice, however, a least squares approach is used for a finite D. This is investigated in the next subsection.

B. 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 $\mathbf{x}_1, \ldots, \mathbf{x}_N$ 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 $y_1, \ldots, y_N$, with $y_n = f(\mathbf{x}_n) + \eta_n$, where $\eta_1, \ldots, \eta_N$ 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

$$\begin{aligned} J_N(\mathbf{c}) &= \sum_{n=1}^{N} \left(y_n - \sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T \mathbf{x}_n + b_k) \right)^2 + \lambda \sum_{k=1}^{D} c_k^2 \\ &= ||\mathbf{y}_N - \mathbf{A}_N \mathbf{c}||_2^2 + \lambda ||\mathbf{c}||_2^2. \end{aligned} \tag{17}$$

Here,

$$\mathbf{y}_N = \begin{bmatrix} y_1 \cdots y_N \end{bmatrix}^T, \quad \mathbf{A}_N = \begin{bmatrix} \cos(\boldsymbol{\omega}_1^T \mathbf{x}_1 + b_1) & \cdots & \cos(\boldsymbol{\omega}_D^T \mathbf{x}_1 + b_D) \\ \vdots & \ddots & \vdots \\ \cos(\boldsymbol{\omega}_1^T \mathbf{x}_N + b_1) & \cdots & \cos(\boldsymbol{\omega}_D^T \mathbf{x}_N + b_D) \end{bmatrix}, \tag{18}$$

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

Since the parameters $\boldsymbol{\omega}_k, b_k$ 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

$$\mathbf{c_N} = \left(\mathbf{A}_N^T \mathbf{A}_N + \lambda \mathbf{I}_{D\times D}\right)^{-1} \mathbf{A}_N^T \mathbf{y}_N. \tag{19}$$

The following theorem shows that RFEs whose coefficient vector have been obtained through a least squares fit as in (19) can approximate the function f arbitrarily well. Similar results were given in [40]–[43], but we emphasize that these convergence results did concern RFEs employing the ideal coefficient vector given earlier in Theorem 3 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 4. *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 $\sup_{\boldsymbol{\omega}\in\mathbb{R}^D, b\in[0,2\pi]} \left| \frac{\bar{c}(\boldsymbol{\omega}, b)}{p_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) p_B(b)} \right| < \infty$. Then, for every $\epsilon > 0$ and $\delta > 0$, there exist constants N_0 and D_0 such that*

$$\int_{\mathcal{X}} \left(f(\mathbf{x}) - \sum_{k=1}^{D} C_{Nk} \cos(\boldsymbol{\Omega}_k^T \mathbf{x} + B_k) \right)^2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} < \epsilon \tag{20}$$

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 (19)*, and $\Lambda \geq 0$ is the solution to*

$$\left|\left| \left(\mathbf{A}_N^T \mathbf{A}_N + N\Lambda\, \mathbf{I}_{D\times D}\right)^{-1} \mathbf{A}_N^T \mathbf{y}_N \right|\right|_2^2 = \sum_{k=1}^{D} \left(\frac{\bar{c}(\boldsymbol{\omega}_k, b_k)}{(2\pi)^d D p_{\boldsymbol{\Omega}}(\boldsymbol{\omega}_k) p_B(b_k)} \right)^2. \tag{21}$$

The proof of this theorem is given in Appendix A. In Section IV-B we show how to obtain Λ in practice.

III. 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.

A. 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 $\mathbf{a}_n = [\cos(\boldsymbol{\omega}_1^T \mathbf{x}_n + b_1) \cdots \cos(\boldsymbol{\omega}_D^T \mathbf{x}_n + b_D)]$, then we aim to find the vector of RFE weights by minimizing the regularized mean square error

$$J_n(\mathbf{c}) = \sum_{i=1}^{n} \left(y_i - \mathbf{a}_i \mathbf{c}\right)^2 + \lambda ||\mathbf{c}||_2^2. \tag{22}$$

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

$$\mathbf{c}_n = \operatorname*{argmin}_{\mathbf{c}} J_n(\mathbf{c}). \tag{23}$$

Assuming we have found $\mathbf{c}_n$, we would like to use this information to find $\mathbf{c}_{n+1}$ without solving (23) again. The recursive least squares algorithm is a computationally efficient method that determines $\mathbf{c}_{n+1}$ from $\mathbf{c}_n$ as follows [44, Sec. 21]:

$$\gamma_n = 1/(1 + \mathbf{a}_n \mathbf{P}_{n-1} \mathbf{a}_n^T), \tag{24}$$

$$\mathbf{g}_n = \gamma_n \mathbf{P}_{n-1}\mathbf{a}_n^T, \tag{25}$$

$$\mathbf{c}_n = \mathbf{c}_{n-1} + \mathbf{g}_n(y_n - \mathbf{a}_n\mathbf{c}_{n-1}), \tag{26}$$

$$\mathbf{P}_n = \mathbf{P}_{n-1} - \mathbf{g}_n\mathbf{g}_n^T/\gamma_n, \tag{27}$$

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 [44, Sec. 21], which is known to be especially numerically reliable. Instead of performing the update rules (24)-(27) explicitly, we find a rotation matrix $\mathbf{\Theta}_n$ that lower triangularizes the upper triangular matrix in Eq. (28) below and generates a post-array with positive diagonal entries:

$$\begin{bmatrix} 1 & \mathbf{a}_n\mathbf{P}_{n-1}^{1/2} \\ \mathbf{0} & \mathbf{P}_{n-1}^{1/2} \end{bmatrix} \mathbf{\Theta}_n = \begin{bmatrix} \gamma_n^{-1/2} & \mathbf{0} \\ \mathbf{g}_n\gamma_n^{-1/2} & \mathbf{P}_n^{1/2} \end{bmatrix}. \tag{28}$$

The rotation matrix $\mathbf{\Theta}_n$ can be found by performing a QR decomposition of the transpose of the matrix on the left hand side of (28), or by the procedure explained in [44, Sec. 21]. The computational complexity of this update is $O(D^2)$ per iteration.

B. 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 $\mathcal{X} \subseteq \mathbb{R}^d$ by updating a RFE $g(\mathbf{x}) = \sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k)$ 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 $\mathbf{x}_1 \in \mathcal{X}$ and the number of basis functions D have to be chosen. The parameters $\boldsymbol{\omega}_k$ 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. IV. 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 II-B, 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 III-A, we update the RFE model $g(\mathbf{x}) = \sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k)$ based on the new measurement from Step 1 by using the inverse QR algorithm given in (24)-(27). Only the weights c_k are updated. The parameters $\boldsymbol{\omega}_k$ 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 [45], [46]. 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:

$$\mathbf{x}_{init} = P_{\mathcal{X}}(\mathbf{x}_n + \zeta_n), \tag{29}$$

where $P_{\mathcal{X}}$ is the projection onto $\mathcal{X}$. 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 [47]:

$$\mathbf{x}_{n+1} = P_{\mathcal{X}}(\hat{\mathbf{x}}_n + \xi_n). \tag{30}$$

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 $\mathcal{X} = [lb, ub]^d$.

Algorithm 1 DONE Algorithm

procedure DONE($f, \mathbf{x}_1, N, lb, ub, D, \lambda, \sigma_\zeta, \sigma_\xi$)
  Draw $\boldsymbol{\omega}_1 \dots \boldsymbol{\omega}_D$ from $p_{\mathbf{\Omega}}$ independently.
  Draw $b_1 \dots b_D$ from $\mathrm{Uniform}(0, 2\pi)$ 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**
    $\mathbf{a}_n = [\cos(\boldsymbol{\omega}_1^T\mathbf{x}_n + b_1) \cdots \cos(\boldsymbol{\omega}_D^T\mathbf{x}_n + b_D)]$
    $y_n = f(\mathbf{x}_n) + \eta_n$
    $g(\mathbf{x}) = \mathrm{updateRFE}(\mathbf{c}_{n-1}, \mathbf{P}_{n-1}^{1/2}, \mathbf{a}_n, y_n)$
    Draw ζ_n from $\mathcal{N}(0, \sigma_\zeta^2\mathbf{I}_{d\times d})$.
    $\mathbf{x}_{init} = \max(\min(\mathbf{x}_n + \zeta_n, ub), lb)$
    $[\hat{\mathbf{x}}_n, \hat{g}_n] = \text{L-BFGS}(g(\mathbf{x}), \mathbf{x}_{init}, lb, ub)$
    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 (28)
  $\mathbf{c}_n = \mathbf{c}_{n-1} + \mathbf{g}_n(y_n - \mathbf{a}_n\mathbf{c}_{n-1})$
  $g(\mathbf{x}) = [\cos(\boldsymbol{\omega}_1^T\mathbf{x} + b_1)\cdots\cos(\boldsymbol{\omega}_D^T\mathbf{x} + b_D)]\mathbf{c}_n$
  return $g(\mathbf{x})$

IV. 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 performance of DONE depends on the following hyper-parameters:

- number of basis functions D,
- p.d.f. $p_{\boldsymbol{\Omega}}$,
- 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 [15], [16]. 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 $p_{\boldsymbol{\Omega}}$ and λ.

A. Probability Distribution of Frequencies

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

Unfortunately, as shown in Theorem 7 in Appendix B, it turns out that the optimal p.d.f. is

$$p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) = \frac{|\hat{f}(\boldsymbol{\omega})|\sqrt{\cos(2\angle\hat{f}(\boldsymbol{\omega}) + 2\boldsymbol{\omega}^T\mathbf{x}) + 2}}{\int_{\mathbb{R}^d}|\hat{f}(\tilde{\boldsymbol{\omega}})|\sqrt{\cos(2\angle\hat{f}(\tilde{\boldsymbol{\omega}}) + 2\tilde{\boldsymbol{\omega}}^T\mathbf{x}) + 2}d\tilde{\boldsymbol{\omega}}}. \quad (31)$$

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 $\boldsymbol{\omega}_k$ that does not depend on the input $\mathbf{x}$, since the $\boldsymbol{\omega}_k$ 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_B such that $b_k \in [0, 2\pi]$. However, the magnitude $|\hat{f}|$ can be measured in this case. Section V-B describes an example of such a problem. We will now derive a way to choose $p_{\boldsymbol{\Omega}}$ for calibration problems.

In order to get a close to optimal p.d.f. for $\boldsymbol{\omega}_k$ 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 5. *Let $\tilde{G}(\mathbf{x}) = \sum_{k=1}^D \tilde{C}_k e^{i\boldsymbol{\Omega}_k^T\mathbf{x}+B_k}$, with $\boldsymbol{\Omega}_k$ being i.i.d. random vectors with a continuous p.d.f. $\tilde{p}_{\boldsymbol{\Omega}}$ over $\mathbb{R}^d$ that satisfies $\tilde{p}_{\boldsymbol{\Omega}_k}(\boldsymbol{\omega}) > 0$ if $|\hat{f}(\boldsymbol{\omega})| > 0$, and 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*

$$\tilde{C}_k = \frac{\hat{f}(\boldsymbol{\Omega}_k)e^{-iB_k}}{D(2\pi)^d\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\Omega}_k)}. \quad (32)$$

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

$$\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) = \frac{|\hat{f}(\boldsymbol{\omega})|}{\int_{\mathbb{R}^d}|\hat{f}(\tilde{\boldsymbol{\omega}})|d\tilde{\boldsymbol{\omega}}}, \quad (33)$$

giving a variance of

$$\mathrm{Var}[\tilde{G}(\mathbf{x})] = \frac{1}{D(2\pi)^{2d}}\left(\int_{\mathbb{R}^d}|\hat{f}(\boldsymbol{\omega})|d\boldsymbol{\omega}\right)^2 - f(\mathbf{x})^2. \quad (34)$$

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

$$\begin{aligned}
\mathbb{E}\left[\tilde{G}(\mathbf{x})\right] &= \sum_{k=1}^D\int_{\mathbb{R}^d}\int_0^{2\pi}\frac{\hat{f}(\boldsymbol{\omega}_k)e^{-ib_k}e^{i\boldsymbol{\omega}_k^T\mathbf{x}+b_k}}{D(2\pi)^d\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega}_k)2\pi}db_k\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega}_k)d\boldsymbol{\omega}_k \\
&= D\int_{\mathbb{R}^d}\int_0^{2\pi}\frac{\hat{f}(\boldsymbol{\omega})e^{-ib}}{D(2\pi)^d\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}e^{i\boldsymbol{\omega}^T\mathbf{x}+b}\frac{1}{2\pi}db\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega})d\boldsymbol{\omega} \\
&= D\int_{\mathbb{R}^d}\frac{\hat{f}(\boldsymbol{\omega})}{D(2\pi)^d\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}e^{i\boldsymbol{\omega}^T\mathbf{x}}\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega})\int_0^{2\pi}\frac{1}{2\pi}dbd\boldsymbol{\omega} \\
&= \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d}\hat{f}(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^T\mathbf{x}}d\boldsymbol{\omega} \\
&= f(\mathbf{x}). \quad (35)
\end{aligned}$$

The proof of minimum variance is similar to the proof of [48, 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, $\tilde{p}_{\boldsymbol{\Omega}}$, is still close-to-optimal (in terms of the second moment) when used in the real RFE from Definition 1.

Theorem 6. *Let $\tilde{p}_{\mathbf{\Omega}}$ be as in* (33) *and let G with weights C_k be as in Theorem 3. Let P be the set of probability distribution functions for $\mathbf{\Omega}_k$ that are positive when $|\hat{f}(\boldsymbol{\omega})| > 0$. Then, we have*

$$\mathbb{E}_{\tilde{p}_{\mathbf{\Omega}},p_B}[G(\mathbf{x})^2] \leq \sqrt{3} \min_{p_{\mathbf{\Omega}} \in P} \mathbb{E}_{p_{\mathbf{\Omega}},p_B}[G(\mathbf{x})^2]. \tag{36}$$

The proof is given in Appendix B. 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 $p_{\boldsymbol{\omega}}$

1: **if** $|\hat{f}|$ is known exactly **then**
2: Set $p_{\mathbf{\Omega}} = |\hat{f}|/\int |\hat{f}(\boldsymbol{\omega})| d\boldsymbol{\omega}$.
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 $|\hat{f}|/\int |\hat{f}(\boldsymbol{\omega})| d\boldsymbol{\omega}$.
6: Set $p_{\mathbf{\Omega}} = \mathcal{N}(0, \sigma^2 \mathbf{I}_{d\times d})$.

B. Upper Bound on the Regularization Parameter

The regularization parameter λ in the performance criterion (17) is used to prevent under- or over-fitting of the RFE under noisy conditions or when dealing with few measurements. Theorem 4 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 (21). Here we will provide a method to estimate Λ.

During the proof of Theorem 4, it was shown that the upper bound Λ corresponds to the λ that satisfies

$$\begin{aligned} &\left|\left|\left(\mathbf{A}_N^T \mathbf{A}_N + N\lambda\ \mathbf{I}_{D\times D}\right)^{-1} \mathbf{A}_N^T \mathbf{y}_N\right|\right|_2^2 \\ &= \sum_{k=1}^{D} \left(\frac{\bar{c}(\boldsymbol{\omega}_k, b_k)}{(2\pi)^d D p_{\mathbf{\Omega}}(\boldsymbol{\omega}_k) p_B(b_k)}\right)^2 = M^2. \end{aligned} \tag{37}$$

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

$$\begin{aligned} M &= \frac{2}{(2\pi)^d} \sqrt{\frac{1}{D^2} \sum_{k=1}^{D} \left(\frac{|\hat{f}(\boldsymbol{\omega}_k)|}{p_{\mathbf{\Omega}}(\boldsymbol{\omega}_k)} \cos(\angle\hat{f}(\boldsymbol{\omega}_k) - b_k)\right)^2} \\ &\approx \frac{2}{(2\pi)^d} \sqrt{\frac{1}{D} \mathbb{E}\left[\left(\frac{|\hat{f}(\mathbf{\Omega}_1)|}{p_{\mathbf{\Omega}}(\mathbf{\Omega}_1)} \cos(\angle\hat{f}(\mathbf{\Omega}_1) - B_1)\right)^2\right]} \\ &= \frac{2}{(2\pi)^d} \sqrt{\frac{1}{2\pi D} \int_{\mathbb{R}^d} \int_0^{2\pi} \frac{|\hat{f}(\boldsymbol{\omega})|^2}{p_{\mathbf{\Omega}}(\boldsymbol{\omega})} \cos^2(\angle\hat{f}(\boldsymbol{\omega}) - b) db d\boldsymbol{\omega}} \\ &= \frac{\sqrt{2}}{(2\pi)^d \sqrt{D}} \sqrt{\int_{\mathbb{R}^d} \frac{|\hat{f}(\boldsymbol{\omega})|^2}{p_{\mathbf{\Omega}}(\boldsymbol{\omega})} d\boldsymbol{\omega}} \\ &\approx \frac{\sqrt{2}}{(2\pi)^d \sqrt{D}} \sqrt{\int_{\mathbb{R}^d} \frac{|\hat{f}(\boldsymbol{\omega})|^2}{\tilde{p}_{\mathbf{\Omega}}(\boldsymbol{\omega})} d\boldsymbol{\omega}} \\ &= \frac{\sqrt{2}}{(2\pi)^d \sqrt{D}} \int |\hat{f}(\boldsymbol{\omega})| d\boldsymbol{\omega} = M_a. \end{aligned} \tag{38}$$

The squared cosine was removed as in Eq. (12). Using the exact value or an estimate of $\int_{\mathbb{R}^d} |\hat{f}(\boldsymbol{\omega})| d\boldsymbol{\omega}$ as in Algorithm 3 to determine M_a, we calculate the left-hand in (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 $\int_{\mathbb{R}^d} |\hat{f}(\boldsymbol{\omega})| d\boldsymbol{\omega}$.
2: Take N measurements to get $\mathbf{A}_N$ and $\mathbf{y}_N$.
3: Determine Λ for which the left-hand side of (37) is close to $M_a^2 = \frac{2}{(2\pi)^{2d} D} \left(\int |\hat{f}(\boldsymbol{\omega})| d\boldsymbol{\omega}\right)^2$.

V. Numerical Examples

In this section, we compare the DONE algorithm to the Bayesian optimization library BayesOpt [13] in several numerical examples.

A. Analytic Benchmark Problem: Camelback Function

The camelback function

$$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1 x_2 + \left(-4 + 4x_2^2\right) x_2^2, \tag{39}$$

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 $h(\boldsymbol{\omega}) = \frac{C}{\sigma\sqrt{2\pi}} e^{-\frac{\boldsymbol{\omega}^2}{2\sigma^2}}$ 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 $\boldsymbol{\omega}_k$, using the least squares approach from Section II-B. 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)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(f(\mathbf{x}_n) - g(\mathbf{x}_n))^2}, \tag{40}$$

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 5, for $\sigma = 10$, and for different values of D ($D \in \{10, 20, 40, 80, 160, 320, 640, 1280\}$). Fig. 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 8 in Appendix B.

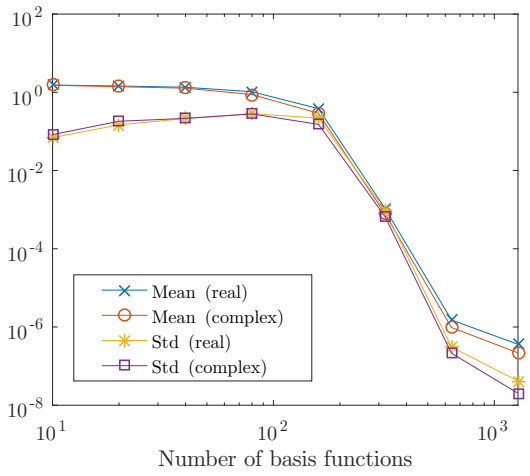

Fig. 1. Mean and standard deviation of the root mean square error for a real and a complex RFE over 100 runs.

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 [13, 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. 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 I. As in [13], 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.

TABLE I
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)

B. 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 [19], [22]. 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 remove as much of the aberrations as possible. In [19] 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 [13]. The numerical results are obtained by simulating the OCT transfer function as described in [49], [50] 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. 2. For the DONE algorithm the same parameters are used as described in [19], 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.

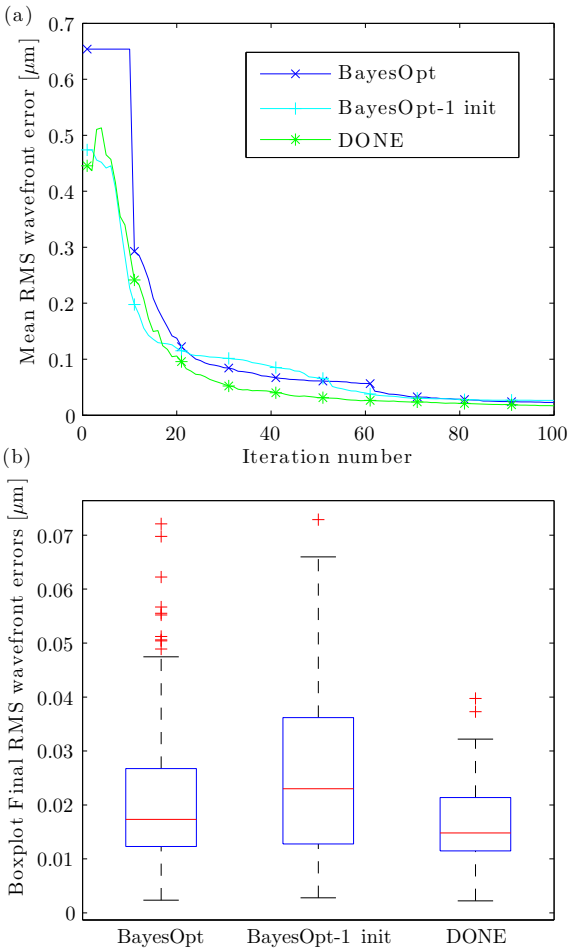

Fig. 2. (a) The RMS wavefront error of DONE and BayesOpt averaged over 100 simulations versus the number of iterations. (b) A boxplot of 100 final RMS wavefront errors after 100 iterations for DONE and BayesOpt.On each box, the central line is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers. Outliers are plotted individually.

C. 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 [23] 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 [24] (ORRs) that can perform this signal processing technique in the optical domain was proposed in [25]. 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 [28]. Here, we demonstrate the advantage of online processing in terms of performance by using DONE instead of the offline algorithm in [28].

An OBFN simulation based on the same physical models as in [28] will be used in 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 [27, 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 [13]. 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. 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 [28], 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.

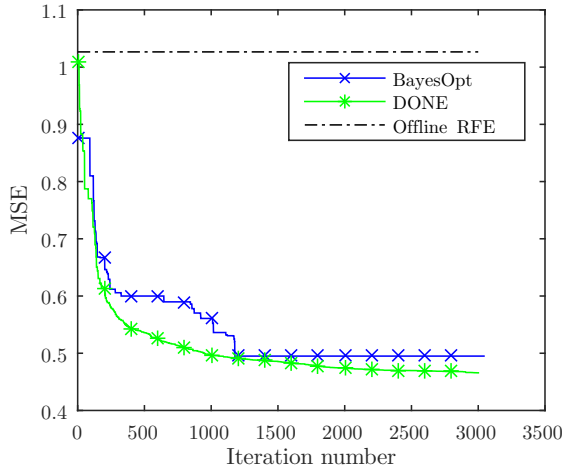

Fig. 3. The mean square error of DONE and BayesOpt applied to the OBFN application, plotted versus the number of iterations. For DONE, the values are averaged over 10 runs. For BayesOpt only 1 run is shown. The dotted line is the result of fitting a RFE using 3000 random measurements and optimizing that RFE, averaged over 10 runs.

D. 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 [29], is considered:

$$a_i(k) = u_i(k) + \sin\left(\pi/180 \sum_{j=1}^{i} \alpha_j(k-1)\right) \cdot 9.8 \cdot 0.05, \tag{41}$$

$$v_i(k) = v_i(k-1) + a_i(k), \tag{42}$$

$$\alpha_i(k) = \alpha_i(k-1) + v_i(k), \tag{43}$$

$$x(k) = \sum_{j=1}^{3} l_j \cos\left(\pi/2 + \pi/180 \sum_{j=1}^{i} \alpha_j(k)\right), \tag{44}$$

$$y(k) = \sum_{j=1}^{3} l_j \sin\left(\pi/2 + \pi/180 \sum_{j=1}^{i} \alpha_j(k)\right). \tag{45}$$

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 II. 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 4

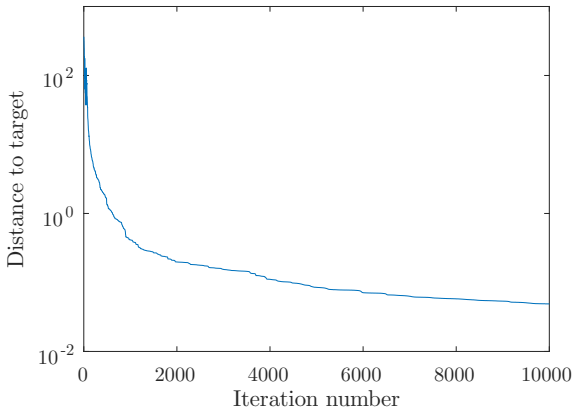

Fig. 4. The mean distance to target for the robot arm at time step 50, after minimizing this distance with DONE, plotted versus the number of iterations, averaged over 10 runs.

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 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 II.

TABLE II
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

VI. 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.

APPENDIX A
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 [42] and of [51, Thm. 2] and [52, Key Thm.].

Proof of Theorem 4. Let the constant $m > 0$ be given by

$$m = \left|\left| \left(\frac{1}{N}\mathbf{A}_N^T\mathbf{A}_N + \frac{\lambda}{N}\mathbf{I}_{D\times D} \right)^{-1} \frac{1}{N}\mathbf{A}_N^T\mathbf{y}_N \right|\right|_2, \quad (46)$$

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

$$\begin{aligned}\mathbf{c_N} &= \left(\mathbf{A}_N^T\mathbf{A}_N + \lambda\mathbf{I}_{D\times D}\right)^{-1}\mathbf{A}_N^T\mathbf{y}_N \\ &= \left(\frac{1}{N}\mathbf{A}_N^T\mathbf{A}_N + \frac{\lambda}{N}\mathbf{I}_{D\times D}\right)^{-1}\frac{1}{N}\mathbf{A}_N^T\mathbf{y}_N, \end{aligned} \quad (47)$$

is also the solution to the constrained, but unregularized least squares problem (see [53, Sec. 12.1.3])

$$\mathbf{c_N} = \underset{\mathbf{c}\in C_m}{\operatorname{argmin}} \frac{1}{N}||\mathbf{y}_N - \mathbf{A}_N\mathbf{c}||_2^2. \quad (48)$$

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 4 satisfies

$$\left|\left| \left(\frac{1}{N}\mathbf{A}_N^T\mathbf{A}_N + \Lambda\ \mathbf{I}_{D\times D} \right)^{-1} \frac{1}{N}\mathbf{A}_N^T\mathbf{y}_N \right|\right|_2 = M, \quad (49)$$

$$M = \sqrt{\sum_{k=1}^{D}\left(\frac{\bar{c}(\boldsymbol{\omega}_k, b_k)}{(2\pi)^d D p_{\boldsymbol{\Omega}}(\boldsymbol{\omega}_k) p_B(b_k)}\right)^2}, \quad (50)$$

we have that $m \geq M$. We will need this lower bound on m to make use of the results in [42] later on in this proof.

Recall from Section II-B 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

$$E(\mathbf{x}, \mathbf{c}) = f(\mathbf{x}) - \sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k). \quad (51)$$

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

$$\begin{aligned} E(\mathbf{x},\mathbf{c})^2 &= f(\mathbf{x})^2 + \left(\sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k)\right)^2 \\ &\quad - 2f(\mathbf{x})\sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k) \\ &\leq f(\mathbf{x})^2 + \left(\sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k)\right)^2 \\ &\quad + 2\left|f(\mathbf{x})\right|\left|\sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k)\right| \\ &\leq f(\mathbf{x})^2 + \sum_{k=1}^{D}|c_k|^2 + 2\left|f(\mathbf{x})\right|\sqrt{\sum_{k=1}^{D}|c_k|^2} \\ &\leq f(\mathbf{x})^2 + m^2 + 2f(\mathbf{x})m \\ &\leq \left(||f||_\infty + m\right)^2. \end{aligned} \quad (52)$$

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 [51, Thm. 2] we get, with probability one,

$$\lim_{N\to\infty}\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N}E(\mathbf{X}_n,\mathbf{c})^2 - \int_{\mathcal{X}}E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})d\mathbf{x}\right| = 0. \quad (53)$$

Since almost sure convergence implies convergence in probability [54, Ch. 2], we also have:

$$\lim_{N\to\infty} P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N} E(\mathbf{X}_n,\mathbf{c})^2 - \int_{\mathcal{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})d\mathbf{x}\right| > \epsilon\right) = 0 \ \forall \epsilon > 0. \quad (54)$$

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

$$\begin{aligned}\tilde{E}(\mathbf{x},\eta,\mathbf{c})^2 &= \left(f(\mathbf{x}) + \eta - \sum_{k=1}^{D} c_k \cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k)\right)^2 \\ &= E(\mathbf{x},\mathbf{c})^2 + 2\eta E(\mathbf{x},\mathbf{c}) + \eta^2. \end{aligned} \quad (55)$$

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

$$\begin{aligned}&\int_{\mathbb{R}}\int_{\mathcal{X}} \tilde{E}(\mathbf{x},\eta,\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta \\ &= \int_{\mathcal{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x}) \left(\int_{\mathbb{R}} p_H(\eta)d\eta\right) d\mathbf{x} \\ &\quad + 2\int_{\mathcal{X}} E(\mathbf{x},\mathbf{c}) \left(\int_{\mathbb{R}} \eta p_H(\eta)d\eta\right) p_{\mathbf{X}}(\mathbf{x})d\mathbf{x} \\ &\quad + \int_{\mathcal{X}} p_{\mathbf{X}}(\mathbf{x}) \left(\int_{\mathbb{R}} \eta^2 p_H(\eta)d\eta\right) d\mathbf{x} \\ &= \int_{\mathcal{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})d\mathbf{x} + \int_{\mathcal{X}} E(\mathbf{x},\mathbf{c}) \underbrace{\mathbb{E}[H_n]}_{=0} p_{\mathbf{X}}(\mathbf{x})d\mathbf{x} \\ &\quad + \mathbb{E}[H_n^2] \\ &= \int_{\mathcal{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})d\mathbf{x} + \sigma_H^2. \end{aligned} \quad (56)$$

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 [55, 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_{\mathcal{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}) + \frac{1}{N}\sum_{n=1}^{N} H_n^2 - \int_{\mathcal{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_{\mathcal{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})d\mathbf{x}\right| + \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_{\mathcal{X}} E(\mathbf{x},\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x})d\mathbf{x}\right| + \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_{\mathcal{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_{\mathcal{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 (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 [51, Thm. 2] again, noting that $\eta_n E(\mathbf{x}_n,\mathbf{c})$ is continuous in $\mathbf{c}$. We use (52) to get

$$|\eta E(\mathbf{x},\mathbf{c})| \le |\eta|\left(||f||_\infty + m\right) \ \forall \mathbf{x},\eta,\mathbf{c}. \quad (57)$$

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 [51, Thm. 2] gives the desired convergence in probability

$$\lim_{N\to\infty} P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N} H_n E(\mathbf{X}_n,\mathbf{c})\right| > \epsilon_2\right) = 0 \ \forall \epsilon_2. \quad (58)$$

Together with the other two convergences and (57) we get:

$$\lim_{N\to\infty} P\left(\sup_{\mathbf{c}\in C_m}\left|\frac{1}{N}\sum_{n=1}^{N}\tilde{E}(\mathbf{X}_n,H_n,\mathbf{c})^2 - \int_{\mathbb{R}}\int_{\mathcal{X}} \tilde{E}(\mathbf{x},\eta,\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta)d\mathbf{x}d\eta\right| > \epsilon\right) = 0. \quad (59)$$

The following bound follows from (52) and (56):

$$\begin{aligned}0 &\le \int_{\mathbb{R}}\int_{\mathcal{X}} \tilde{E}(\mathbf{x},\eta,\mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta)d\mathbf{x}d\eta \\ &\le \left(||f||_\infty + m\right)^2 + \sigma_H^2. \end{aligned} \quad (60)$$

In light of this bound, [52, Key Thm.] now implies that the mean square error between the output of the RFE with least squares weight vector and the noisy meansurements 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 $N > N_0$,

$$\left|\int_{\mathbb{R}}\int_{\mathcal{X}} \tilde{E}(\mathbf{x},\eta,\mathbf{C_N})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta)d\mathbf{x}d\eta - \int_{\mathbb{R}}\int_{\mathcal{X}} \tilde{E}(\mathbf{x},\eta,\mathbf{C^0})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta)d\mathbf{x}d\eta\right| < \epsilon_4 \quad (61)$$

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 $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_D, b_1, \ldots, b_D$, and $\mathbf{C^0} \in C_m$ minimizes $\int_{\mathbb{R}} \int_{\mathcal{X}} \tilde{E}(\mathbf{x}, \eta, \mathbf{c}) p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta$. 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 [42, Thm 3.2], for any $\delta_2 > 0$, with probability at least $1-\delta_2$ w.r.t. $\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_D$ and $B_1, \ldots, B_D$, there exists a $\mathbf{c} \in C_m$ with the following bound*:

$$\int_{\mathcal{X}} \left(f(\mathbf{x}) - \sum_{k=1}^{D} c_k \cos(\boldsymbol{\Omega}_k^T \mathbf{x} + B_k) \right)^2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} < \frac{\gamma(\delta_2)^2}{D},$$
$$\gamma(\delta_2) = \sup_{\boldsymbol{\omega}, b} \left| \frac{1}{(2\pi)^d} \frac{\bar{c}(\boldsymbol{\omega}, b)}{p_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) p_B(b)} \right| \left(\sqrt{\log \frac{1}{\delta_2}} + 4r \right),$$
$$r = \sup_{\mathbf{x} \in \mathcal{X}} ||\mathbf{x}||_2 \sqrt{\sigma^2 d + \pi^2/3}, \tag{62}$$

with σ^2 denoting the variance of $p_{\boldsymbol{\Omega}}$. For this particular $\mathbf{c}$, (55), (56) and (62) imply that

$$\int_{\mathbb{R}} \int_{\mathcal{X}} \tilde{E}(\mathbf{x}, \eta, \mathbf{c})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta < \frac{\gamma(\delta_2)^2}{D} + \sigma_H^2. \tag{63}$$

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

$$\int_{\mathbb{R}} \int_{\mathcal{X}} \tilde{E}(\mathbf{x}, \eta, \mathbf{C^0})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta < \frac{\gamma(\delta_2)^2}{D} + \sigma_H^2 \tag{64}$$

with probability at least $1-\delta_2$. Since the event in (64) only depends on $\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_D$ and $B_1, \ldots, B_D$, while the event in (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 $N > N_0$,

$$\int_{\mathbb{R}} \int_{\mathcal{X}} \tilde{E}(\mathbf{x}, \eta, \mathbf{C_N})^2 p_{\mathbf{X}}(\mathbf{x}) p_H(\eta) d\mathbf{x} d\eta < \epsilon_4 + \frac{\gamma(\delta_2)^2}{D} + \sigma_H^2 \tag{65}$$

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

$$\int_{\mathcal{X}} E(\mathbf{x}, \mathbf{C_N})^2 p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} < \epsilon_4 + \frac{\gamma(\delta_2)^2}{D} \tag{66}$$

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.

□

APPENDIX B
MINIMUM-VARIANCE PROPERTIES

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

*The weights found in the proof of the cited theorem satisfy $\mathbf{c} \in C_m$ if $m \geq M$, which was shown in the beginning of this appendix. Here we also made use of the result from Theorem 1 of this paper to get what is denoted with α in [42]. We have also used, with the notation of [42], that $||f - \hat{f}||_\mu \leq ||f - \hat{f}||_\infty$.

Theorem 7. *Given $\mathbf{x}$, the p.d.f. $p^*_{\boldsymbol{\Omega}}$ that minimizes the variance of the unbiased estimator $G(\mathbf{x}) = \sum_{k=1}^{D} C_k \cos(\boldsymbol{\Omega}_k^T \mathbf{x} + B_k)$ as defined in Theorem 1, with C_k as defined in Theorem 3, is equal to*

$$p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) = \frac{|\hat{f}(\boldsymbol{\omega})| \sqrt{\cos(2\angle \hat{f}(\boldsymbol{\omega}) + 2\boldsymbol{\omega}^T \mathbf{x}) + 2}}{\int_{\mathbb{R}^d} |\hat{f}(\tilde{\boldsymbol{\omega}})| \sqrt{\cos(2\angle \hat{f}(\tilde{\boldsymbol{\omega}}) + 2\tilde{\boldsymbol{\omega}}^T \mathbf{x}) + 2} d\tilde{\boldsymbol{\omega}}}. \tag{67}$$

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

$$\frac{1}{2D(2\pi)^{2d}} \left(\int_{\mathbb{R}^d} |\hat{f}(\boldsymbol{\omega})| \sqrt{\cos(2\angle \hat{f}(\boldsymbol{\omega}) + 2\boldsymbol{\omega}^T \mathbf{x}) + 2} d\boldsymbol{\omega} \right)^2 - f(\mathbf{x})^2. \tag{68}$$

Proof. The proof is similar to the proof of [48, Thm. 4.3.1]. Let $q_{\boldsymbol{\Omega}}$ be any p.d.f. of $\boldsymbol{\Omega}_k$ that satisfies $q_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) > 0$ if $|\hat{f}(\boldsymbol{\omega})| > 0$. Let $\mathrm{Var}_{q_{\boldsymbol{\Omega}}, p_B}$ be the variance of $G(\mathbf{x})$ under the assumption that $p_{\boldsymbol{\Omega}} = q_{\boldsymbol{\Omega}}$, $p_B = \mathrm{Uniform}(0, 2\pi)$, and $C_k = \frac{2}{D(2\pi)^d} \frac{|\hat{f}(\boldsymbol{\Omega}_k)|}{q_{\boldsymbol{\Omega}}(\boldsymbol{\Omega}_k)} \cos(\angle \hat{f}(\boldsymbol{\Omega}_k) - B_k)$. According to Theorem 3, 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:

$$\begin{aligned}
&\mathrm{Var}_{q_{\boldsymbol{\Omega}}, p_B}[G(\mathbf{x})] \\
&= \mathrm{Var}_{q_{\boldsymbol{\Omega}}, p_B} \left[\sum_{k=1}^{D} C_k \cos(\boldsymbol{\Omega}_k^T \mathbf{x} + B_k) \right] \\
&= D \ \mathrm{Var}_{q_{\boldsymbol{\Omega}}, p_B} \left[C_1 \cos(\boldsymbol{\Omega}_1^T \mathbf{x} + B_1) \right] \\
&= \frac{D}{2\pi} \int_{\mathbb{R}^d} \int_0^{2\pi} \left(\frac{2}{D(2\pi)^d} \frac{|\hat{f}(\boldsymbol{\omega})|}{q_{\boldsymbol{\Omega}}(\boldsymbol{\omega})} \cos(\angle \hat{f}(\boldsymbol{\omega}) - b) \right)^2 \\
&\qquad \cos(\boldsymbol{\omega}^T \mathbf{x} + b)^2 q_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) db d\boldsymbol{\omega} - f(\mathbf{x})^2.
\end{aligned} \tag{69}$$

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

$$\begin{aligned}
&\int_0^{2\pi} \cos(\angle \hat{f}(\boldsymbol{\omega}) - b)^2 \cos(\boldsymbol{\omega}^T \mathbf{x} + b)^2 db \\
&= \int_0^{2\pi} \frac{1}{4} (1 + \cos(2\angle \hat{f}(\boldsymbol{\omega}) - 2b))(1 + \cos(2\boldsymbol{\omega}^T \mathbf{x} + 2b)) db \\
&= \int_0^{2\pi} \frac{1}{4} db + \frac{1}{4} \int_0^{2\pi} \cos(2\angle \hat{f}(\boldsymbol{\omega}) - 2b) db \\
&\quad + \frac{1}{4} \int_0^{2\pi} \cos(2\boldsymbol{\omega}^T \mathbf{x} + 2b) db \\
&\quad + \frac{1}{4} \int_0^{2\pi} \cos(2\angle \hat{f}(\boldsymbol{\omega}) - 2b) \cos(2\boldsymbol{\omega}^T \mathbf{x} + 2b) db \\
&= \frac{2\pi}{4} + \frac{1}{8} \int_0^{2\pi} \cos(2\angle \hat{f}(\boldsymbol{\omega}) + 2\boldsymbol{\omega}^T \mathbf{x}) \\
&\quad + \cos(2\angle \hat{f}(\boldsymbol{\omega}) - 2\boldsymbol{\omega}^T \mathbf{x} - 4b) db \\
&= \frac{2\pi}{4} + \frac{2\pi}{8} \cos(2\angle \hat{f}(\boldsymbol{\omega}) + 2\boldsymbol{\omega}^T \mathbf{x}) \\
&= \frac{\pi}{4} (\cos(2\angle \hat{f}(\boldsymbol{\omega}) + 2\boldsymbol{\omega}^T \mathbf{x}) + 2)
\end{aligned} \tag{70}$$

we get:

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

$$
\begin{aligned}
&\cos(\boldsymbol{\omega}^T\mathbf{x}+b)^2 p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega})dbd\boldsymbol{\omega}\\
&= \frac{D}{2\pi}\int_{\mathbb{R}^d}\frac{1}{p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}\left(\frac{2}{D(2\pi)^d}\right)^2|\hat{f}(\boldsymbol{\omega})|^2\\
&\quad\int_0^{2\pi}\cos(\angle\hat{f}(\boldsymbol{\omega})-b)^2\cos(\boldsymbol{\omega}^T\mathbf{x}+b)^2 dbd\boldsymbol{\omega}\\
&= \frac{D}{2\pi}\int_{\mathbb{R}^d}\frac{1}{p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}\left(\frac{2}{D(2\pi)^d}\right)^2|\hat{f}(\boldsymbol{\omega})|^2\\
&\quad\frac{\pi}{4}(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)d\boldsymbol{\omega} \qquad (71)\\
&\overset{(67)}{=} \frac{D}{2\pi}\left(\frac{2}{D(2\pi)^d}\right)^2\\
&\quad\left(\int_{\mathbb{R}^d}|\hat{f}(\boldsymbol{\omega})|\sqrt{\frac{\pi}{4}(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)}d\boldsymbol{\omega}\right)^2\\
&= \frac{1}{2D(2\pi)^{2d}}\left(\int_{\mathbb{R}^d}|\hat{f}(\boldsymbol{\omega})|\sqrt{(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)}d\boldsymbol{\omega}\right)^2 \qquad (72)
\end{aligned}
$$

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

$$
\begin{aligned}
&\mathrm{Var}_{p^*_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})]+f(\mathbf{x})^2\\
&= \frac{D}{2\pi}\left(\frac{2}{D(2\pi)^d}\right)^2\\
&\quad\left(\int_{\mathbb{R}^d}\frac{|\hat{f}(\boldsymbol{\omega})|}{q_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}\sqrt{\frac{\pi}{4}(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)}q_{\boldsymbol{\Omega}}(\boldsymbol{\omega})d\boldsymbol{\omega}\right)^2\\
&\overset{\text{Jensen}}{\leq} \frac{D}{2\pi}\left(\frac{2}{D(2\pi)^d}\right)^2\\
&\quad\int_{\mathbb{R}^d}\frac{|\hat{f}(\boldsymbol{\omega})|^2}{q_{\boldsymbol{\Omega}}(\boldsymbol{\omega})^2}\frac{\pi}{4}(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)q_{\boldsymbol{\Omega}}(\boldsymbol{\omega})d\boldsymbol{\omega}\\
&\overset{(70)}{=} \frac{D}{2\pi}\int_{\mathbb{R}^d}\int_0^{2\pi}\left(\frac{2}{D(2\pi)^d}\frac{|\hat{f}(\boldsymbol{\omega})|}{q_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}\cos(\angle\hat{f}(\boldsymbol{\omega})-b)\right)^2\\
&\quad\cos(\boldsymbol{\omega}^T\mathbf{x}+b)^2 q_{\boldsymbol{\Omega}}(\boldsymbol{\omega})dbd\boldsymbol{\omega}\\
&\overset{(69)}{=} \mathrm{Var}_{q_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})]+f(\mathbf{x})^2. \qquad (73)
\end{aligned}
$$

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

□

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

Theorem 8. *Let $\tilde{p}_{\boldsymbol{\Omega}}$, $p^*_{\boldsymbol{\Omega}}$, $\tilde{G}$ and G be as in Theorems 5 and 7. Then*

$$
\frac{1}{\sqrt{3}}\mathbb{E}_{p^*_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})^2] \leq \mathbb{E}_{\tilde{p}_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})^2] \leq \sqrt{3}\,\mathbb{E}_{p^*_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})^2], \qquad (74)
$$

$$
\frac{1}{2}\mathbb{E}_{\tilde{p}_{\boldsymbol{\Omega}},p_B}[\tilde{G}(\mathbf{x})^2] \leq \mathbb{E}_{\tilde{p}_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})^2] \leq \frac{3}{2}\,\mathbb{E}_{\tilde{p}_{\boldsymbol{\Omega}},p_B}[\tilde{G}(\mathbf{x})^2]. \qquad (75)
$$

Proof. From

$$
1 \leq \sqrt{(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)} \leq \sqrt{3}, \qquad (76)
$$

and from (67) and (33) it follows that

$$
\begin{aligned}
&\frac{1}{\sqrt{3}}p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) \leq \tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega}) \leq \sqrt{3}p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega}),\\
&\frac{1}{\sqrt{3}}\frac{1}{p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega})} \leq \frac{1}{\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega})} \leq \sqrt{3}\frac{1}{p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}. \qquad (77)
\end{aligned}
$$

Combining the above with (71) yields:

$$
\begin{aligned}
&\frac{1}{\sqrt{3}}\mathbb{E}_{p^*_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})^2]\\
&= \frac{1}{\sqrt{3}}\frac{1}{2D(2\pi)^{2d}}\\
&\quad\int_{\mathbb{R}^d}\frac{1}{p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}|\hat{f}(\boldsymbol{\omega})|^2(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)d\boldsymbol{\omega}\\
&\leq \frac{1}{2D(2\pi)^{2d}}\\
&\quad\int_{\mathbb{R}^d}\frac{1}{\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}|\hat{f}(\boldsymbol{\omega})|^2(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)d\boldsymbol{\omega}\\
&= \mathbb{E}_{\tilde{p}_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})^2]\\
&\leq \sqrt{3}\frac{1}{2D(2\pi)^{2d}}\\
&\quad\int_{\mathbb{R}^d}\frac{1}{p^*_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}|\hat{f}(\boldsymbol{\omega})|^2(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)d\boldsymbol{\omega}\\
&= \sqrt{3}\;\mathbb{E}_{p^*_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})]. \qquad (78)
\end{aligned}
$$

Combining (76) with (34) yields:

$$
\begin{aligned}
&\frac{1}{2}\mathbb{E}_{\tilde{p}_{\boldsymbol{\Omega}}}[\tilde{G}(\mathbf{x})^2]\\
&= \frac{1}{2D(2\pi)^{2d}}\int_{\mathbb{R}^d}\frac{1}{\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}|\hat{f}(\boldsymbol{\omega})|^2 d\boldsymbol{\omega}\\
&\leq \frac{1}{2D(2\pi)^{2d}}\int_{\mathbb{R}^d}\frac{1}{\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}|\hat{f}(\boldsymbol{\omega})|^2\\
&\quad(\cos(2\angle\hat{f}(\boldsymbol{\omega})+2\boldsymbol{\omega}^T\mathbf{x})+2)d\boldsymbol{\omega}\\
&= \mathbb{E}_{\tilde{p}_{\boldsymbol{\Omega}},p_B}[G(\mathbf{x})^2]\\
&\leq \frac{3}{2D(2\pi)^{2d}}\int_{\mathbb{R}^d}\frac{1}{\tilde{p}_{\boldsymbol{\Omega}}(\boldsymbol{\omega})}|\hat{f}(\boldsymbol{\omega})|^2 d\boldsymbol{\omega}\\
&= \frac{3}{2}\mathbb{E}_{\tilde{p}_{\boldsymbol{\Omega}}}[\tilde{G}(\mathbf{x})^2]. \qquad (79)
\end{aligned}
$$

□

ACKNOWLEDGMENT

This research was supported by the Netherlands Enterprise Agency (RVO) for Innovation in Photonic Devices (IPD12020), by the European Research Council Advanced Grant Agreement (No. 339681) and by the Dutch Technology Foundation STW (project 13336).

REFERENCES

[1] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*. Siam, 2009, vol. 8.

[2] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *J. Global Optim.*, vol. 56, no. 3, pp. 1247–1293, 2013.

[3] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.

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

[5] ——, "The BOBYQA algorithm for bound constrained optimization without derivatives," 2009.

[6] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *J. Optimiz. Theory App.*, vol. 79, no. 1, pp. 157–181, 1993.

[7] P. Gilmore and C. T. Kelley, "An implicit filtering algorithm for optimization of functions with many local minima," *SIAM J. Optimiz.*, vol. 5, no. 2, pp. 269–285, 1995.

[8] A. L. Custódio and L. N. Vicente, "Using sampling and simplex derivatives in pattern search methods," *SIAM J. Optimiz.*, vol. 18, no. 2, pp. 537–555, 2007.

[9] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[10] 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.

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

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

[13] R. Martinez-Cantin, "BayesOpt: a Bayesian optimization library for nonlinear optimization, experimental design and bandits," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3735–3739, 2014.

[14] O. Roustant, D. Ginsbourger, and Y. Deville, "Dicekriging, Diceoptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization," 2012.

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

[16] 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 preprint arXiv:1012.2599*, 2010.

[17] R. Martinez-Cantin, N. 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*, vol. 27, no. 2, pp. 93–103, 2009.

[18] S. ur Rehman and M. Langelaar, "Efficient global robust optimization of unconstrained problems affected by parametric uncertainties," *Struct. Multidiscip. O.*, pp. 1–18, 2015.

[19] H. R. G. W. Verstraete, S. Wahls, J. Kalkman, and M. Verhaegen, "Model-based sensor-less wavefront aberration correction in optical coherence tomography," *Opt. Lett.*, vol. 40, no. 24, pp. 5722–5725, Dec 2015.

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

[21] 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)*, pp. 669–674, 2009.

[22] S. Bonora and R. Zawadzki, "Wavefront sensorless modal deformable mirror correction in adaptive optics: optical coherence tomography," *Opt. Lett.*, vol. 38, no. 22, pp. 4801–4804, 2013.

[23] R. C. Hansen, *Phased array antennas*. John Wiley & Sons, 2009, vol. 213.

[24] C. Roeloffzen, L. Zhuang, R. Heideman, A. Borreman, and v. W. Etten, "Ring resonator-based tunable optical delay line in LPCVD waveguide technology," 2005.

[25] 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 antennaspart i: Design and performance analysis," *J. Lightwave Technol.*, vol. 28, no. 1, pp. 3–18, 2010.

[26] 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 *Microwave Photonics, 2006. MWP'06. International Topical Meeting on*. IEEE, 2006, pp. 1–4.

[27] L. Zhuang, *Ring resonator-based broadband photonic beam former for phased array antennas*. University of Twente, 2010.

[28] L. Bliek, M. Verhaegen, and S. Wahls, "Data-driven minimization with random feature expansions for optical beam forming network tuning," *IFAC-PapersOnLine*, vol. 48, no. 25, pp. 166 – 171, 2015, 16th {IFAC} Workshop on Control Applications of Optimization CAO2015Garmisch-Partenkirchen, Germany, 69 October 2015.

[29] J. de Lope, M. Santos *et al.*, "A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers," *Neurocomputing*, vol. 72, no. 13, pp. 2806–2814, 2009.

[30] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Stat.*, pp. 1171–1220, 2008.

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

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

[33] 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.

[34] 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.

[35] C. J. Burges *et al.*, "Simplified support vector decision rules," in *ICML*, vol. 96. Citeseer, 1996, pp. 71–77.

[36] D. Schölkopf, "Sampling techniques for kernel methods," in *Adv. Neur. In.*, vol. 1. MIT Press, 2002, p. 335.

[37] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939–1959, 2005.

[38] B. Chen, S. Zhao, P. Zhu, and J. C. Principe, "Quantized kernel recursive least squares algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1484–1491, 2013.

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

[40] F. Girosi and G. Anzellotti, "Convergence rates of approximation by translates," DTIC Document, Tech. Rep., 1992.

[41] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–945, 1993.

[42] 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.

[43] 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.*, pp. 608–613, 1992.

[44] A. H. Sayed and T. Kailath, "Recursive least-squares adaptive filters," *Digit. Signal Process. Handbook*, pp. 21–1, 1998.

[45] J. Nocedal, "Updating quasi-Newton matrices with limited storage," *Math. Comp.*, vol. 35, no. 151, pp. 773–782, 1980.

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

[47] M. Pogu and J. S. De Cursi, "Global optimization by random perturbation of the gradient method with a fixed parameter," *J. of Global Optim.*, vol. 5, no. 2, pp. 159–180, 1994.

[48] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*. John Wiley & Sons, 2011, vol. 707.

[49] 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.

[50] H. R. G. W. Verstraete, B. Cense, R. Bilderbeek, M. Verhaegen, and J. Kalkman, "Towards model-based adaptive optics optical coherence tomography," *Opt. Express*, vol. 22, no. 26, pp. 32 406–32 418, Dec 2014.

[51] R. I. Jennrich, "Asymptotic properties of non-linear least squares estimators," *Ann. Math. Stat.*, pp. 633–643, 1969.

[52] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, 1999.

[53] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[54] A. W. Van der Vaart, *Asymptotic statistics*. Cambridge university press, 2000, vol. 3.

[55] A. Beitollahi and P. Azhdari, "Convergence in probability and almost surely convergence in probabilistic normed spaces," *Math. Sci.*, vol. 6, no. 1, pp. 1–5, 2012.