Cost-aware Adaptive Sampling for Environmental Sensing

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von

M.Sc. Johannes Westermann

geb. am 01.09.1991
in Rastatt

Gutachter

Prof. Dr.-Ing. Benjamin Noack
Prof. Dr. habil. Lino José Forte Marques
Dr. Jan-Peter Calliess

Eingereicht am 29. Juli 2024
Verteidigt am 22. November 2024

Abstract

Motivated by advancing climate change and ever-increasing environmental pollution, our research is concerned with cost-aware adaptive sampling methods for environmental sensing. We consider the problem of reconstructing unknown environmental phenomena (black-box functions) based on samples taken by a mobile robot, where both the sampling process and the robot's travel incur substantial costs. Existing methods from the field of global metamodeling and Bayesian optimization focus on sampling of potentially complex black-box functions (e.g., functions containing jumps) on mixed-domain (continuous, discrete, categorical, etc.) and often high-dimensional design spaces. However, they do not take into account application-specific conditions relevant in environmental sensing, such as the robot's travel costs, characteristics of the design space, and properties of the environmental phenomenon. Advances in battery and sensor technology have facilitated the increased use of mobile robots in environmental sensing. Applied methods utilize real-time capable sensors to record numerous samples along the robot's path. However, these methods are unsuitable for scenarios where each sample incurs a significant cost. We address this gap by proposing novel cost-aware space-filling and adaptive sampling methods, specifically designed for their application in environmental sensing.

Initially, we consider the metamodeling of Lipschitz-continuous black-box functions on low-dimensional design spaces with equally scaled dimensions. For this we derive three novel space-filling sampling criteria that outperform existing methods with respect to the resulting global accuracy of the metamodel. Subsequently, we examine the sensing of positive-valued environmental phenomena, such as concentration distributions. For this, we employ adaptive sampling, which leverages the available information about the black-box function to select sample points, obtaining more information per sample compared to space-filling sequential methods. We introduce the novel class of weighted explorative sampling criteria (WESC) that takes into account the properties of the considered type of environmental phenomena, thereby outperforming existing adaptive sampling criteria. Finally, we extend the space-filling and adaptive sampling methods to take into account the location-dependent sampling costs and the robot's travel costs. For this purpose, we define a Pareto-optimal sampling criterion for each scenario: considering location-dependent costs, travel costs, and both costs combined. Moreover, we develop a receding horizon cost-aware adaptive sampling (RHCaAS) algorithm, which drastically reduces travel costs without significantly impacting metamodel accuracy. Subsequently, we combine the methods from both considerations to account for both types of costs in a manner that optimally leverages their strengths.

All the presented methods are evaluated in extensive numerical simulations, demonstrating their superiority over those found in the literature. We expect that the methods presented will increase the information yield in environmental sensing and lead to considerable cost savings, especially when using mobile robots for the evaluation of expensive-to-evaluate environmental phenomena. This applies in particular to decontamination in hostile environments, seabed sampling by diving robots, or extraterrestrial observation, such as with a Mars rover.

Zusammenfassung

Motiviert durch den fortschreitenden Klimawandel und die zunehmende Umweltverschmutzung befasst sich die vorliegende Arbeit mit kosteneffizienten, adaptiven Samplingverfahren für die Umwelterfassung. Wir betrachten das Problem der Rekonstruktion unbekannter Umweltphänomene (Black-Box-Funktionen) auf der Grundlage von Messungen, welche von einem mobilen Roboter durchgeführt werden, wobei sowohl die Messung als auch die Bewegung des Roboters erhebliche Kosten verursachen. Bestehende Methoden aus dem Bereich der globalen Metamodellierung und der Bayes'schen Optimierung befassen sich mit potenziell komplexen Black-Box-Funktionen (z.B. Funktionen, die Sprünge enthalten) auf gemischten (kontinuierlichen, diskreten, kategorialen, etc.) und oft hochdimensionalen Parameterräumen. Allerdings berücksichtigen diese nicht die für die Umwelterfassung relevanten anwendungsspezifischen Rahmenbedingungen, wie z.B. die Reisekosten des Roboters, die Merkmale des Parameterraums und die Eigenschaften des Umweltphänomens. Durch die Fortschritte in der Batterie- und Sensortechnologie werden vermehrt mobile Roboter zur Umwelterfassung eingesetzt. Die dazu verwendeten Methoden setzen auf am Roboter angebrachte, echtzeitfähige Sensoren, mit welchen zahlreiche Messungen während der Bewegung des Roboters entlang dessen Route aufgezeichnet werden. Diese Methoden sind jedoch ungeeignet für Anwendungen, bei denen jede Messung mit erheblichen Kosten verbunden ist. Mit unseren neuartigen, speziell für die Anwendung in der Umwelterfassung entwickelten, kosteneffizienten, raumfüllenden und adaptiven Samplingmethoden schließen wir diese Lücke.

Zunächst betrachten wir die Metamodellierung von Lipschitz-stetigen Black-Box-Funktionen auf niederdimensionalen Parameterräumen, deren Dimensionen dieselbe Skalierung aufweisen. Wir leiten drei neuartige raumfüllende Samplingkriterien her, welche existierende Verfahren bezüglich der resultierenden globalen Genauigkeit des Metamodells übertreffen. Anschließend untersuchen wir adaptive Samplingverfahren zur Erfassung positivwertiger Umweltphänomene (z.B. Konzentrationsverteilungen). Durch die Berücksichtigung vorhandener Informationen über die Black-Box-Funktion erzielen adaptive Samplingverfahren einen höheren Informationsgehalt pro Messung und somit eine bessere Modellgenauigkeit für eine gegebene Anzahl von Messungen. Wir führen die Klasse der gewichteten explorativen Samplingkriterien (WESC) ein, welche durch die Berücksichtigung der Eigenschaften der betrachteten Umweltphänomene existierende adaptive Samplingkriterien bezüglich der resultierenden

Modellgenauigkeit übertreffen. Zuletzt erweitern wir die raumfüllenden und adaptiven Samplingverfahren um die Berücksichtigung der ortsabhängigen Samplingkosten sowie der Reisekosten des mobilen Roboters. Dazu definieren wir jeweils ein Pareto-optimales Samplingkriterium für die Berücksichtigung der ortsabhängigen Kosten, der Reisekosten sowie der gleichzeitigen Berücksichtigung beider Kostenarten. Zudem entwickeln wir den RHCaAS-Algorithmus (Receding Horizon Cost-aware Adaptive Sampling), der eine drastische Reduktion der Reisekosten ohne wesentliche Auswirkung auf die Genauigkeit des Metamodells ermöglicht. Abschließend kombinieren wir RHCaAS mit dem Pareto-optimalen Kriterium auf eine Weise, welche die Stärken beider Ansätze optimal zur Berücksichtigung beider Kostenarten vereint.

Alle vorgestellten Methoden werden in umfangreichen numerischen Simulationen evaluiert sowie deren Überlegenheit gegenüber den Literaturverfahren demonstriert. Wir erwarten, dass die vorgestellten Methoden die Informationsausbeute bei der Umwelterfassung erhöhen und zu erheblichen Kosteneinsparungen führen werden. Dies gilt insbesondere beim Einsatz mobiler Roboter für die Rekonstruktion von Umweltphänomenen, deren Messung mit signifikanten Kosten verbunden ist. Beispiele stellen die Dekontamination in lebensfeindlichen Umgebungen, die Entnahme von Proben auf dem Meeresgrund durch Tauchroboter oder die extraterrestrische Beobachtung (z. B. mit einem Mars-Rover) dar.

Contents

Abstract **i**

Zusammenfassung **iii**

Contents **iv**

List of Figures **vii**

Notation **xii**

1 Introduction **1**

2 Review on Adaptive Sampling for Global Metamodeling **7**

2.1 Gaussian Process Regression 9

2.2 Sampling Methods 15

2.2.1 Space-filling One-shot Designs 15

2.2.2 Space-filling Sequential Sampling 21

2.2.3 Adaptive Sampling 24

3 Space-filling Designs for Environmental Sensing **31**

3.1 Novel Space-filling Sampling Criteria for Lipschitz-continuous Black-box Functions 32

3.1.1 Mean Distance Criterion 34

3.1.2 Root Mean Squared Distance Criterion 35

3.1.3 Modified Maximin Criterion 36

3.1.4 Modified Regular Grid Sampling 37

3.2 Evaluation 38

3.2.1 Experimental Setup 38

3.2.2 Properties of the MD and RMSD Designs 41

3.2.3 Numerical Comparison to Literature Methods 43

3.3 Summary 44

4 Weighted Explorative Sampling Criteria for Environmental Sensing 49

4.1 Problem Formulation 51

4.2 Weighted Explorative Sampling Criteria 52

4.2.1 Explorative Sampling Criteria 52

4.2.2 Weighted Explorative Sampling Criteria 53

4.3 Evaluation 55

4.3.1 Experimental Setup 55

4.3.2 Numerical Comparison to Literature Methods 58

4.4 Summary 58

5 Cost-aware Adaptive Sampling for Environmental Sensing 61

5.1 Pareto-optimal Multi-objective Optimization 62

5.2 Experimental Setup 64

5.3 Travel Costs in Adaptive Sampling for Environmental Sensing Using Robots 67

5.3.1 Problem Formulation 69

5.3.2 Pareto-optimal Consideration of Travel Cost 70

5.3.3 Receding Horizon Cost-aware Adaptive Sampling 71

5.3.4 Evaluation 74

5.4 Sampling Costs in Adaptive Sampling for Environmental Sensing Using Robots 78

5.4.1 Problem Formulation 79

5.4.2 Pareto-optimal Cost-aware Adaptive Sampling 80

5.4.3 Evaluation 81

5.5 Combined Costs in Adaptive Sampling for Environmental Sensing Using Robots 85

5.5.1 Receding Horizon Pareto-optimal Cost-aware Adaptive Sampling 87

5.5.2 Evaluation 87

5.6 Summary 88

6 Conclusions 91

6.1 Summary 91

6.2 Future Research 93

A Contextualization of Our Research within Artificial Intelligence 99

Bibliography 105

List of Figures

1.1 Metamodeling of an unknown function $f(\underline{x})$ (black-box function) based on a space-filling initial design and an adaptive design. 2

1.2 Simplified adaptive sampling procedure for environmental sensing. . 3

2.1 Conditioning of a GP on noise free samples $y_l = f(x)$ of an unknown function $f(x)$. 8

2.2 Impact of the hyperparameter of an SE kernel on the characteristics of the resulting GP. 12

2.3 Dependence of optimized hyperparameters of a kernel in GP regression on the training data. 13

2.4 Properties of GP regression when optimizing the sampling noise variance along with the hyperparameters of the kernel. 14

2.5 Two-dimensional RGS with four levels per dimension on the left and three or five levels per dimension on the right. 15

2.6 Two space-filling designs of 16 and 49 points created with Hammersley sampling on a two-dimensional design space. 16

2.7 Two optimized maximin designs of 16 and 49 points on a two-dimensional design space. 17

2.8 Two minimax designs of 16 and 49 points on a two-dimensional design space. 18

2.9 Clipping of a Voronoi tessellation of 9 points to a two-dimensional design space. 19

2.10 Voronoi tessellation of 9 random sample points (left) and a CVT of size 9 (right). 19

2.11 Properties of a Latin hypercube illustrated by two Latin hypercubes of 9 points on a two-dimensional design space. 20

2.12 Optimal Latin hypercube sampling of 16 and 49 points on a two-dimensional design space. 20

2.13 General procedure of sequential sampling for global metamodeling. . 21

2.14 Two sequential maximin designs with 16 sample points initialized with different initial sample points. 22

2.15 Sequential random sampling of 16 samples on the left and 49 samples on the right. 23

2.16 Sequential maximin designs of 16 samples on the left and 49 samples on the right. 23

2.17 Illustration of the Voronoi sampling procedure. 24

2.18 Two space-filling designs generated by sequential Voronoi sampling of 16 and 49 points on a two-dimensional design space. 25

2.19 General procedure of adaptive sampling for global metamodeling. . . 26

2.20 Selection of the next sample point using a continuous sampling criterion in adaptive sampling. 27

2.21 Adaptive design of 33 points based on the MV criterion, warm-started with a space-filling initial design of 16 points. 28

2.22 Adaptive design of 33 points based on the IMSE criterion, warm-started with a space-filling initial design of 16 points. 28

2.23 Adaptive design of 33 points based on the CVV criterion, warm-started with a space-filling initial design of 16 points. 29

3.1 One-shot and sequential MD design of 7 points on a two-dimensional design space. 34

3.2 One-shot and sequential RMSD design of 7 points on a two-dimensional design space. 35

3.3 One-shot and sequential mod. maximin design of 7 points on a two-dimensional design space. 37

3.4 Mod. RGS (left) and RGS (right) with 16 sample points. 38

3.5 One-shot and sequential MD and RMSD designs for different numbers m of sample points in square. 40

3.6 One-shot MD and RMSD designs for different numbers m of sample points in square. 41

3.7 Illustration of the MD and RMSD criterion evaluated on Hammersley sample points. 42

3.8 Illustration of the MD and RMSD criterion evaluated on random sample points. 43

3.9 Evaluation of the MD and RMSD designs over the number of sample points based on the error measures NMAE and NRMSE. 44

3.10 Evaluation of the one-shot designs from the literature over the number of sample points based on the NRMSE. 45

3.11 Evaluation of the sequential designs from the literature over the number of sample points based on the NRMSE. 45

3.12 Two one-shot mod. maximin designs of 36 and 81 points on a two-dimensional design space. 46

3.13 Comparison of the one-shot mod. maximin and MD design to the literature methods by the NRMSE. 46

3.14 Comparison of the sequential mod. maximin and MD design to the literature methods by the NRMSE. 47

4.1 Illustration of the principle behind the WMV criterion. 54

4.2 Illustration of the WMV and WMM criterion evaluated on random sample points. 55

4.3 Illustration of the adaptive WMV, WMM, CVV, and MV designs on top of the black-box function. 57

4.4 Evaluation of the adaptive WMV, WMM, CVV, and MV designs over the number of sample points based on the NRMSE and NWRMSE. . 59

5.1 Illustration of dominance relationships in multi-objective optimization. 63

5.2 Decision space, objective space, and the Pareto front in multi-objective optimization. 64

5.3 Real sampling analogous to the adaptive sampling procedure in Sec. 2.2.3. Virtual sampling is performed to determine the next sample point in each iteration. 72

5.4 Virtual sampling procedure of RHCaAS with a horizon length of H virtual samples, starting with the current design $\mathcal{D}_i$ as initial design. 73

5.5 Illustration of travel cost savings of the receding horizon cost-aware adaptive sampling (RHCaAS) algorithm. 74

5.6 Evaluation of the consideration of the robot's travel costs by the NRMSE over the number of samples. 76

5.7 Evaluation of the relative mean cumulative travel cost savings over the number of samples. 77

5.8 Evaluation of the location-dependent sampling cost consideration of the space-filling designs. 82

5.9 Evaluation of the location-dependent sampling cost consideration of the adaptive designs. 84

5.10 Evaluation of the location-dependent sampling cost consideration of the adaptive designs when the sampling cost function is learned. . . 86

5.11 Evaluation of the simultaneous consideration of location-dependent sampling costs and the robot's travel costs. 89

A.1 Metamodeling of an unknown function $f(\underline{x})$ (black-box function) based on a space-filling initial design and an adaptive design. 100

A.2 Simplified adaptive sampling procedure for environmental sensing. . 101

Notation

General Notations and Operators

x	Scalar
$\underline{x}$	Vector
$\underline{x}[j]$	jth element of the vector $\underline{x}$
$\boldsymbol{x}$	Matrix
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$(\cdot)_*$	Posterior quantities in Gaussian process regression
$(\cdot)^*$	Optimal solution of an optimization problem
$(\hat{\cdot})$	Predictive / estimated quantity
$\sim$	Distribution operator
$\wedge$	Logical conjunction
$\prec$	Pareto dominance relationship
$\mathrm{E}\left\{\cdot\right\}$	Expectation operator
$\lvert x \rvert$	Absolut value of x
$\lvert\lvert \underline{x} \rvert\rvert$	Euclidean norm of the vector $\underline{x}$
$(\cdot)^{-1}$	Inversion operator

Symbols and Conventions

$\underline{b}$	Point on the boundary of the design space
$c^x(\cdot)$	Function of location-dependent sampling cost
$\bar{c}^x$	Average of $c^x(\underline{x})$ on the design space
$c^{\Delta x}(\cdot, \cdot)$	Travel cost function
$d(\cdot)$	Distance function
$\mathcal{D}$	(Experimental) design
$\mathcal{D}_0$	Initial (experimental) design
$f(\cdot)$	Black-box function
$\hat{f}(\cdot)$	Metamodel
$g(\cdot)$	Gaussian process
$\underline{g}$	Vector of random variables

$\mathcal{GP}$	Gaussian Process
$\mathcal{H}_b(\cdot)$	Van der Corput sequence in base b
i	Indexing variable
$\boldsymbol{I}$	Identity matrix
j	Indexing variable
$J(\cdot)$	Objective function
$\underline{J}(\cdot)$	Vector-valued mapping
$\underline{J}^{\text{ideal}}$	Ideal point
$\underline{J}^{\text{nadir}}$	Nadir point
$\mathcal{J}$	Image of $\underline{J}(\cdot)$
$\mathcal{J}^*$	Pareto front
$k(\cdot,\cdot)$	Covariance function (kernel) of the Gaussian process
$\boldsymbol{K}(\cdot,\cdot)$	Batch evaluation of k
$\boldsymbol{K}$	Covariance matrix
l	Indexing variable
L	Lipschitz constant
m	Current number of sample points
M	Total number of sample points
M_0	Total number of initial sample points
n	Number of dimensions
N^{runs}	Number of runs of the numerical simulation
N^{eval}	Number of evaluation points
$\mathcal{N}$	Normal distribution
$p(\cdot)$	Probability distribution
q	Indexing variable
Q	Number of objective functions in a MOP
R_i	Voronoi region (or cell)
$s(\cdot)$	Scalarization function
$S(\cdot)$	Selection criterion
$\mathcal{S}$	Image of the scalarization function
$\underline{v}[j]$	Hypervolume of Voronoi region R_j
$\underline{v}$	Vector of hypervolumes of the Voronoi regions
w	Scalar weight
$\underline{w}$	Vector of weights
x	Scalar input
$\underline{x}$	Vector input
$\boldsymbol{X}$	Matrix of input vectors $[\underline{x}_1,\ldots,\underline{x}_M]^\top$
X	Set of points
$\mathcal{X}$	Design space
$\mathcal{X}^*$	Pareto set
$\mathcal{X}^{texteval}$	Finite discrete set of evaluation points

y	Scalar sample
$\underline{y}$	Vector of samples
$\underline{z}_i$	Mass centroid of the Voronoi cell R_i
α	Parameter of the WMV criterion
γ	Trade-off parameter
ϵ	Scalar noise term
η	Indexing variable
θ	Hyperparameter
$\boldsymbol{\Theta}$	Matrix of hyperparameters
λ	Threshold parameter
$\mu(\cdot)$	Mean function of the Gaussian process
$\underline{\mu}(\cdot)$	Batch evaluation of $\mu(\cdot)$
π	Mathematical constant (≈ 3.14159)
$\boldsymbol{\pi}$	Permutation
$\boldsymbol{\pi}_i$	i-th element of the permutation $\boldsymbol{\pi}$
$\rho(\cdot)$	Probability density function
σ	Standard deviation
σ_f	Hyperparameter
$\boldsymbol{\Sigma}$	Covariance matrix of sampling noise
γ	Parameter of CArBO criterion
$\varphi(\cdot)$	Sequential or adaptive criterion
$\phi(\cdot)$	One-shot criterion

Abbreviations

AI	Artificial intelligence
ANN	Artificial neural network
AUV	Autonomous underwater vehicle
BO	Bayesian optimization
CArBO	Cost appointed Bayesian optimization
CAV	Cost-aware Voronoi sampling
CeID	Cost-effective initial design (of CArBO paper)
CT	Computed tomography
CVT	Centroidal Voronoi tessellation
CVV	Leave-one-out cross-validation Voronoi
DoE	Design of experiments
EI	Expected improvement
EIGF	Expected improvement global fit
EIpu	Expected improvement per unit
ESC	Explorative sampling criterion

GA	Genetic algorithm
GNN	Graph neural network
GP	Gaussian process
GPR	Gaussian process regression
IDAES	Institute for the design of advanced energy systems
IMSE	Integrated mean squared error
LH	Latin hypercube
LHS	Latin hypercube sampling
MAE	Mean absolute error
MCTC	Mean cumulative travel cost
MDP	Markov decision procedure
Rel. MCTCS	Relative mean cumulative travel cost saving
MD	Mean distance
ML	Machine learning
MMSE	Maximum mean squared error
Mod.	Modified
MOP	Multi-objective optimization problem
MRI	Magnetic resonance imaging
MSE	Mean squared error
MV	Maximum variance
NMAE	Normalized mean absolute error
NMCPS	Normalized mean cost per sample
NRMSE	Normalized root mean squared error
NWRMSE	Normalized weighted root mean squared error
OLHS	Optimal Latin hypercube sampling
PI	Probability of improvement
RBF	Radial basis function
RGS	Regular grid sampling
RHCaAS	Receding horizon cost-aware adaptive sampling
RHPoCaAS	Receding horizon Pareto-optimal cost-aware adaptive sampling
RL	Reinforcement learning
RMSD	Root mean squared distance
RMSE	Root mean squared error
SAR	synthetic aperture radar
SE	Squared exponential
Seq.	Sequential
SMT	Python surrogate modeling toolbox
SOP	Multi-objective optimization problem
UCB	Upper confidence bound

USV	Unmanned surface vehicles
WESC	Weighted explorative sampling criterion
WMM	Weighted square modified maximin
WMV	Weighted maximum variance
WRMSE	Weighted root mean squared error
w.r.t.	With respect to

CHAPTER 1

Introduction

Environmental sensing is becoming increasingly important in the context of global warming, the associated climate change, and the steadily increasing environmental pollution. Remote sensing is invaluable for large-scale and long-term monitoring, offering a comprehensive overview of environmental conditions using imagery such as hyperspectral, multispectral, high-resolution panchromatic, and synthetic aperture radar (SAR) images, which are commonly captured by satellites as well as airborne platforms including unmanned aerial vehicles (UAVs) and aircraft [1]. Examples include land use classification based on satellite data [2, 3] and the assessment of environmental damage using image data [1, 4], both leveraging deep artificial neural networks trained on large amounts of data.

In addition to remote sensing, advances in sensor technology, wireless communication networks, and the miniaturization of computing and storage units have enhanced in-situ measurements for sensing various physical, chemical, and biological parameters, thereby enabling faster, more targeted, and hence more effective interventions in response to problematic environmental changes, pollution, disasters, etc. [5–8].

Mobile robots play a crucial role in performing in-situ measurements, as they can operate in hostile environments, are often more cost-effective than manned missions, and can flexibly navigate to required sample points. Equipped with sensors, mobile robots are employed for tasks such as detecting gas leaks indoors [9], measuring soil properties in agriculture [10, 11], or tracking phytoplankton blooms [12, 13]. Existing solutions utilize real-time capable sensors that permit samples at regular intervals (such as a specific sampling rate) during the robot's motion. However, there are also applications where the robot has to interrupt its motion to perform expensive sampling (in terms of energy and/or time) and the number of samples is limited by constrained resources (e.g., battery capacity, time limit).

A prominent example is the ROBDEKON project [14], which investigates the use of autonomous construction machines for the decontamination of contaminated sites. The autonomous construction machine performs expensive and time-consuming dynamic probing to collect samples, which are then used to reconstruct the unknown

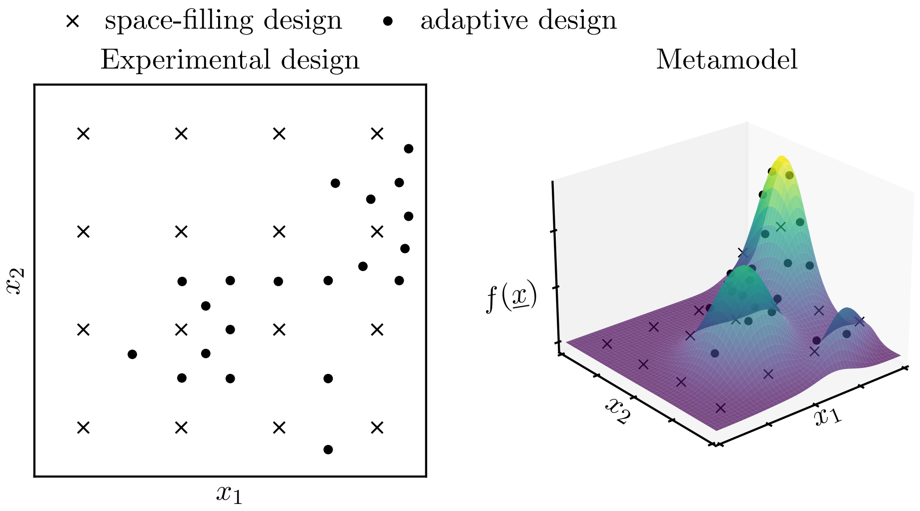

Figure 1.1: Metamodeling of an unknown function $f(\underline{x})$ (black-box function) involving a space-filling initial design (crosses) and an adaptive design (dots) on a two-dimensional parameter space (design space). The adaptive design iteratively exploits the existing knowledge about the black-box function to intensify sampling in regions of interest.

distribution of pollutants in the soil. This reconstruction allows for the precise identification and removal of regions exceeding legal contamination limits, ensuring effective site decontamination.

The project illustrates that there may only be a limited number of expensive samples available for modeling the phenomenon, necessitating a preference for data-efficient methods over large artificial intelligence (AI) models, such as those used in satellite data processing. Additionally, there may be location-dependent differences in sampling costs due to variations in soil composition, slopes, vegetation, or even obstacles. Furthermore, travel costs incurred by the mobile robot between sample points may vary significantly depending on the sequence of the points.

The process of reconstructing an unknown function (black-box function) based on a finite set of expensive samples is referred to as metamodeling [15, 16], as illustrated in Fig. 1.1 on the right. A metamodel (also known as surrogate model or response surface model [16]) represents the relationships between target variables (outputs) and parameters (inputs) in physical experiments (e.g., wind tunnel tests [17]) and computer experiments (e.g., numerical simulations [18]) [15, 16]. In the context of industry 4.0 metamodels are also referred to as digital twins [19] with applications in manufacturing, smart cities, healthcare and retail [20]. The rapidly evaluable metamodel mimics the physical relationships across the entire parameter space and serves as a surrogate for the experiments [21].

When using the metamodel as a substitute for the black-box function, ensuring sufficient model accuracy is of critical importance. However, the accuracy of the

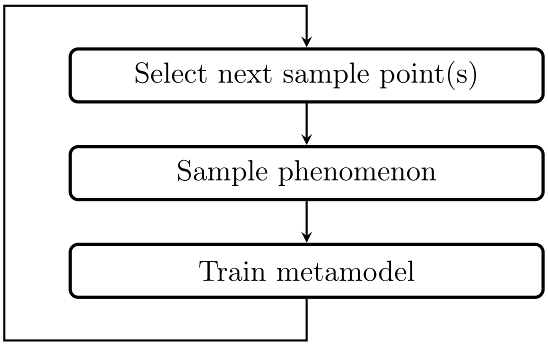

Figure 1.2: Simplified adaptive sampling procedure for environmental sensing.

metamodel depends crucially on the selection of the points in the parameter space at which the black-box function is evaluated [15, 16, 22, 23]. Design of Experiments (DoE) represents a class of methods that generate parameter combinations to accurately capture the variations of the black-box function within the parameter space [24]. In this process, adaptive sampling (also known as active learning [21]) techniques are utilized to select the parameter combinations for training the metamodel in such a way that it achieves maximum model accuracy or identifies the optimum of the underlying causal relationships [15, 25]. In some cases, cost-aware methods are applied, aiming to achieve the best possible results within a given budget by considering the parameter-dependent experimental costs [17, 26].

Thanks to advances in mobile robotics, battery technology, and sensor technology, adaptive sampling methods are becoming increasingly important in environmental sensing and monitoring. As shown in Fig. 1.2, the adaptive sampling procedure iterates by selecting the next sample point, sampling the environmental phenomenon at the selected point, and then training the new data points into the metamodel. This sequential approach allows for the selection of subsequent sample points to be adapted based on the most recent knowledge about the phenomenon, thereby gaining more information per sample than would be obtained using non-adaptive sampling methods (e.g., distance-based space-filling designs [21, 27]) [15]. An example of a space-filling initial design and an adaptive design is illustrated in Fig. 1.1.

Adaptive sampling was employed in environmental sensing for the exploration and reconstruction of scalar fields (such as depth maps or concentration distributions) using unmanned surface vehicles (USV) [28–30]. However, as in the aforementioned examples, real-time capable sensors were employed, which recorded samples at regular intervals along the robot's route and while it was in motion. In contrast, in the context of global metamodeling, expensive (adaptive) sample points are chosen to maximize the information gained about the black-box function per sample. Furthermore, global metamodeling typically addresses high-dimensional mixed-domain (continuous, discrete, categorical, various scalings, etc.) parameter spaces and highly nonlinear or even discontinuous black-box functions (e.g., functions containing jumps) [27].

However, no methods in the literature combine the challenges of global metamodeling with environmental sensing using mobile robots, along with their associated conditions and application-specific costs. In this thesis, we address this gap by considering the scenario of environmental sensing motivated by the ROBDEKON project, where a continuous black-box function (e.g., a concentration distribution) is to be reconstructed on the (low-dimensional and equally scaled) location space based on expensive samples. Additionally, incurred costs are to be taken into account, including the robot's travel costs as well as potentially location-dependent sampling costs.

To our best knowledge, there are no methods that collectively address these challenges. We address this gap by integrating methods from machine learning, adaptive sampling, global multi-objective optimization strategies, and model predictive control. By considering the characteristics of environmental phenomena and the specific requirements for deployment in environmental sensing with mobile robots, we have achieved significant contributions within our field of research listed in the following structure of this thesis:

Chapter 2 – Review on Adaptive Sampling for Global Metamodeling This chapter provides an overview of the literature relevant to this thesis, introducing global metamodeling with a focus on Gaussian process regression, as well as various space-filling and adaptive sampling techniques. Both Gaussian process regression and the sampling methods are central tools in all subsequent chapters. Additionally, a contextualization within topic-specific literature is provided at the beginning of each chapter.

Chapter 3 – Space-filling Designs for Environmental Sensing We consider the scenario where no information about the unknown environmental phenomenon is used for selecting sample points. Drawing on global error measures for the metamodel, we derive three novel sampling criteria, for which we establish both sequential and one-shot variants. In this process, we exploit properties of the design space as well as assumed continuity properties of the phenomenon. Through comprehensive simulation studies, we demonstrate that these novel criteria surpass existing methods reported in the literature.

> **Contribution**: Three novel space-filling sampling criteria (MD, RMSD, mod. maximin) designed for metamodeling of Lipschitz-continuous black-box functions.

Chapter 4 – Weighted Explorative Sampling Criteria for Environmental Sensing We consider the reconstruction of positive-valued environmental phenomena (e.g., concentration distributions). In this context, the model accuracy in regions of high concentration values should be enhanced compared to surrounding regions, to precisely model and delineate peaks. For this, we define the class of explorative sampling criteria (ESC) and illustrate how most existing space-filling criteria can be transformed into such. Building on the ESC, we introduce the class of weighted explorative sampling criteria (WESC) and demonstrate through comprehensive simulations the superiority of two specific WESC over conventional methods in the literature for the described problem scenario.

> **Contribution**: The novel class of WESC, tailored to the application in environmental sensing.

Chapter 5 – Cost-aware Adaptive Sampling for Environmental Sensing We extend the approaches discussed in previous chapters to include practically relevant constraints. Initially, we examine the inclusion of location-dependent sampling costs and the robot's travel costs separately. Subsequently, we combine the methods from both considerations to account for both types of costs in a manner that optimally leverages their strengths. Each scenario is extensively simulated, and the advantages of our novel methods over existing approaches are demonstrated.

> **Contribution**: The novel receding horizon cost-aware adaptive sampling (RHCaAS) algorithm that significantly reduces the robot's travel cost with minimal impact on model accuracy, independent of the criterion used. In addition, the formulation of the cost-aware adaptive sampling criteria PoLC and PoTC based on multi-objective optimization for consideration of the location-dependent (LC) and travel costs (TC) as well as PoLTC for consideration of both cost types. Finally, the integration of PoLTC and RHCaAS to combine advantages of both methods.

The contributions listed above have led to several peer-reviewed publications that can be found in [31–34]. In Appendix A, we discuss the problem addressed in this work within the broader context of recent developments in the field of data-efficient artificial intelligence.

CHAPTER 2

Review on Adaptive Sampling for Global Metamodeling

Metamodeling refers to the process of constructing a metamodel (also known as surrogate model, response surface model [16], or digital twin [19]) of an expensive-to-evaluate (and unknown) black-box function $f : \mathcal{X} \rightarrow \mathbb{R}$ based on a finite number of samples $y_l = f(\underline{x}_l) + \epsilon_l \in \mathbb{R}$ obtained by evaluating the black-box function at the sample points $\underline{x}_l \in \mathcal{X}$, where ϵ_l is a noise term and $\mathcal{X} \subset \mathbb{R}^n$ is the design space of dimension n [15, 16]. In general, the design space may be composed of continuous, discrete, and categorical dimensions [25] and the response of the black-box function may be multi-dimensional. The vast majority of the literature on global metamodeling, however, considers continuous design spaces and single response black-box functions (e.g., [15–18, 21, 35–37]). Throughout this thesis, the design space is assumed to be an n-dimensional nonempty set of real numbers and a single response metamodel is considered. While local metamodels are utilized by an optimization algorithm to determine the global optimum of the black-box function (and are dropped afterwards), global metamodels are to mimic the characteristics of the black-box function and serve as a cheap replacement of the black-box function [21]. The latter is applied in various disciplines of engineering design where metamodels of physical experiments and computer experiments, such as crash tests [35], wind tunnel experiments [17], numerical simulations [18], and hyperparameter optimization of artificial neural networks (ANNs) [25] are built. In case of deterministic black-box functions (e.g., deterministic computer simulations [23]) the noise term ϵ_l disappears. Among modeling techniques like ANNs, support vector machines, polynomial regression, etc. the Gaussian process [38] is most frequently used [16, 22, 39] and most intensively investigated as a metamodel [15, 17] and will also be the model of choice in this thesis.

When relying on the metamodel as a replacement of the black-box function it is crucial to ensure sufficient model accuracy. However, the accuracy of the metamodel decisively depends on the choice of sample points $\underline{x}_l$ [15, 16, 22, 23]. Design of Experiments (DoE) represents a class of methods that generate sample points in the

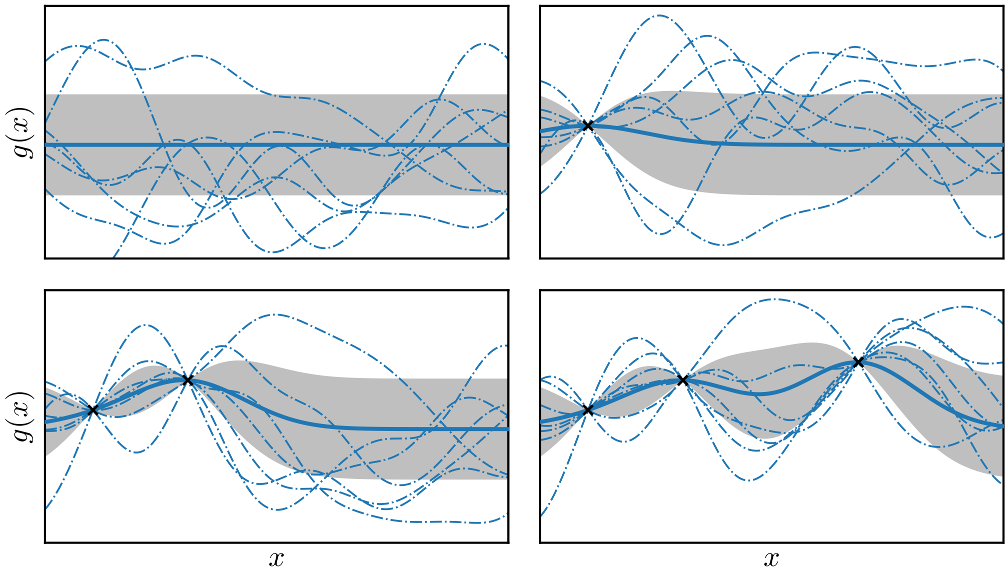

Figure 2.1: This figure shows the process of conditioning a GP on noise free samples $y_l = f(x)$ of an unknown function $f(x)$. The upper left plot shows the prior GP. Then, one sample is added per plot until the GP is conditioned on three samples in the bottom right plot. The solid blue line indicates the mean function of the GP, the gray area has a width of one predictive standard deviation above and below the mean function, respectively, and represents model uncertainty; the dashed blue functions are sample functions of the GP.

design space in such a way that the variations of the black-box function in the design space are reflected as precisely as possible [24]. A basic distinction is made between one-shot designs and sequential designs [15, 21]. One-shot designs like factorial designs, Latin hypercubes, and orthogonal arrays [21, 23, 27] generate a predefined number of space-filling sample points at once (space-filling means that the sample points cover the entire design space as evenly as possible). Sequential designs can again be divided into space-filling and adaptive approaches [15].

Adaptive approaches sequentially determine one or more next sample points based on the already existing samples or metamodel [15]. Note that most space-filling sequential designs only depend on the already existing sample points (e.g., [21]) but do not depend on samples of the black-box function. Hence, just like the one-shot approaches, they can be computed prior to the first evaluation of the black-box function. However, unlike one-shot designs, they allow to easily add more sample points to the existing design if required [23] (e.g., until a termination condition is satisfied).

In the following sections of this chapter, Gaussian process regression is introduced in Sec. 2.1 and the sampling methods for global metamodeling are discussed in Sec. 2.2 including space-filling one-shot designs in Sec. 2.2.1, space-filling sequential sampling in Sec. 2.2.2, and adaptive sampling in Sec. 2.2.3.

2.1 Gaussian Process Regression

Gaussian process regression (GPR) is a non-parametric Bayesian regression method that is applied across various domains, such as machine learning [38], Bayesian optimization [25, 40], control theory [41, 42], reinforcement learning [43–45], and environmental sensing [28, 46]. In geostatistics GPR is also known as Kriging [15, 16, 38] which was initially developed by Krige [47] in 1951 and later extended by Matheron in the context of his work on regionalized variable theory [48]. According to [38], the Gaussian process (GP) is defined as follows:

Definition 2.1.1 (Gaussian process)**.** *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

As stated in [38, 49], a real GP

$$g(\underline{x}) \sim \mathcal{GP}\left(\mu(\underline{x}), k(\underline{x}, \underline{x}')\right) \tag{2.1}$$

is fully specified by its mean function

$$\mu(\underline{x}) = \mathrm{E}\left\{\mathrm{g}(\underline{\mathrm{x}})\right\} \tag{2.2}$$

and its covariance function, also called *kernel*

$$k(\underline{x}, \underline{x}') = \mathrm{E}\left\{(\mathrm{g}(\underline{\mathrm{x}}) - \mu(\underline{\mathrm{x}}))(\mathrm{g}(\underline{\mathrm{x}}') - \mu(\underline{\mathrm{x}}'))\right\}. \tag{2.3}$$

The GP can be considered a distribution over functions whose characteristics are determined by the mean function and kernel. In that sense it is a generalization of a multivariate Gaussian distribution to a stochastic process [38]. According to Def. 2.1.1 we can consider the GP at any finite number M of points $\boldsymbol{X} = [\underline{x}_1, \dots, \underline{x}_M]^\top$ and will obtain a random vector

$$\underline{g} \sim \mathcal{N}\left(\underline{\mu}(\boldsymbol{X}), \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})\right) \tag{2.4}$$

that follows a multivariate Gaussian distribution, where the batch evaluation $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) = [k(\underline{x}_i, \underline{x}_j)]_{i,j=1}^{M} \in \mathbb{R}^{M \times M}$ of the kernel is the covariance matrix of $\underline{g}$ and $\underline{\mu}(\boldsymbol{X}) = [\mu(\underline{x}_1), \dots, \mu(\underline{x}_M)]^\top$ is a batch evaluation of the mean function. This property of the GP is exploited when using it for regression of an unknown function $f(\underline{x})$ based on the dataset $\mathcal{D} = \{(\underline{x}_l, y_l)\}_{l=1}^{m}$ containing m samples $y_l = f(\underline{x}_l) + \epsilon_l \in \mathbb{R}$ taken at sample points x_l, where $\epsilon_l \sim \mathcal{N}(0, \sigma_l^2)$ is a Gaussian noise term and σ_l^2 is the sampling noise variance. First, the joint multivariate Gaussian distribution

$$\begin{bmatrix} \underline{y} \\ \underline{g}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \underline{\mu}(\boldsymbol{X}) \\ \underline{\mu}(\boldsymbol{X}_*) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma} & \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_*) \\ \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) \end{bmatrix}\right) \tag{2.5}$$

of the samples $\underline{y} = [y_1, \dots, y_m]^\top$ and the random vector $\underline{g}_*$ of function values at the evaluation points $\boldsymbol{X}_*$ is defined. Under the assumption of uncorrelated sampling noise

$(\mathrm{E}\left\{\epsilon_{\mathrm{i}}\epsilon_{\mathrm{j}}\right\}_{\mathrm{i}\neq\mathrm{j}} = 0)$ the matrix $\boldsymbol{\Sigma} = \mathrm{diag}([\sigma_1^2, \ldots, \sigma_M^2]^\top)$ is a diagonal matrix containing the sampling noise variances. In practice the sampling noise is often assumed to be stationary $(\sigma_l = \sigma)$ which simplifies the matrix to $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$. Subsequently, the conditional distribution

$$\underline{g}_*|\underline{y}, \boldsymbol{X}, \boldsymbol{X}_* \sim \mathcal{N}\left(\underline{\hat{\mu}}, \hat{\boldsymbol{\Sigma}}\right) \tag{2.6}$$

is formed, where

$$\underline{\hat{\mu}} = \underline{\mu}(\boldsymbol{X}_*) + \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma})^{-1}(\underline{y} - \underline{\mu}(\boldsymbol{X})) \tag{2.7}$$

is the vector of expected function values at the evaluation points and

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) - \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma})^{-1}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_*)$$

is the covariance matrix of the random vector $\underline{g}_*$. Based on Def. 2.1.1, the conditional distribution of $\underline{g}_*$ in (2.6) can again be interpreted as a finite selection of the GP

$$\hat{g}(\underline{x}) \sim \mathcal{GP}\left(\hat{\mu}(\underline{x}), \hat{k}(\underline{x}, \underline{x}')\right) \tag{2.8}$$

that is conditioned on the dataset, where

$$\hat{\mu}(\underline{x}) = \mu(\underline{x}) + \underline{k}(\underline{x}, \boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma})^{-1}(\underline{y} - \underline{\mu}(\boldsymbol{X})) \tag{2.9}$$

is the predictive mean function,

$$\hat{k}(\underline{x}, \underline{x}') = k(\underline{x}, \underline{x}') - \underline{k}(\underline{x}, \boldsymbol{X})(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma})^{-1}\underline{k}(\boldsymbol{X}, \underline{x}') \tag{2.10}$$

is the predictive covariance function (or kernel). $\underline{k}(\underline{x}, \boldsymbol{X})$ is a row vector of covariances between $\underline{x}$ and each point in $\boldsymbol{X}$ (points contained in the dataset) and $\underline{k}(\boldsymbol{X}, \underline{x}')$ is a column vector of covariances between $\underline{x}'$ and each point in $\boldsymbol{X}$. When summarizing the constant terms in (2.9) as $\underline{w}^{GP} = (\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma})^{-1}(\underline{y} - \underline{\mu}(\boldsymbol{X}))$ the predictive mean

$$\hat{\mu}(\underline{x}) = \mu(\underline{x}) + \sum_{i=1}^{m} w_i^{GP} k(\underline{x}_i, \underline{x}) \tag{2.11}$$

can be written as a weighted sum of kernel functions added to the prior mean. It is worth noticing that the properties of a GP let us consider it at a finite number of points, perform inference (compute the posterior distribution), and yield the same result as if we had taken into account all the infinitely many other points [38].

In contrast to many other regression techniques, the GP allows us to directly query the predictive covariance function and thereby obtain an uncertainty measure of our predictions [38]. Most applications are specifically interested in the predictive variance

$$\hat{\sigma}^2(\underline{x}) = \hat{k}(\underline{x}, \underline{x}) \tag{2.12}$$

which is the predictive kernel evaluated at the point of interest (or predictive standard deviation $\hat{\sigma}(\underline{x}) = \sqrt{\hat{k}(\underline{x}, \underline{x})}$, respectively). Fig. 2.1 illustrates how knowledge about the unknown function $f(\underline{x})$ is incorporated into the distribution over functions by conditioning the GP on samples of f.

In Bayesian terms $g(\underline{x})$ is called the *prior* distribution over functions (short prior) and $\hat{g}(\underline{x})$ is called the *posterior* distribution over functions (short posterior). The prior may be chosen to reflect all the knowledge available about the unknown function $f(\underline{x})$. Often little (or nothing at all) is known about the prior mean function which is why it is commonly set to zero everywhere [50]. By defining the correlation between the function values the kernel implies the class of functions that can be represented by the GP. When using the term kernel in the context of a GP we refer to a positive definite kernel which is defined as follows in [49]:

Definition 2.1.2 (Positive definite kernels)**.** *Let $\mathcal{X}$ be a nonempty set. A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite kernel, if for any $n \in \mathbb{N}$, $w_1, \ldots, w_n \in \mathbb{R}$ and $x_1, \ldots, x_n \in \mathcal{X}$,*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j k(\underline{x}_i, \underline{x}_j) \geq 0. \tag{2.13}$$

More loosely speaking the above definition says that the covariance matrix constructed by evaluating $k(\cdot, \cdot)$ for any finite number of points in $\mathcal{X}$ must be positive semidefinite in order for $k(\cdot, \cdot)$ to be a positive definite kernel [49]. Many kernels can be found in literature, such as the linear, polynomial, exponential, periodic, squared exponential, and Mathern kernel [38]. One of the most frequently used kernels is the *squared exponential (SE)* (or radial basis function (RBF)) kernel [38, 49]

$$k^{\mathrm{SE}}(\underline{x}, \underline{x}') = \exp\left(-\frac{1}{2}(\underline{x} - \underline{x}')^\top \boldsymbol{\Theta}^{-1} (\underline{x} - \underline{x}')\right) \tag{2.14}$$

with hyperparameters $\boldsymbol{\Theta} = \mathrm{diag}([\theta_1^2, \ldots, \theta_n^2])$. Any sum or product of a kernel is again a valid kernel [38]. This way the properties of multiple kernels can be combined to construct a kernel that suits the application (details on combining kernels are well described, e.g., in [50]). E.g., the prior uncertainty about the mean function can be expressed by multiplying the SE kernel with the constant kernel $k^{\mathrm{const}}(\underline{x}) = \sigma_f^2$, where σ_f is a hyperparameter.

In the first sentence of this section, the GP was introduced as a non-parametric regression method. This is true for a given kernel, since then the posterior distribution in (2.8) and hence the predictive mean in (2.9) as well as predictive covariance in (2.10) can be computed based on the kernel and dataset. However, the choice of the kernel and even the choice of hyperparameters for a selected kernel have a great impact on the regression result. In Fig. 2.2 the effect of different choices of the hyperparameter on the SE kernel and samples of the corresponding GP prior is shown.

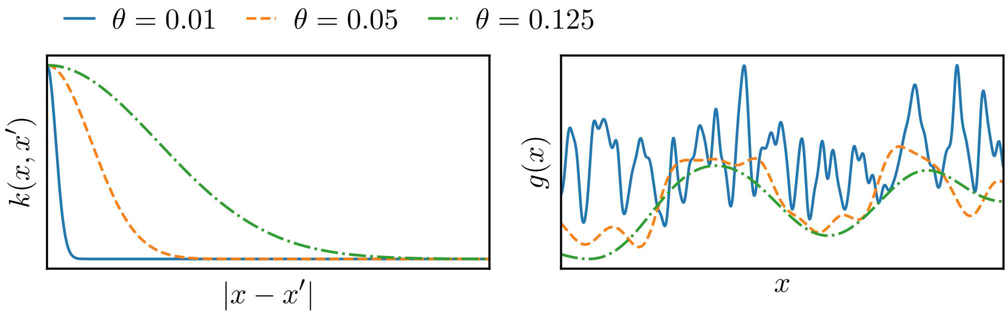

Figure 2.2: In this figure, a GP with an SE kernel $k^{\mathrm{SE}}(x, x') = \exp(-\frac{(x-x')^2}{2\theta^2})$ is considered for different hyperparameters θ to illustrate how the characteristics of the GP depend on the choice of hyperparameter. On the left, the kernel is shown over the Euclidian distance $|x - x'|$ between the evaluation points and on the right are sample functions from the prior GP.

The process of selecting the kernel and its hyperparameters is known as *model selection*. For that, a measure is needed to evaluate how well the model fits the data. In case of a GP we can compute the *marginal likelihood*

$$p(\underline{y}|\boldsymbol{X}) = \int \underbrace{p(\underline{y}|\underline{g}, \boldsymbol{X})}_{\text{likelihood}} \underbrace{p(\underline{g}|\boldsymbol{X})}_{\text{prior}} \, d\underline{g} \tag{2.15}$$

by integrating the likelihood of the samples weighted by the prior over all possible function values [38] (which can be thought of as a weighted average over all possible function values of the GP at the sample points in the dataset). The marginal likelihood is again a multivariate Gaussian distribution that reflects the probability of the observed samples in the dataset given a kernel and its hyperparameters. Hence, it can be used for determination of the optimal model through maximization w.r.t the kernel and hyperparameters. Instead of maximizing the marginal likelihood we can instead maximize the *log marginal likelihood* [38]

$$\begin{aligned} \log p(\underline{y}|\boldsymbol{X}) = & \underbrace{-\frac{1}{2}(\underline{y} - \underline{\mu}(\boldsymbol{X}))^\top (\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma})^{-1} (\underline{y} - \underline{\mu}(\boldsymbol{X}))}_{\text{model to data fit}} \\ & \underbrace{- \frac{1}{2} \log \det(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\Sigma})}_{\text{model complexity}} \\ & \underbrace{- \frac{m}{2} \log 2\pi}_{\text{constant}} \end{aligned} \tag{2.16}$$

which in practice is usually computed by means of Cholesky factorization to improve numerical stability [38, 51]. The first term in (2.16) measures how well the model fits to the dataset, the second term measures the complexity of the model and provides regularization, and the third term is a constant (thus not relevant for optimization). As mentioned above, the prior mean function is often set to zero if there is no

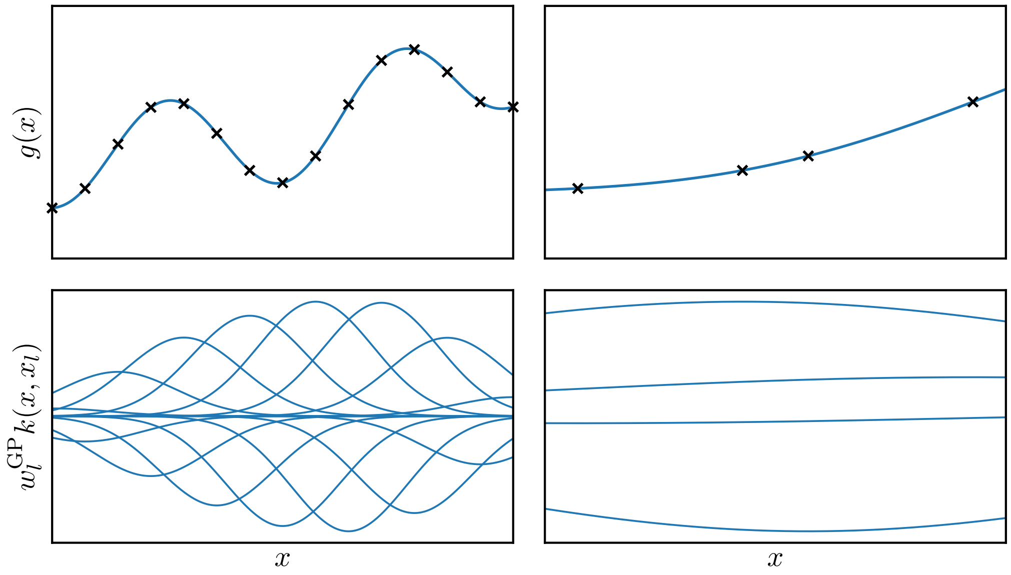

Figure 2.3: A GP with a zero-mean prior and an SE kernel is fit to two sample sets of the same underlying function. The upper row shows the predictive mean (solid blue line) and standard deviation (gray shaded area) of the GP after fitting the GP to 15 samples on the left and to four samples on the right (black crosses) by maximizing the log marginal likelihood in (2.16) w.r.t. the hyperparameter. In the bottom row, each summand of the sum in (2.11) (weighted covariances between the sample points x_l and the evaluation points x) is plotted separately. The sum of the curves shown adds up to the respective predictive mean function shown above. Note that the predictive standard deviation is almost not recognizable since the (large) optimized hyperparameters cause the model to have a strong confidence (low predictive variance) in its predictions between the sample points.

knowledge about the mean function. However, this leads to the predictive mean approach the prior mean in regions where no samples are available. If a global mean or a trend is expected, the mean function can also be modeled as a parametric function whose parameters can be estimated along with the hyperparameters by maximizing the log marginal likelihood [38]. In the special case that the mean function can be represented as a sum of weighted basis functions (such as a polynomial) with Gaussian distributed weights, then the weights can be integrated out which yields another GP that has the basis functions incorporated into its equations [38].

The optimization of the hyperparameters affects our prior assumptions about the characteristics of the unknown function. For an SE kernel, e.g., increasing the hyperparameter causes the regression function to have smaller maximum rates of change. In addition, the prediction variance is significantly dependent on the hyperparameters. Especially if few samples are available, this can lead to an underestimation of the model uncertainty. Fig. 2.3 shows the regression result of the same underlying function based on two different sample sets. Less samples do not cause the model to be less confident in regions where there are no samples available, since the hyperpa-

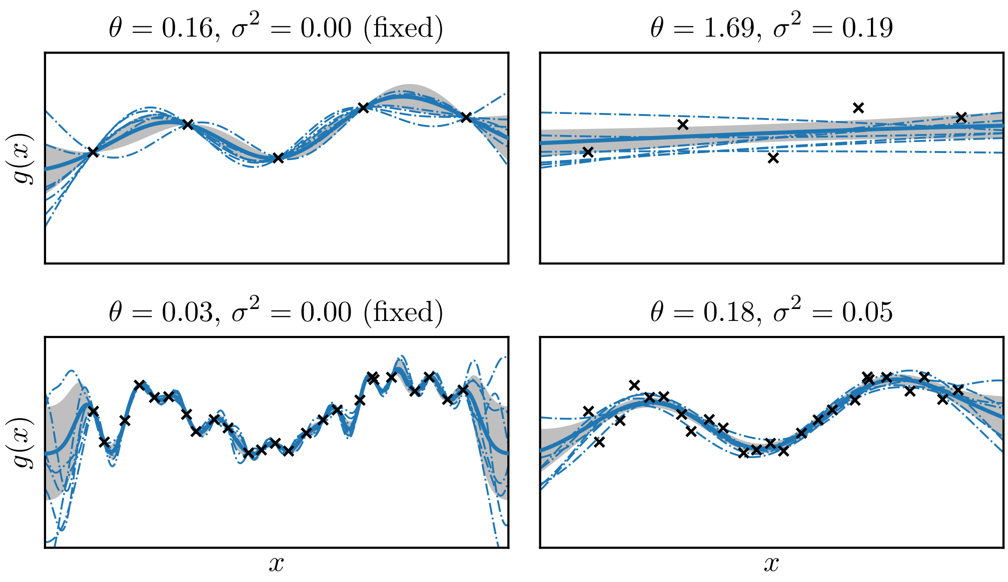

Figure 2.4: A GP with a zero-mean prior and an SE kernel is fit to two sets of samples of the same underlying function. Five noise-free samples were used in the top row and 25 noisy samples ($\sigma^2 = 0.25^2$) in the bottom row. The hyperparameter θ of the SE kernel was optimized by maximizing the log marginal likelihood in (2.16) w.r.t. the hyperparameter. In the left column the sample noise variance in the GP equations ($\mathbf{\Sigma} = \sigma^2 \boldsymbol{I}$) was set to zero and in the right column it was optimized along with the hyperparameter. The solid blue line indicates the mean function of the GP, the gray area has a width of one predictive standard deviation above and below the mean function, respectively, and represents model uncertainty; the dashed blue functions are sample functions of the GP.

rameter (indicated by the kernel functions) after optimization takes a larger value. This implies a low rate of change as prior knowledge about the unknown function which then leads to the predictive variance increase slowly with distance to the existing samples. Hence, the optimum of the log marginal likelihood may, dependent on the application, lead to unrealistic prior assumptions about the unknown function (which has to be considered or taken care of by the user, respectively).

Note that if no sampling noise is considered in the equations the GP performs interpolation of the samples. Since adding to the diagonal of the covariance matrix improves numerical stability of matrix inversion, it is common practice to set the noise level to a small number (e.g., $\sigma^2 = 10^{-5}$) even if the samples should be interpolated. If the sampling noise is unknown but should be taken into account, it can be optimized along with the hyperparameters. By adding sampling noise ($\mathbf{\Sigma}$) to the equations the predictive mean function no longer has to run exactly through the samples. This effect is often exploited in practice to achieve a regularization of the Gaussian process by manual adjustment of the noise variance.

Fig. 2.4 shows how optimizing the sampling noise variance σ^2 along with the hyperparameter impacts the optimization result for the hyperparameter. For a large number

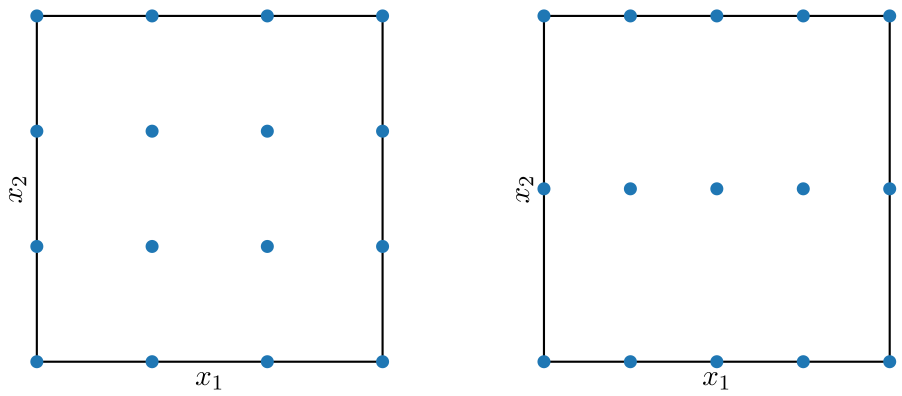

Figure 2.5: Two-dimensional RGS with four levels per dimension on the left and three or five levels per dimension on the right.

of noisy samples, the predictive mean function captures the course of the underlying function, and the estimated noise variance is close to the actual one. However, this automatic regularization leads to the GP averaging between the sample points. Especially for a small number of samples, this may lead to the GP interpreting true variation in the data as noise. In practice this can be controlled by either manually providing the noise variance or by constraining the optimization to reasonable bounds for the noise variance.

For more details on the derivation of the GP, associated formulas, numerically stable implementations, and further discussion on GP regression, we refer to [38, 49, 51].

2.2 Sampling Methods

The distribution of sample points on the design space has a significant impact on the accuracy of the metamodel [15, 16, 22, 23]. Accordingly, a variety of sampling methods for global metamodeling can be found in the literature. In this section, the sampling methods relevant to this thesis from the context of global metamodeling are presented. First, the space-filling sampling methods are discussed in Sec. 2.2.1 and Sec. 2.2.2. Then, the general adaptive sampling procedure for global metamodeling and the adaptive sampling criteria are introduced in Sec. 2.2.3.

2.2.1 Space-filling One-shot Designs

One-shot designs are sampling methods that generate a predefined number of samples at once in the design space. They include a wide variety of methods that can be divided into two classes. On the one hand, the points can be constructed according to a predefined scheme, such as regular grid sampling, Hammersley sampling, and Latin hypercube sampling. On the other hand, the points can be optimized w.r.t.

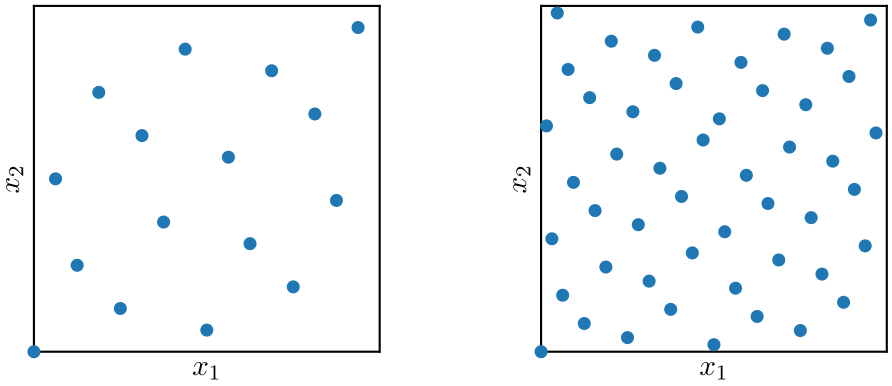

Figure 2.6: Two space-filling designs (16 points on the left and 49 points on the right) created with Hammersley sampling on a two-dimensional design space.

a criterion using global optimization techniques. In the following, the space-filling one-shot designs relevant to this thesis are presented.

Regular Grid Sampling *Regular grid sampling (RGS)* (also referred to as factorial experimental design) divides each dimension of the design space into levels of equal spacing. Each dimension can have a different number of levels. In this way, space-filling sample points can be efficiently generated. Fig. 2.5 shows examples of RGS for different levels per dimension.

Hammersley Sampling *Hammersley sampling* uses the Hammersley sequence (or set) to compute quasi-random sample points. The Hammersley sequence belongs to the class of low-discrepancy sequences [52, 53]. Low-discrepancy sequences are deterministic mathematical formulas for computation of sample points which are provably more uniformly distributed than random numbers drawn from a uniform distribution [54] and are vastly applied in computer graphics, numerical integration, and global optimization. In addition, most low-discrepancy sequences show Latin hypercube properties [54]. For a given number of sample points in an (n-dimensional) unit cube, discrepancy is defined for any subinterval within the unit cube as the difference between the proportion of sample points that fall into the subinterval and the volume of the subinterval. Formal definitions of discrepancy and low-discrepancy sequences can be found e.g., in [52, 55–57]. Note that the unit cube is used in the definition without loss of generality (and can be formulated for any n-dimensional volume). The lower the maximum discrepancy of points within a given volume, the more uniformly the points are distributed. A set of m two-dimensional Hammersley sample points

$$\left\{\left(\mathcal{H}_2(l), \frac{l}{m}\right)\right\}_{l=1}^{m} \tag{2.17}$$

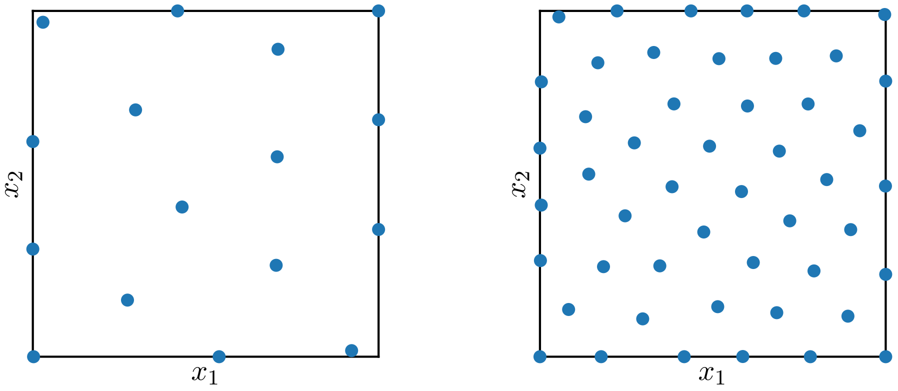

Figure 2.7: Two optimized maximin designs (16 points on the left and 49 points on the right) on a two-dimensional design space. The designs were optimized as described in Sec. 3.2.1.

in a unit square can be defined by elements of the van der Corput sequence $\mathcal{H}_b(l)$ in base b and fractions $\frac{l}{m}$ of the interval $[0, 1]$ [53]. $\mathcal{H}_b(l)$ can be efficiently computed by means of bit-wise operations. For that the integer representation of l in base b is reversed as the fractional part of a fixed-point number [53]. Refer to [55] for a formal definition of the van der Corput sequence and to [58] for the definition of the n-dimensional Hammersley sequence. Fig. 2.6 shows examples of Hammersley sampling for different numbers m of sample points on a two-dimensional design space.

Maximin Sampling The *maximin* criterion is one of the most popular space-filling sampling criteria due to its numerical robustness and space-filling properties [21]. For a given set $X = \{\underline{x}_1, \ldots, \underline{x}_m\} \subset \mathcal{X}$ of sample points the maximin criterion

$$\phi^{\text{maximin}}(X) = \min_{\substack{\underline{x}_i, \underline{x}_j \in X \\ i \neq j}} d(\underline{x}_i, \underline{x}_j) \tag{2.18}$$

computes the minimum distance $d(\underline{x}_i, \underline{x}_j)$ between any two sample points $\underline{x}_i, \underline{x}_j \in X$ with $i \neq j$. Mostly (and throughout in this thesis) the l_2-norm (Euclidian distance) $d(\underline{x}_i, \underline{x}_j) = ||\underline{x}_i - \underline{x}_j||$ is used to calculate the distance between the points. In general, however, the maximin criterion also allows the use of other distance measures. As its name suggests, the optimal maximin design

$$X^* = \arg\max_{X \subset \mathcal{X}} \min_{\substack{\underline{x}_i, \underline{x}_j \in X \\ i \neq j}} d(\underline{x}_i, \underline{x}_j) = \arg\max_{X \subset \mathcal{X}} \phi^{\text{maximin}}(X) \tag{2.19}$$

is obtained by (global) maximization of the maximin criterion w.r.t. the sample points X on the design space $\mathcal{X}$. Two maximin designs on a two-dimensional design space with a different number of sample points are shown in Fig. 2.7.

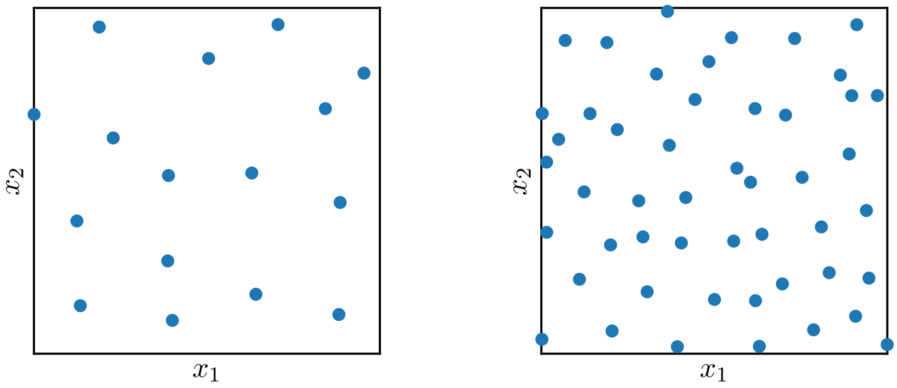

Figure 2.8: Two (suboptimal) minimax designs (16 points on the left and 49 points on the right) on a two-dimensional design space. Despite extensive optimization efforts (as described in Sec. 3.2.1), no global optimum was found.

Minimax Sampling The *minimax* criterion is closely related to the maximin criterion. In fact, given a set $X = \{\underline{x}_1, \dots, \underline{x}_m\} \subset \mathcal{X}$ of sample points, the minimax criterion

$$\phi^{\text{minimax}}(X) = \max_{\underline{x} \in \mathcal{X}} \min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i|| \tag{2.20}$$

is defined as the maximum distance that a point in the design space can have to its nearest given sample point in X. A set of points

$$X^* = \arg \min_{X \subset \mathcal{X}} \max_{\underline{x} \in \mathcal{X}} \min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i|| = \arg \min_{X \subset \mathcal{X}} \phi^{\text{minimax}}(X) \tag{2.21}$$

that minimize the minimax criterion form an optimal minimax design. When considering a Lipschitz-continuous function f that is to be approximated on the design space by a model trained on samples of f taken at the sample points X, then optimizing (2.21) is equal to minimizing the maximum Lipschitz error [59, 60] (since the maximum distance an evaluation point can have to its closest sample point is minimized). Hence the minimax design leaves smaller gaps between the sample points on the design space [60] and places less samples on the boundary of the design space than the maximin design [59]. Fig. 2.8 shows two minimax designs on a two-dimensional design space with a different number of sample points. Note that finding an optimal maximin design (even on a two-dimensional design space) is challenging and practically intractable for higher dimensions or a large number of points. However, finding an optimal minimax design is an even harder problem [59].

Centroidal Voronoi Tessellation Given a set of unique (discrete) points $X = \{\underline{x}_i\}_{i=1}^m \subset \mathbb{R}^n$ (the generators) in an n-dimensional Euclidian space, the Voronoi region $R_i = \{\underline{x} \in \mathbb{R}^n : \ ||\underline{x} - \underline{x}_i|| \leq ||\underline{x} - \underline{x}_j|| \ \forall j \in \{1, \dots, m\} \backslash i\}$ (also referred to as Voronoi cell) associated with point $\underline{x}_i$ contains all points in $\mathbb{R}^n$ that are closer (by a distance measure, here the Euclidian distance) to $\underline{x}_i$ than to any other point

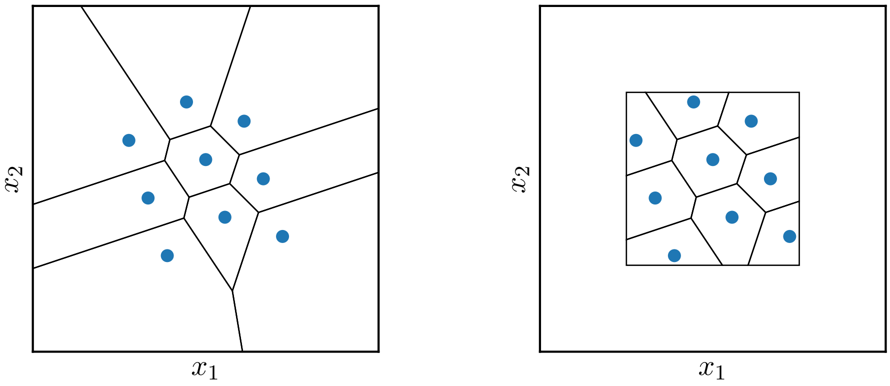

Figure 2.9: A Voronoi tessellation (left) and its clipped variant (right) with 9 points.

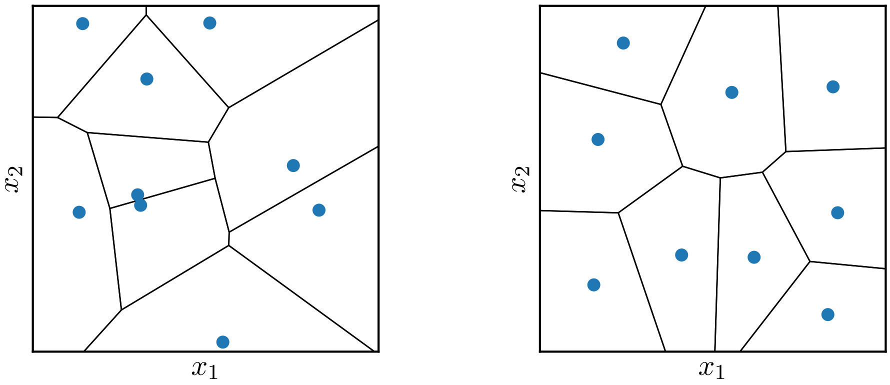

Figure 2.10: Voronoi tessellation of 9 random sample points (left) and a CVT of size 9 (right).

in X [23, 61]. The *Voronoi tessellation* (or Voronoi diagram) is the set of Voronoi regions $\{\underline{x}_i\}_{i=1}^m$ corresponding to the generators X. In practice, usually only an open bounded subset $\mathcal{X} \subset \mathbb{R}^n$ of $\mathbb{R}^n$ is considered. Hence, all m Voronoi regions $R_i^{\text{clipped}} = \{\underline{x} \in \mathcal{X} : \; ||\underline{x} - \underline{x}_i|| \leq ||\underline{x} - \underline{x}_j|| \; \forall j \in \{1, \ldots, m\} \backslash i\}$ of the clipped Voronoi diagram have a finite volume. Fig. 2.9 shows a Voronoi tessellation before and after it was clipped to a bounding box.

A *centroidal Voronoi tessellation (CVT)* is a special variant of a Voronoi tessellation where for each Voronoi region R_i the corresponding generator point $\underline{x}_i$ is also the mass centroid $\underline{z}_i$ of the Voronoi region [62, 63], making it a space-filling design. The mass centroid for the Voronoi region R_i is defined as

$$\underline{z}_i = \frac{\int_{R_i} \underline{x}\rho(\underline{x})d\underline{x}}{\int_{R_i} \underline{x}d\underline{x}} \, , \tag{2.22}$$

where $\rho(\underline{x}) \geq 0$ is a probability density function that is defined on R_i [63, 64]. Many algorithms exist for efficient computation of the CVT for a given probability density function, such as quasi-Newton-based methods, conjugate gradient-based methods, Lloyd's method, and MacQueen's algorithm [61, 62]. Note that a CVT for

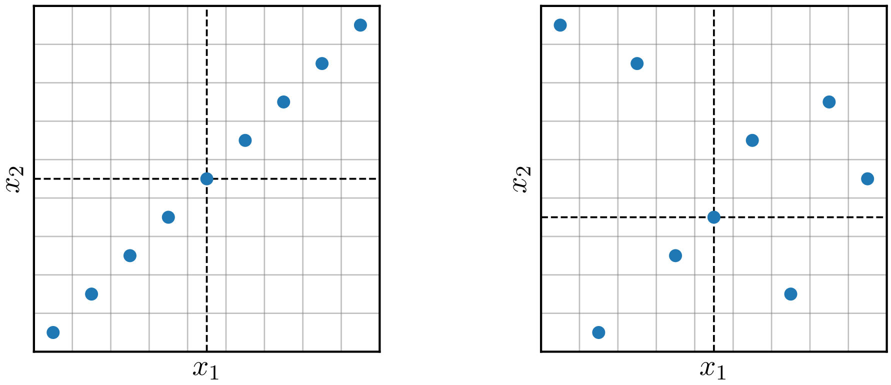

Figure 2.11: Two Latin hypercube designs with 9 samples (correlated design on the left and a random design on the right) on a two-dimensional design space. The gray grid marks the discretization levels, and the dashed black cross illustrates that there is exactly one sample point located in each row and column (or generally axis-aligned hyperplane).

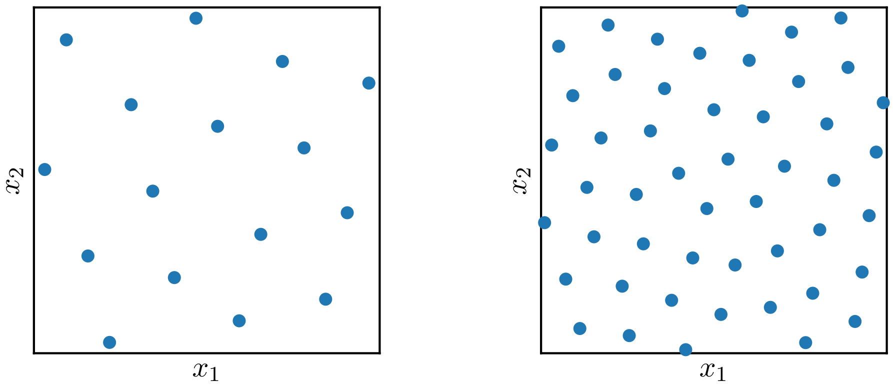

Figure 2.12: OLHS with 16 samples on the left and 49 samples on the right on a two-dimensional design space. The enhanced stochastic evolutionary (ESE) algorithm [65] was used along with the maximin criterion to compute the optimal LHs, as described in Sec. 3.2.1.

a given probability density function is in general not unique [64]. Examples for a (non-centroidal) Voronoi tessellation and a CVT are shown in Fig. 2.10.

Latin Hypercube Sampling *Latin hypercube sampling (LHS)* is a popular method for generating sample points that form a Latin hypercube. A *Latin hypercube (LH)* is constructed by discretizing each dimension of an n-dimensional design space into m equidistant levels and placing m samples on the resulting n-dimensional grid such that there is exactly one sample point contained in each axis-aligned hyperplane [21] as shown in Fig. 2.11. Hence, the LH is composed of points that are unique in each dimension, making it a non-collapsing design (which implies good projective properties) [15, 21, 27]. When projecting an n-dimensional LH onto an $(n-1)$-dimensional space along one axis, no points ever coincide [21, 27].

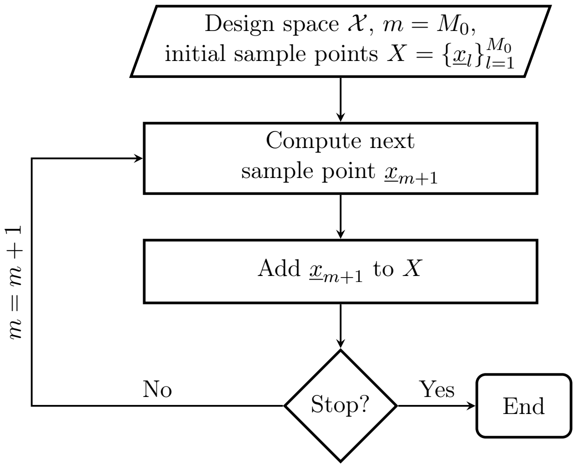
Design space $\mathcal{X}$, $m = M_0$,
initial sample points $X = \{\underline{x}_l\}_{l=1}^{M_0}$
Compute next sample point $\underline{x}_{m+1}$
Add $\underline{x}_{m+1}$ to X
Stop?
No
Yes
End
$m = m + 1$

Figure 2.13: General procedure of sequential sampling for global metamodeling.

However, LHs are not guaranteed to have good space-filling properties, especially when they are randomly generated [21, 27]. *Optimal Latin hypercube sampling (OLHS)* generates an optimal LH w.r.t. a space-filling criterion (such as e.g., the maximin criterion in 2.2.1) by means of global optimization techniques. Even for a small number of samples OLHS guarantees a good coverage of the design space [21, 66]. Fig. 2.12 shows optimal LHs for different numbers of samples. Since it is computationally demanding to generate optimal LHs many near-optimal solutions have been researched [66]. It is worth noticing that LHs do not degenerate for a large number n of dimensions which means that they keep their properties for large n and do not suffer from the curse of dimensionality (as many other methods do) [18].

2.2.2 Space-filling Sequential Sampling

One-shot designs like the ones in Sec. 2.2.1 provide an optimal distribution of sample points on the design space according to some sampling criterion (e.g., the maximin criterion) or deterministic procedure (e.g., Hammersley sampling). However, the optimization required can be complex and computationally demanding [27, 59]. In addition, one-shot designs need the total number M of sample points to be known a priori which can result in gathering too many or too few samples (oversampling / undersampling) [23]. In contrast, sequential sampling iteratively determines one (or multiple) next sample point(s) taking into account the existing sample points $X = \{\underline{x}_l\}_{l=1}^{m}$. Hence, just like the one-shot approaches, they can be computed prior to the first evaluation of the black-box function. However, unlike one-shot designs, they allow to easily add more sample points to the existing design if required [23] (e.g., until a termination condition is satisfied) as illustrated in Fig. 2.13. To start the sequential sampling procedure, at least M_0 initial sample points are required

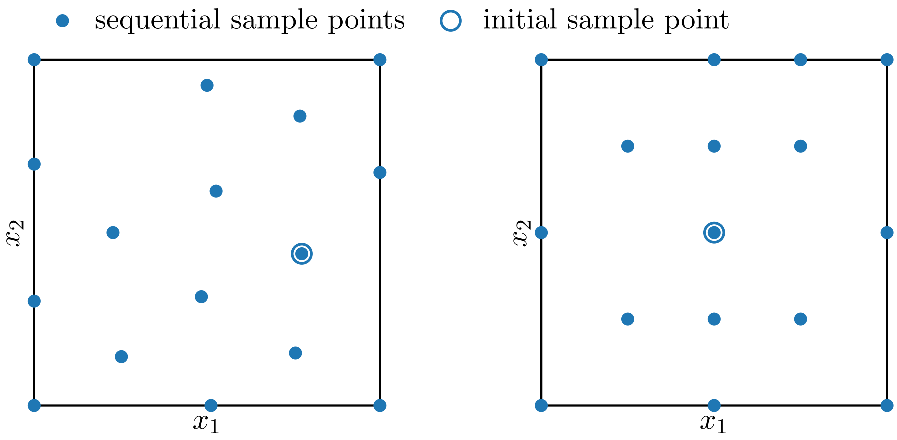

Figure 2.14: Two sequential maximin designs with 16 sample points initialized with different initial sample points (surrounded by circle). This illustrates the impact of the choice of the initial sample point on the resulting sequential design.

depending on the method used. The methods considered in this thesis (except of random sampling) require at least one initial sample point ($M_0 \geq 1$). In addition, one next sample point is determined in each iteration of the sequential sampling algorithm. Fig. 2.14 shows two sequential designs that were generated using the same space-filling sequential criterion but varying initial sample points. This illustrates how significantly the choice of the initial sample point affects the resulting design. Note, however, that there is no general way for selecting the initial sample point(s). The following sections present the space-filling sequential sampling methods relevant to this work.

Random Sampling *Random sampling* randomly selects sample points on the design space according to a probability distribution. In the context of space-filling designs samples are drawn from the uniform distribution. Hence, in contrast to other sequential space-filling sampling methods, the choice of the next sample point does not depend on the existing sample points [21]. As shown in Fig. 2.15, this leads to poor space-filling properties (uneven coverage of the design space). Especially for a small number of samples, several samples may be located close to each other while leaving big gaps in the design space [21, 23].

Sequential Maximin Sampling *Sequential maximin sampling* uses a sequential formulation of the maximin criterion in Sec. 2.2.1. Instead of considering the distances between all existing sample points, the sequential maximin criterion

$$\varphi^{\text{maximin}}(\underline{x}) = \min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i|| \tag{2.23}$$

computes the minimum distance of the point $\underline{x}$ to its nearest existing sample point.

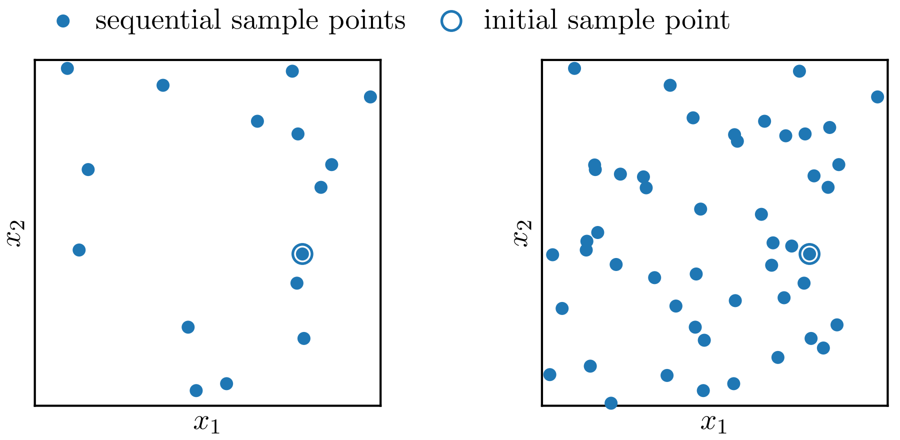

Figure 2.15: Sequential random sampling of 16 samples on the left and 49 samples on the right.

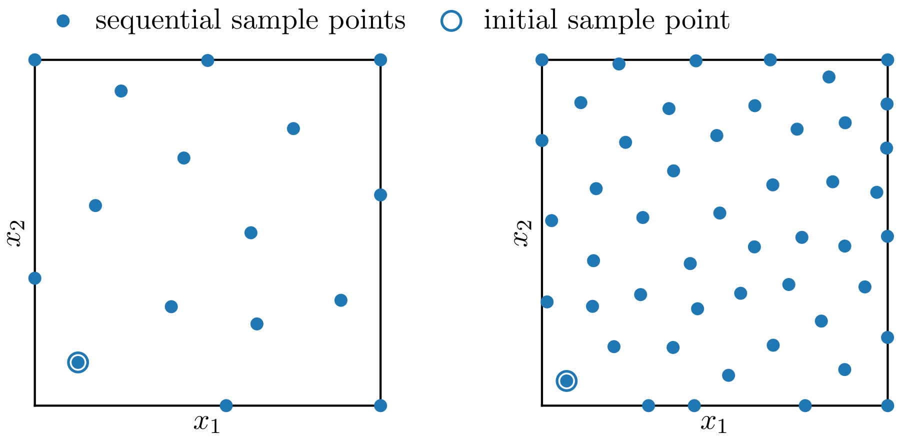

Figure 2.16: Sequential maximin designs of 16 samples on the left and 49 samples on the right.

The point on the design space that maximizes (2.23) is in each iteration selected as the next sample point

$$\underline{x}_{m+1} = \arg\max_{\underline{x}\in\mathcal{X}} \min_{\underline{x}_i\in X} ||\underline{x} - \underline{x}_i|| = \arg\max_{\underline{x}\in\mathcal{X}} \varphi^{\text{maximin}}(\underline{x}) \tag{2.24}$$

and added to the existing samples. Examples for sequential maximin designs are shown in Fig. 2.16.

Voronoi Sampling In *Voronoi sampling (VS)* a Voronoi tessellation (as defined in 2.2.1) is constructed on the existing sample points $X = \{\underline{x}_l\}_{l=1}^{m}$. The hypervolume $\underline{v}[j]$ of the jth Voronoi region (the Voronoi region associated with the jth existing sample point) is used as cell selection criterion

$$S^{\text{VS}}(j) = \underline{v}[j] \ , \tag{2.25}$$

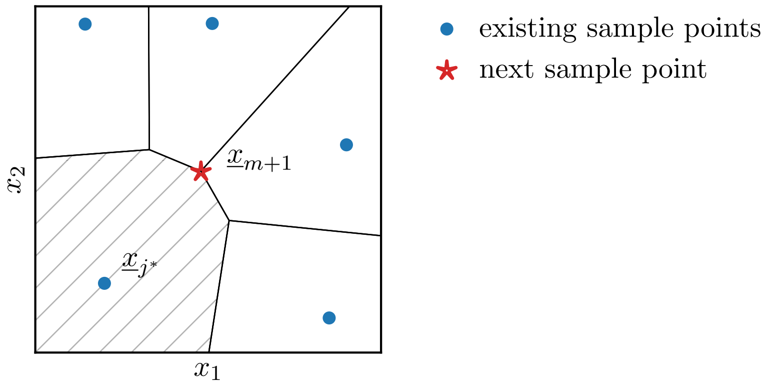

Figure 2.17: The polygons (black lines) represent the Voronoi cells constructed based on the existing sample points. The selected cell (the cell with the largest hypervolume) is hatched. The vertex of the selected cell that is farthest from the corresponding existing sample point $\underline{x}_{j^*}$ is selected as the next sample point $\underline{x}_{m+1}$.

where $\underline{v}$ are the hypervolumes of the Voronoi regions and $j \in \{1, \dots, m\}$ is the index of the Voronoi region [21]. Taking the maximum of the cell selection criterion in (2.25) yields the index

$$j^* = \arg\max_j S^{\mathrm{VS}}(j) \tag{2.26}$$

of the Voronoi region that is largest by hypervolume. As illustrated in Fig. 2.17, the vertex of the Voronoi region R_{j^*} that is farthest from the existing sample point $\underline{x}_{j^*}$ associated with R_{j^*} is then selected as the next sample point $\underline{x}_{m+1}$ [21]. Fig. 2.18 shows two space-filling designs with different numbers of samples generated by Voronoi sampling.

2.2.3 Adaptive Sampling

Adaptive sampling methods (also called sequential designs [23] or active learning [21]) sequentially determine one or multiple next sample points based on the existing samples or metamodel. In contrast to space-filling sequential sampling methods that try to spread samples as evenly as possible over the entire design space, adaptive sampling bases its suggestion for the next sample point on the existing knowledge about the black-box function. In this way, the information per sample about the course of the black-box function on the design space should be maximized [21].

Since initially no information about the black-box function is available, a space-filling initial design $\mathcal{D}_0 = \{\underline{x}_l, y_l\}_{l=1}^{M_0}$ is first created to cover the design space $\mathcal{X}$ as evenly as possible [15, 27]. This is done by first determining M_0 initial sample points using a space-filling sampling scheme (e.g., one described in Sec. 2.2.1 or Sec. 2.2.2) at which the black-box function f is evaluated. The metamodel $\hat{f}$ is trained for the first time on the complete initial design $\mathcal{D}_0$. It has been shown that the more evenly the

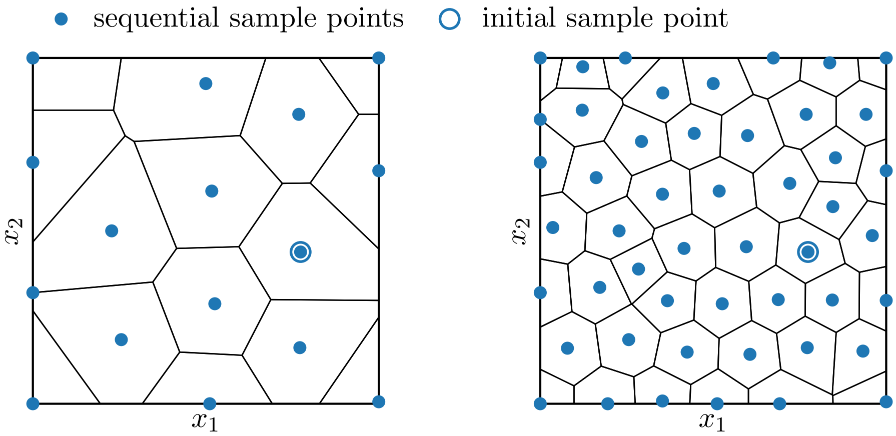

Figure 2.18: Two space-filling designs with 16 samples on the left and 49 samples on the right, generated by sequential Voronoi sampling. The polygons (black lines) represent the Voronoi cells.

points are distributed, the lower the expected model error becomes [16, 23]. Thus, the better the initial design gathers information from all regions of the design space, the better the adaptive algorithm can subsequently place samples at points of interest w.r.t. the criterion used, thereby reducing the uncertainty of the model.

The initial design and metamodel are used to warm-start the adaptive sampling algorithm. Then, until a termination criterion (e.g., maximum number of iterations, estimated model accuracy, available budget) is met, the following steps are repeated in each iteration i of the algorithm:

- determine the next sample point $\underline{x}_{m+1}$ using some criterion
- evaluate the black-box function f at $\underline{x}_{m+1}$ and obtain y_{m+1}
- add the new data to design $\mathcal{D}_{i+1} = \mathcal{D}_i \cup \{\underline{x}_{m+1}, y_{m+1}\}$
- train the metamodel $\hat{f}$ on the current data set $\mathcal{D}_{i+1}$.

Fig. 2.19 shows the general procedure of adaptive sampling for global metamodeling.

The resulting adaptive design is significantly dependent on the sampling criterion used, since it determines the choice of the next sample point in each iteration of the algorithm. A sampling criterion $\varphi : \mathcal{X} \to \mathbb{R}$ (also called acquisition function) maps the design space $\mathcal{X}$ to the real line and indicates how interesting a point is in the design space. Consequently, the point $\underline{x}_{m+1} = \arg\max_{\underline{x} \in \mathcal{X}} \varphi(\underline{x})$ on the design space that maximizes the sampling criterion is selected as the next sample point. Fig. 2.20 illustrates how the next sample point is selected by optimization of a sampling criterion on the design space.

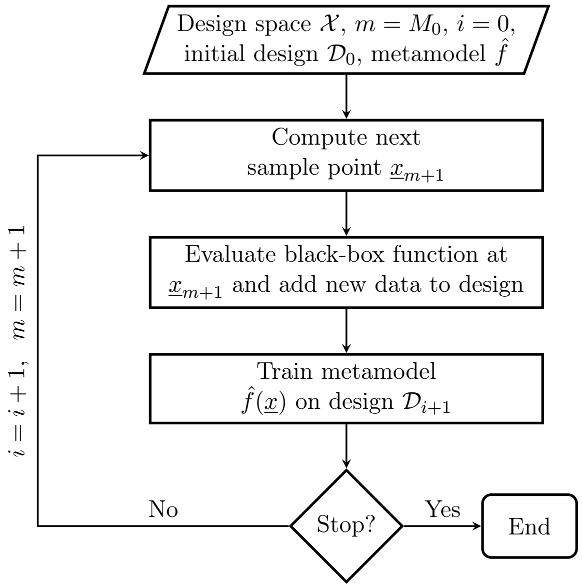
Design space $\mathcal{X}$, $m = M_0$, $i = 0$, initial design $\mathcal{D}_0$, metamodel $\hat{f}$
Compute next sample point $\underline{x}_{m+1}$
Evaluate black-box function at $\underline{x}_{m+1}$ and add new data to design
Train metamodel $\hat{f}(\underline{x})$ on design $\mathcal{D}_{i+1}$
Stop?
Yes
End
No
$i = i + 1,\ m = m + 1$

Figure 2.19: General procedure of adaptive sampling for global metamodeling.

Adaptive sampling methods need to perform a trade-off between exploration and exploitation [27, 36], where exploitation increases the local model accuracy and exploration ensures that no relevant regions of the design space are omitted. Numerous distance-, variance-, gradient-, and cross-validation-based exploration and exploitation strategies have been researched [15]. However, there is no general solution to how the trade-off between exploration and exploitation is realized. Most adaptive sampling criteria are mathematically formulated to inherently execute a trade-off between exploration and exploitation. One way to do this is to combine an exploratory term and an exploitative term into one criterion in the form of a weighted sum. Some more advanced methods formulate the weights as a function of the iteration of the algorithm, thus allowing a smooth transition from exploration to exploitation as the number of samples increases or switching between exploration and exploitation at certain intervals [15].

The criteria can also be differentiated according to the type of exploitation. In global metamodeling, the black-box function should be reconstructed on the entire design space, while in global optimization the optimum of the black-box function is to be localized as precisely as possible [15, 25]. Adaptive sampling methods for global optimization of black-box functions, in which a GP is used as a metamodel, are known as *Bayesian optimization (BO)* [25, 26]. The most popular sampling criteria in the context of BO are probability of improvement (PI), expected improvement (EI), upper confidence bound (UCB), and extensions of these, such as expected improvement global fit (EIGF) [25, 26, 40, 67]. Besides BO, GPs are by far the most commonly used metamodel in the field of global metamodeling (including geostatistics and environmental sensing) [15, 17, 68] and are also used as a metamodel in this

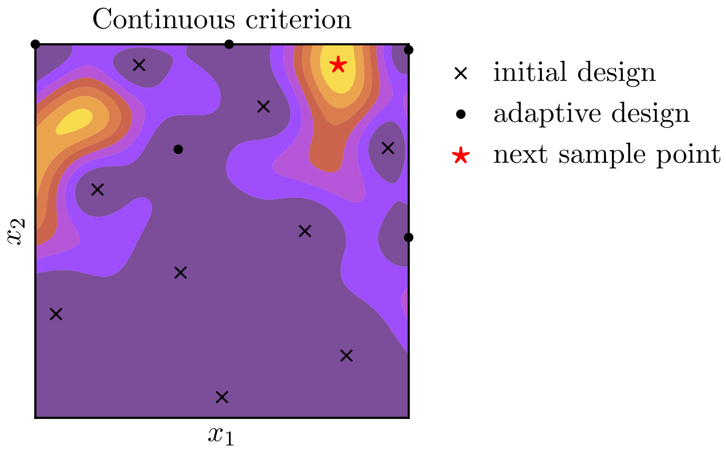

Figure 2.20: Initial and adaptive design shown on a sampling criterion $\varphi(\underline{x})$. As indicated in the figure, the maximizer of the criterion is selected as next sample point.

thesis. The following paragraphs introduce the adaptive sampling criteria relevant to this thesis.

Maximum Variance and Maximum Mean Squared Error Criteria In many applications, sample points are selected such that they maximize the information collected [16]. In the case of GPs, this can be achieved by the *maximum variance (MV)* criterion

$$\varphi^{\mathrm{MV}}(\underline{x}) = \hat{\sigma}^2(\underline{x}) \ , \tag{2.27}$$

which corresponds to the predictive variance $\hat{\sigma}^2$ of the GP and thus also to the maximum entropy criterion [24]. The predictive standard deviation $\hat{\sigma}$ of a GP is also known as the mean squared error (MSE) [16, 69]. Hence the *maximum mean squared error (MMSE)* criterion $\varphi^{\mathrm{MMSE}}(\underline{x}) = \hat{\sigma}(\underline{x})$ and the MV criterion yield identical adaptive sample points.

Integrated Mean Squared Error Criterion The *integrated mean squared error (IMSE)* is defined as the integral of the MSE over the entire design space ($\mathrm{IMSE} = \int_{\mathcal{X}} \hat{\sigma}(\underline{x}) d\underline{x}$) and often serves as a measure for overall model performance [69, 70]. In adaptive sampling, the IMSE criterion

$$\varphi^{\mathrm{IMSE}}(\underline{x}) = \int_{\mathcal{X}} \hat{\sigma}_{\mathcal{D}_i}(\underline{\nu}) d\underline{\nu} - \int_{\mathcal{X}} \hat{\sigma}_{\mathcal{D}_i \cup \{\underline{x}\}}(\underline{\nu}) d\underline{\nu} \tag{2.28}$$

indicates how much the IMSE of the model trained on the current design $\mathcal{D}_i$ is reduced by the addition of the sample point $\underline{x}$ [16, 24]. In contrast to the MMSE criterion, which selects the maximizer of the MSE as next sample point and hence minimizes the worst cast error, the IMSE criterion maximizes the average model performance. To evaluate the IMSE criterion for a point $\underline{x}$, two computationally

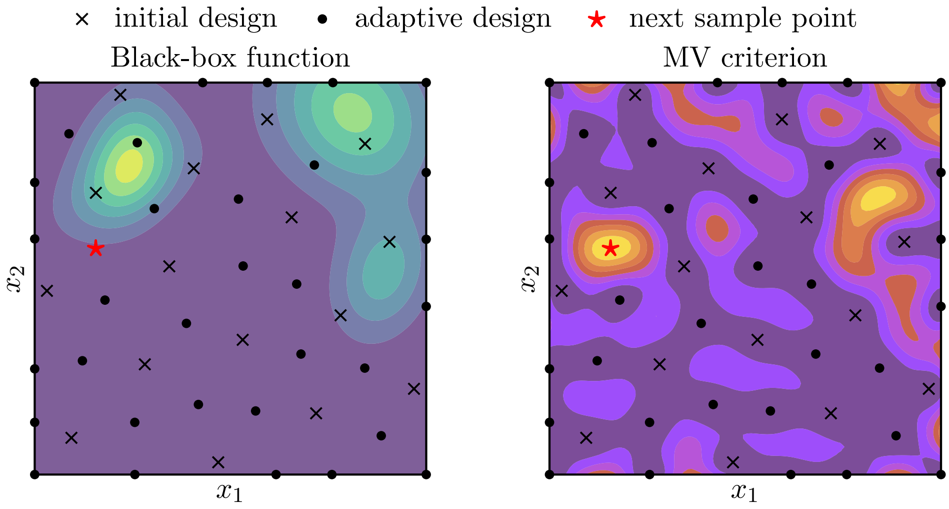

Figure 2.21: The space-filling initial design (16 points) and the adaptive sample points (33 points) generated based on the MV criterion are shown on the black-box function (left) and the MV criterion (right), where the MV criterion was evaluated on the GP that was trained on all initial and adaptive sample points. As indicated in the figure, the maximizer of the criterion is selected as next sample point.

expensive steps are required. The model has to be conditioned w.r.t. $\mathcal{D}_i \cup \{\underline{x}\}$ (the model parameters have to be adjusted) and the MSE must be integrated over the entire design space. The global optimization of the IMSE criterion to determine the next sample point requires a large number of evaluations of the IMSE criterion.

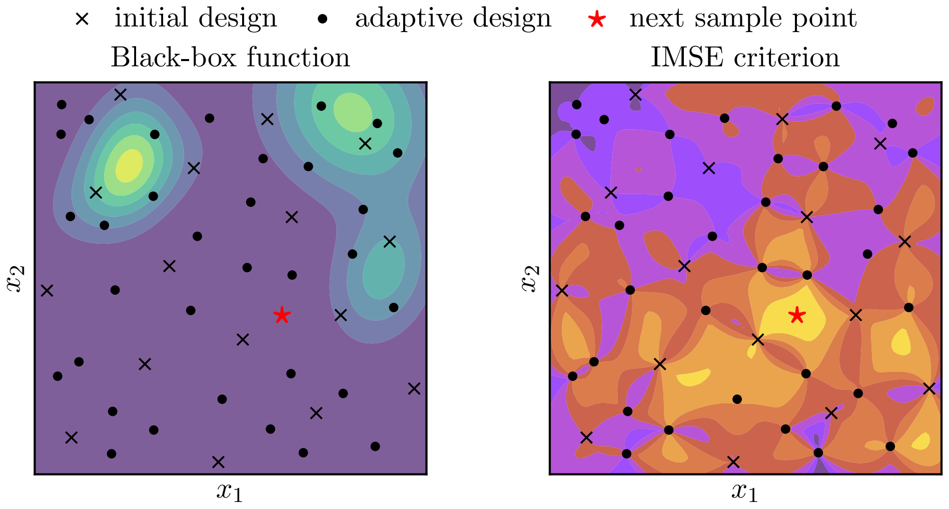

Figure 2.22: The space-filling initial design (16 points) and the adaptive sample points (33 points) generated based on the IMSE criterion are shown on the black-box function (left) and the IMSE criterion (right), where the IMSE criterion was evaluated on the GP that was trained on all initial and adaptive sample points. The maximizer of the criterion is selected as next sample point.

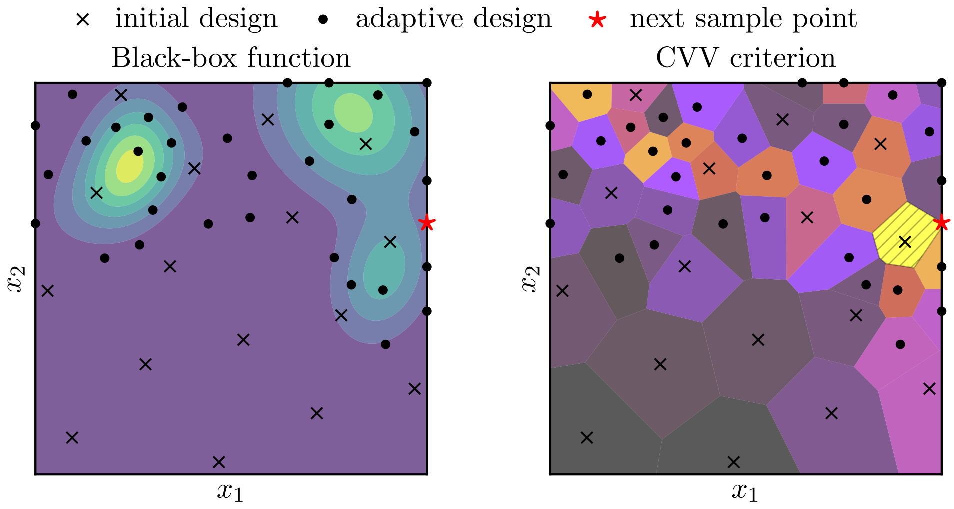

Figure 2.23: The space-filling initial design (16 points) and the adaptive sample points (33 points) generated based on the CVV criterion are shown on the black-box function (left) and the Voronoi tessellation constructed on the existing sample points (right). Each Voronoi cell is colored according to the corresponding value of the CVV criterion. The selected Voronoi cell is indicated by hatching and its vertex that is farthest from the associated existing sample point is selected as next sample point.

Cross-validation Voronoi Criterion Some adaptive sampling methods use Voronoi diagrams to determine the next sample point. To do this, a Voronoi diagram is first constructed on the existing sample points (just as for Voronoi sampling in Sec. 2.2.2), which is clipped to the design space as described in CVT in Sec. 2.2.1. Instead of a continuous sampling criterion, a cell selection criterion $S : j \to \mathbb{R}$ is used that maps the index of the respective Voronoi cell to the real line. Then the cell $j^* = \arg\max_j S(j)$ that maximizes the selection criterion is selected and the vertex of the selected cell which is farthest from the corresponding existing sample point $\underline{x}_{j^*}$ is used as the next sample point $\underline{x}_{m+1}$. In the *leave-one-out cross-validation Voronoi (CVV)* approach [36], the cell selection criterion

$$S^{\mathrm{CVV}}(j) = \left| \hat{f}_{-j}(\underline{x}_j) - \hat{f}(\underline{x}_j) \right| \tag{2.29}$$

is defined as the leave-one-out cross-validation error of the metamodel, where $\underline{x}_j$ is the sample point associated with the jth cell, $\hat{f}$ is the metamodel trained on the current design $\mathcal{D}_i$, and $\hat{f}_{-j}$ is the metamodel trained on the current design leaving out the sample of the jth Voronoi cell $\mathcal{D}_i \setminus \{\underline{x}_j, y_j\}$.

CHAPTER 3

Space-filling Designs for Environmental Sensing

Space-filling designs play a central role in global metamodeling as they guarantee information about the black-box function is collected from all regions of the design space [15, 16]. Coverage of the entire design space is particularly relevant when no information about the course of the black-box function is available [21, 27]. In this case, the more uniformly the sample points are distributed in the design space, the lower the expected global model error [21]. To do this, space-filling sampling methods must reliably identify the region in the design space with the lowest density of sample points [21]. Numerous mathematical formulations for describing space-filling properties as well as the resulting sequential and one-shot sampling methods for generating space-filling designs (such as the ones in Sec. 2.2.1 and Sec. 2.2.2) can be found in the literature and continue to be the subject of current research.

Global metamodeling is often used to replace an expensive simulator (e.g., numerical simulation [18]) or a physical experiment (e.g., wind tunnel testing [17]) with an easy to evaluate metamodel [15, 27, 71]. Applications in this context mostly have a high-dimensional mixed-domain design space and the black-box function is highly nonlinear or even discontinuous [27]. Therefore, in addition to good space-filling properties (e.g., measured via discrepancy [52]), other properties such as a large projected distance, a large intersite distance, or orthogonality of the design are desirable [27]. The projected distance $d^{\text{proj}} = \min_{l\in\{1,\dots,n\}} |\underline{x}_i[l] - \underline{x}_j[l]|$ between the points $\underline{x}_i$, $\underline{x}_j$ is the minimum distance between the two points along any dimension of the design space [27], where $\underline{x}[l]$ is the l-th element of $\underline{x}$. Hence, it defines the minimum distance between the points $\underline{x}_i$, $\underline{x}_j$ when projected from an n-dimensional to an $(n-1)$-dimensional design space along any axis. A design with a large minimal projected distance (a large projected distance between any two points of the design) is said to have good projected properties.

Latin hypercubes provide good projective properties by definition (refer to Sec. 2.2.1 for the definition of the Latin hypercube) and good space-filling properties when optimized using a space-filling criterion [27, 65, 72]. In addition, they scale well to high-dimensional design spaces which makes them one of the most widely used space-filling designs [18]. The requirement for good projective properties is based on the assumption that there are regions in the design space where a variation of a parameter (displacement of a point $\underline{x}$ along one axis in the design space) does not (or only slightly) result in a change of the black-box function [27]. Sample points that only differ in the dimension of such a parameter would lead to redundant evaluations of the black-box function [18, 72]. Since an evaluation of the black-box function is expensive and therefore the maximum information about the black-box function should be obtained per evaluation, sample points are generally preferred whose coordinates differ in all spatial directions.

However, environmental sensing differs significantly from applications such as computer simulations or complex physical experiments in terms of prerequisites. The design space is the location space. Hence, it is low-dimensional (usually two- or three-dimensional) and all dimensions are continuous and have the same scale [30, 73]. The black-box function is assumed to be Lipschitz continuous (e.g., a concentration distribution resulting from diffusion processes) and its variations depend on all dimensions of the design space [28, 29] which questions the need for good projective properties. Rather, it raises the question of which space-filling sampling methods provide the best results under the aforementioned circumstances.

In this chapter, we investigate space-filling sampling methods for global metamodeling of Lipschitz-continuous black-box functions on low-dimensional continuous design spaces. Based on considerations of model error, we derive three novel space-filling sampling methods in Sec. 3.1, one involving a modification of the maximin criterion. For each of our novel criteria we formulate a variant for one-shot as well as for sequential sampling. We evaluate our methods through extensive simulation and compare their performance to existing space-filling sampling methods in terms of global model error in Sec. 3.2.

3.1 Novel Space-filling Sampling Criteria for Lipschitz-continuous Black-box Functions

In this section, we derive three novel space-filling criteria for global metamodeling of black-box functions $f : \mathcal{X} \to \mathbb{R}$ on a low-dimensional continuous design space $\mathcal{X} \subset \mathbb{R}^n$ (e.g., $\mathcal{X} = [0,1]^2$), where n is the number of dimensions. We assume the black-box function f to be Lipschitz continuous on $\mathcal{X}$ which implies that there exists an $L > 0$ such that $|f(\underline{x}_i) - f(\underline{x}_j)| \leq L||\underline{x}_i - \underline{x}_j|| \; \forall \underline{x}_i, \underline{x}_j \in \mathcal{X}$. A metamodel $\hat{f} : \mathcal{X} \to \mathbb{R}$ is trained on error-free evaluations $y_l = f(\underline{x}_l) \in \mathbb{R}$ of the black-box function at sample points $X = \{\underline{x}_l\}_{l=1}^m$. The goal is to find space-filling sample points that minimize the expected global model error.

Pronzato et al. [59] motivate the minimax design for approximation of Lipschitz continuous functions. Based on the fact that the one-shot minimax criterion ϕ^{minimax} in (2.20) is defined as the maximum distance a point on the design space can have to its closest existing sample point, they conclude that the maximum model error

$$\begin{aligned} &\max_{\underline{x}\in\mathcal{X}} |f(\underline{x}) - f(\underline{x}_{i^*(\underline{x})})| \leq L \max_{\underline{x}\in\mathcal{X}} ||\underline{x} - \underline{x}_{i^*(\underline{x})}|| = L\phi^{\text{minimax}}(X) \qquad (3.1)\\ &\text{with } i^*(\underline{x}) = \arg \min_{i\in\{1,\dots,m\}} ||\underline{x} - \underline{x}_i|| \end{aligned}$$

is proportional to the minimax criterion. Hence, minimizing the minimax criterion also minimizes the upper limit for the maximum model error [59].

However, there are several difficulties associated with the minimax design. On the one hand, it is extremely difficult to compute [59]. Even for a small number of sample points in a low-dimensional design space, a (near-)optimal design is usually not found. On the other hand, the overall model performance is usually evaluated based on the global model error in global metamodeling, not the maximum possible model error. Numerous works in the field, such as [15, 17, 21, 23, 68, 74], have employed the global error between the true function and the metamodel (mean absolute error, mean squared error, (normalized) root mean squared error) on the design space for evaluation.

Another widely used method is to minimize the integral of the predictive standard deviation $\int_{\mathcal{X}} \hat{\sigma}(\underline{x}) d\underline{x}$ over the design space of a linear model (usually GPs / Kriging) conditioned on the sample points w.r.t. the sample points to determine the optimal design (cf. IMSE criterion in Sec. 2.2.3) [24, 65, 72]. However, this entails difficulties: the design is significantly dependent on the hyperparameters of the kernel function [65], and it is unclear how these must be selected. Second, it is often argued that the hyperparameters are independent of the samples [59]. However, this is only correct if the hyperparameters are specified independently of the samples. In fact, however, the hyperparameters are usually optimized as described in Sec. 2.1 based on the existing samples [38]. The optimal hyperparameters are then dependent on the samples and thus also the choice of the sample points, if the predictive standard deviation is used to determine the design. In addition, a sufficiently accurate estimation of the hyperparameters requires a minimum number of samples (while in case of an one-shot design no samples exist initially that could be used to determine the hyperparameters).

In Sec. 3.1.1 and Sec. 3.1.2, space-filling sampling criteria are derived based on global error measures. In addition, in Section Sec. 3.1.3, the maximin criterion is modified such that no sample points fall on the boundary of the design space and the properties of a minimax design are approximated. Finally, Sec. 3.1.4 defines a modified version of RGS with no sample points on the boundary, motivated by literature as well as by Sec. 3.1.1, Sec. 3.1.2.

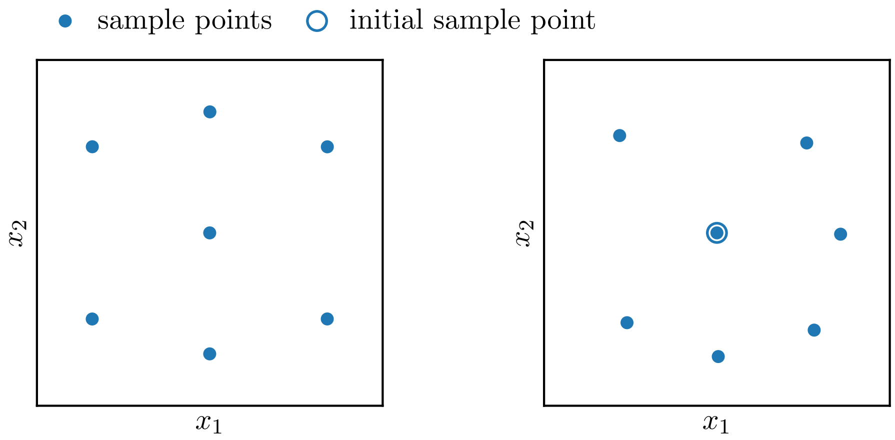

Figure 3.1: One-shot (left) and sequential (right) MD design of size 7. The distribution of the sequential samples is highly dependent on the initial sample point.

3.1.1 Mean Distance Criterion

Let the black-box function and the metamodel be Lipschitz continuous. Let there be an $L > 0$ (Lipschitz constant) such that $|f(\underline{x}_i) - f(\underline{x}_j)| \leq L||\underline{x}_i - \underline{x}_j||$ and $|\hat{f}(\underline{x}_i) - \hat{f}(\underline{x}_j)| \leq L||\underline{x}_i - \underline{x}_j|| \; \forall \underline{x}_i, \underline{x}_j \in \mathcal{X}$. Then for any $\underline{x} \in \mathcal{X}$ the absolute error

$$|f(\underline{x}) - \hat{f}(\underline{x})| \leq 2L \min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i|| \tag{3.2}$$

between the black-box function and the metamodel will in the worst case be the distance between $\underline{x}$ and its closest existing sample point scaled by twice the Lipschitz constant. The factor of two arises since the variations of the black-box function and the metamodel evolve in opposite directions in the worst case with maximum rate of change. By integrating the absolute model error in (3.2) over the entire design space, we obtain the *mean absolute error (MAE)*

$$\int_{\mathcal{X}} |f(\underline{x}) - \hat{f}(\underline{x})| d\underline{x} \leq 2L \int_{\mathcal{X}} \min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i|| d\underline{x} \tag{3.3}$$

as a measure of global model performance. It follows that minimizing the integral over the distance from each point in the design space to the nearest exiting sample point minimizes the upper bound for the MAE. On this basis, we define the *mean distance (MD)* criterion

$$\phi^{\mathrm{MD}}(X) = \int_{\mathcal{X}} \min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i|| d\underline{x} \tag{3.4}$$

as the integral of the distance from each point $\underline{x}$ in the design space to its nearest point in X. The optimal one-shot MD design is obtained by optimization of the MD criterion in (3.4). A sequential version of the MD criterion

$$\varphi^{\mathrm{MD}}(\underline{x}) = \int_{\mathcal{X}} \min_{\underline{x}_i \in X \cup \{\underline{x}\}} ||\underline{u} - \underline{x}_i|| d\underline{u} \tag{3.5}$$

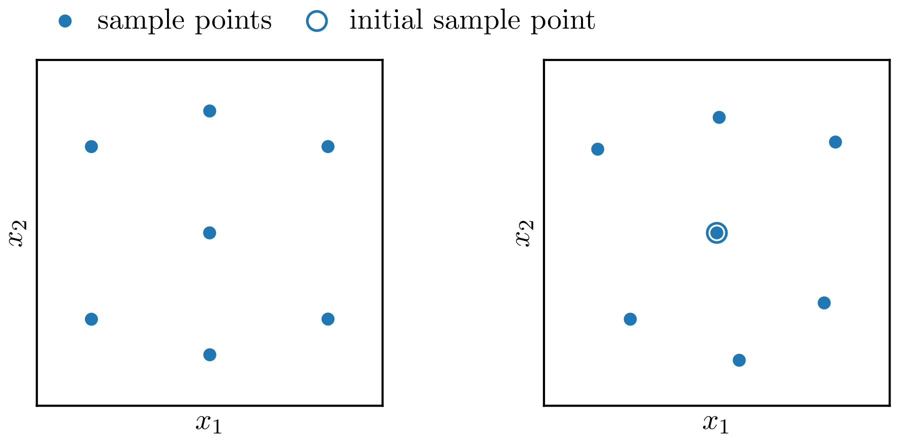

Figure 3.2: One-shot (left) and sequential (right) RMSD design of size 7. The distribution of the sequential samples is highly dependent on the initial sample point.

can be formulated by fixing the existing sample points X and optimizing the criterion w.r.t. the location of the next sample point $\underline{x}$ only. Fig. 3.1 shows examples for an one-shot and a sequential MD design.

3.1.2 Root Mean Squared Distance Criterion

In analogy to the considerations in Sec. 3.1.1, we aim to employ the *root mean squared error (RMSE)*

$$\sqrt{\int_{\mathcal{X}} (f(\underline{x}) - \hat{f}(\underline{x}))^2 d\underline{x}} \tag{3.6}$$

in this section as the foundation for establishing space-filling sampling criteria. To do so, we first square the absolute model error in (3.2)

$$|f(\underline{x}) - \hat{f}(\underline{x})|^2 = (f(\underline{x}) - \hat{f}(\underline{x}))^2 \leq 4L^2 (\min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i||)^2 \tag{3.7}$$

to get the inner term of the RMSE in (3.6). Substituting the inequality in (3.7) into the RMSE in (3.6)

$$\sqrt{\int_{\mathcal{X}} (f(\underline{x}) - \hat{f}(\underline{x}))^2 d\underline{x}} \leq 2L \sqrt{\int_{\mathcal{X}} (\min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i||)^2 d\underline{x}} \tag{3.8}$$

we obtain an upper bound for the RMSE, depending on the location of the sample points in the design space.

On this basis we define the *root mean squared distance (RMSD)* criterion

$$\phi^{\mathrm{RMSD}}(X) = \sqrt{\int_{\mathcal{X}} (\min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i||)^2 d\underline{x}} \tag{3.9}$$

as the integral of the root mean squared distance of all points $\underline{x}$ on the design space to their nearest existing sample point in X. As with all one-shot criteria, the optimal one-shot RMSD design is obtained by optimizing the RMSD criterion in (3.9) w.r.t. the locations of the sample points X. A sequential version of the RMSD criterion

$$\varphi^{\text{RMSD}}(\underline{x}) = \sqrt{\int_{\mathcal{X}} \left(\min_{\underline{x}_i \in X \cup \{\underline{x}\}} ||\underline{u} - \underline{x}_i|| \right)^2 d\underline{u}} \tag{3.10}$$

can be formulated by fixing the existing sample points X and optimizing the criterion w.r.t. the location of the next sample point $\underline{x}$ only. Fig. 3.2 shows examples for an one-shot and a sequential RMSD design.

Remark. For implementation, the integral contained in the MD and RMSD criterion can be approximated using Monte Carlo integration in low-dimensional design spaces. As the dimensionality of the design space increases, Monte Carlo integration suffers from the cause of dimensionality, which makes the evaluation of the integral computationally intensive. In addition, the integral improves the robustness and efficiency of the global optimization of the one-shot designs because, unlike minimax and maximin based criteria, a gradient can be used for local search.

3.1.3 Modified Maximin Criterion

The maximin design described in Sec. 2.2.1 is one of the most widely used space-filling designs. Since the maximin criterion maximizes the minimum distance between sample points, more sample points are placed on the boundary of the design space than in the minimax design [59, 75]. This results in larger gaps in the design space [60], leading to a larger possible model error for the maximin design than for the minimax design, as shown in (3.2). Intuitively, this can be explained by the fact that due to the limited rate of change of f, a sample approximates the function value of f well near the sample point and the approximation decreases with distance from the sample point. Thus, when sample points are on the boundary of the design space, regions where the sample provides a good representation of the actual function value lie outside the design space and the resulting larger gaps between sample points within the design space increase the upper bound for the model error there.

The MD and RMSD criteria intrinsically account for this by integrating distance measures over the design space, yielding an even distribution of sample points with no sample point on or close to the boundary, as shown in Fig. 3.1 and Fig. 3.2. We define the *modified maximin (mod. maximin)* criterion

$$\phi^{\text{mod. maximin}}(X) = \min\Big(\min_{\substack{\underline{x}_i, \underline{x}_j \in X \\ i \neq j}} ||\underline{x}_i - \underline{x}_j||, 2 \min_{\underline{x}_i \in X} ||\underline{x}_i - \underline{b}^*(\underline{x}_i)|| \Big) \tag{3.11}$$

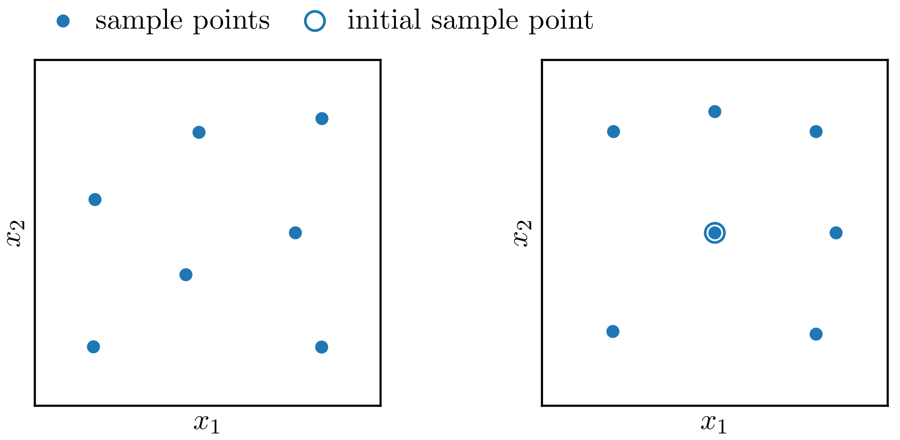

Figure 3.3: One-shot (left) and sequential (right) mod. maximin design of size 7. The distribution of the sequential samples is highly dependent on the initial sample point.

which takes into account the distance to the boundary $\partial\mathcal{X}$ of the design space in addition to the distance between the sample points, where $\underline{b}^*(\underline{x}_i) = \arg\min_{\underline{x}\in\partial\mathcal{X}} ||\underline{x} - \underline{x}_i||$ is the point in $\partial\mathcal{X}$ that is closest to $\underline{x}_i$.

This ensures that the mod. maximin design exhibits the space-filling characteristics of the maximin design while avoiding the placement of points on or near the boundary of the design space. Thus, the modified maximin criterion combines properties of the MD criterion and the maximin criterion. In addition, the mod. maximin criterion can be considered an approximation to the minimax criterion. This is because the maximin and minimax designs are related as described in Sec. 2.2.1. In fact, an inward-shifted maximin design can serve as a good approximation of a minimax design [59]. Unlike the minimax design, however, the mod. maximin design no longer contains points near the boundary of the design space. A sequential version of the mod. maximin criterion

$$\varphi^{\text{mod. maximin}}(\underline{x}) = \min\big(\min_{\underline{x}_i\in X} ||\underline{x} - \underline{x}_i||, 2\min_{\underline{b}\in\partial\mathcal{X}} ||\underline{x} - \underline{b}||\big) \tag{3.12}$$

can be formulated by fixing the existing sample points X and considering the distances between the location of the next sample point $\underline{x}$ and the existing samples as well as the boundary only. Fig. 3.3 shows examples for an one-shot and a sequential mod. maximin design.

3.1.4 Modified Regular Grid Sampling

As introduced in Sec. 2.2.1, RGS maximizes the intersite distance between the sample points while providing an even distribution of space-filling sample points [27]. However, it contains a large number of sample points on the boundary of the design space which negatively impacts the expected global model error, as discussed for

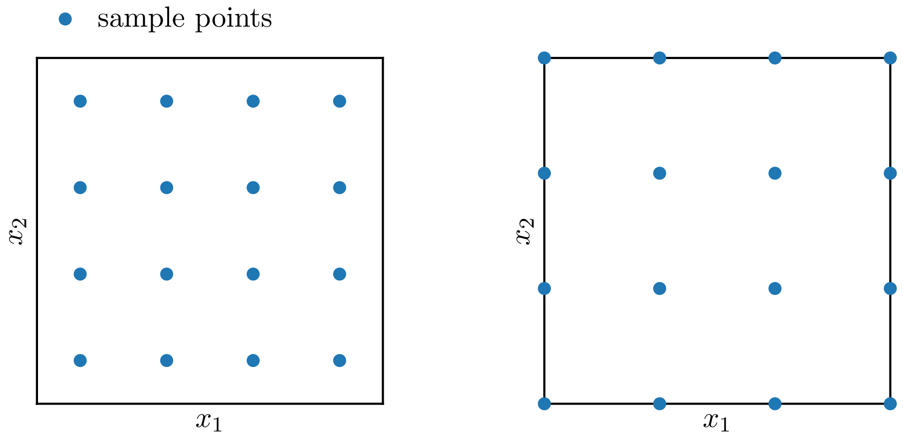

Figure 3.4: Mod. RGS (left) and RGS (right) with 16 sample points.

the maximin design in Sec. 3.1.3. Following the same reasoning as in Sec. 3.1.3, we define *modified regular grid sampling (mod. RGS)* as an inward shifted variant of RGS, where the distance between the lowest and highest level to the bounds is half the distance between the equally spaced levels in each dimension of the design space. This results in a more even occupancy of the design space. Each sample point of the mod. RGS design represents the same proportion of the design space, which is not true for the RGS design. If the design space is an n-dimensional hypercube (a square in 2d), then in the RGS design each sample point in the corner of the design space represents a 2^{-n}th of the proportion and each sample point on the sides represents a 2^{-n+1}th of the proportion occupied by an inner sample point. A mod. RGS and a RGS design are shown in Fig. 3.4.

3.2 Evaluation

In this section our novel space-filling criteria MD, RMSD, and mod. maximin introduced in Sec. 3.1.1 – Sec. 3.1.4 are evaluated against state-of-the-art literature described in Sec. 2.2.1 and Sec. 2.2.2. First the experimental setup is described in Sec. 3.2.1. Then the properties of the MD and RMSD designs are analyzed in Sec. 3.2.2 and an intensive numerical comparison to literature methods is provided in Sec. 3.2.3.

3.2.1 Experimental Setup

For evaluation, the space-filling designs from this chapter and the literature methods from Sec. 2.2.1 – Sec. 2.2.2 were computed for the design space $\mathcal{X} = [0,1] \times [0,1]$. The one-shot designs were computed as follows:

CVT: To calculate the CVT, the class CVTSampling of the Python package IDAES [76] was used, which uses McQueens algorithm to optimize the CVT.

LHS: The implementation of the enhanced stochastic evolutionary (ESE) algorithm [72] of the Python surrogate modeling toolbox (SMT) [77] was used along with the maximin criterion for OLHS.

Hammersley: The implementation of the Python package Scikit-Optimize [78] was used for Hammersley sampling.

One-shot designs based on criteria: One-shot designs were generated by optimizing the criteria through the genetic algorithm (GA) optimizer of the Python package PyMOO [79] for a duration of 24 h per design and number of sample points with subsequent local gradient-based optimization through the L-BFGS-B algorithm of Python package SciPy [80]. In addition, local optimizations were performed, each starting from 250 randomly generated initial distributions of sample points. The best design w.r.t. the criterion of the two approaches was then selected.

Sequential designs that include optimization of a sequential sampling criterion for computation of the next sample point in each iteration of the algorithm were treated differently based on their numerical properties. The maximin and mod. maximin criteria were first evaluated on a grid of 10^4 candidate points. Then, the best candidate point was used as starting point for a local gradient-based optimization. Since the MD and RMSD criterion require the computation of an integral they are computationally more expensive to evaluate. Hence, each vertex of a Voronoi tessellation constructed on the existing sample points was used as starting point for a local gradient-based optimization (performing multiple gradient-based optimizations from varying starting points is often referred to as n-starts optimization). Then, the best optimization result was selected as next sample point. The integral over the distance measures of the MD and RMSD criteria in (3.5) and (3.10) were approximated by means of Monte Carlo integration using 250^2 Hammersley sample points. The point $[\frac{1}{3}, \frac{1}{3}]$ was chosen as initial sample point of the sequential designs.

All designs were computed for different numbers of samples and used to evaluate $N^{\text{runs}} = 100$ randomly generated black-box functions f_q, $q \in \{1, \ldots, N^{\text{runs}}\}$. Deterministic evaluations (no sampling noise) of the black-box functions were performed. Each of the black-box functions was modeled as weighted sum of ten squared exponential functions

$$f_q(\underline{x}) = \sum_{t=1}^{10} w_t \cdot e^{-\frac{1}{2}(\underline{x}-\underline{\bar{x}}_t)^\top \boldsymbol{S}_t^{-1}(\underline{x}-\underline{\bar{x}}_t)} , \tag{3.13}$$

where $w_t \in [0, 1]$ are uniformly distributed random weights, the locations of the squared exponentials $\underline{\bar{x}}_t \in \mathcal{X}$ are uniformly distributed on the design space $\mathcal{X}$, and $\boldsymbol{S}_t$ are covariance matrices (positive semidefinite, and symmetric matrices with non-

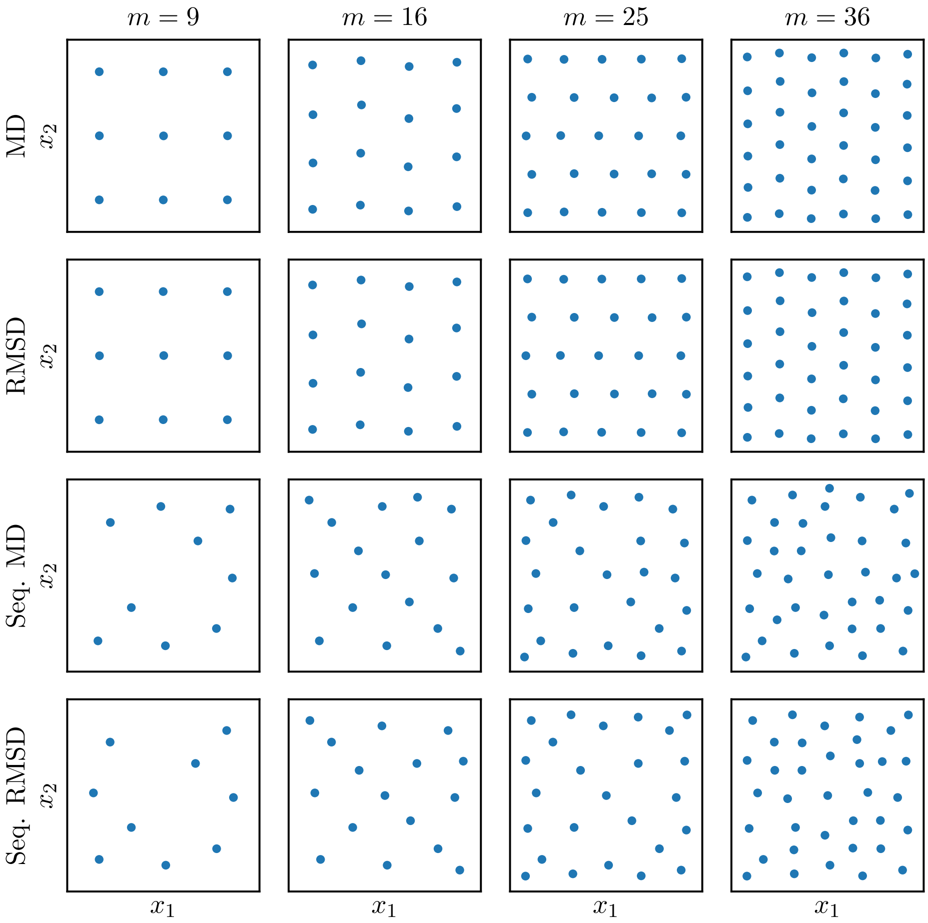

Figure 3.5: One-shot and sequential MD and RMSD designs for different numbers m of sample points in square.

negative diagonal elements). The covariance matrices were generated by rotating a diagonal matrix with uniformly distributed random entries between $[0.001, 0.01]$ by a uniformly distributed random angle using a rotation matrix. This resulted in smooth distributions that could already be well approximated with 100 samples. A GP with an SE kernel was used as a metamodel and the hyperparameters of the kernel were obtained by optimizing the log marginal likelihood as described in Sec. 2.1. To evaluate the accuracy of the metamodel, the design space $\mathcal{X}$ was discretized into $\mathcal{X}^{\text{eval}} \subset \mathcal{X}$ by means of a 100 by 100 mod. RGS, resulting in $N^{\text{eval}} = 10,000$ evaluation points $\underline{x}_\eta \in \mathcal{X}^{\text{eval}}$, $\eta = 1, \ldots, N^{\text{eval}}$. Based on the evaluation points, the normalized root-mean-square error

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{N^{\text{eval}}} \sum_{\eta=1}^{N^{\text{eval}}} \left(\hat{f}(\underline{x}_\eta) - f(\underline{x}_\eta) \right)^2}}{f_{\max} - f_{\min}} \tag{3.14}$$

and the normalized mean absolute error

$$\text{NMAE} = \frac{\frac{1}{N^{\text{eval}}} \sum_{\eta=1}^{N^{\text{eval}}} \left| \hat{f}(\underline{x}_\eta) - f(\underline{x}_\eta) \right|}{f_{\max} - f_{\min}} \tag{3.15}$$

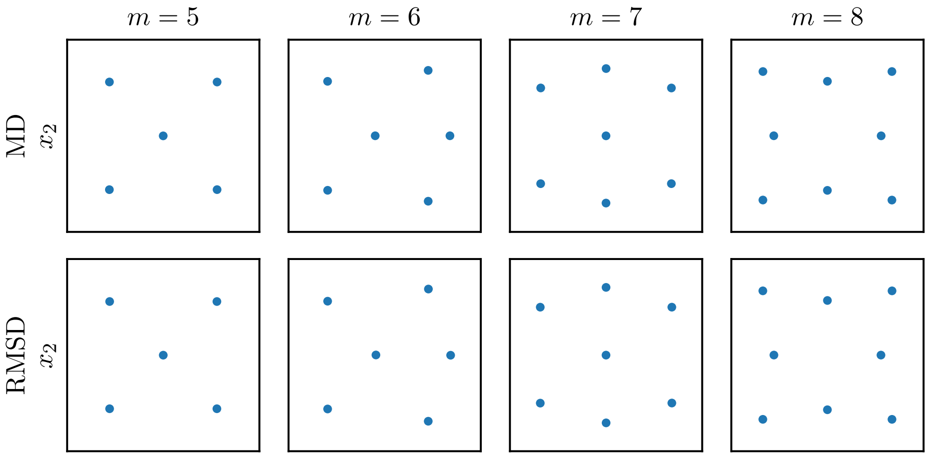

Figure 3.6: One-shot MD and RMSD designs for different numbers m of sample points in square.

were computed as a measure for the overall error between the black-box function and the metamodel, where $f_{\max} = \max_{\eta} f(\underline{x}_\eta)$ and $f_{\min} = \min_{\eta} f(\underline{x}_\eta)$.

3.2.2 Properties of the MD and RMSD Designs

In this section we investigate the properties of space-filling designs based on the MD and RMSD criteria introduced in Sec. 3.1.1 and Sec. 3.1.2. Specifically, we compare the distribution of sample points on the design space and compare the resulting model error. Since the MD and RMSD criteria have been derived based on different error measures (MAE and RMSE), we evaluate whether this is reflected in the error measures used to evaluate the global model error of the metamodel trained on samples of the black-box function, taken at sample points generated by means of the criteria.

Fig. 3.5 shows the one-shot and sequential MD and RMSD designs for the square numbers from 9 to 36 of sample points. Interestingly, the One-shot MD and RMSD designs are identical except for minor numerical differences (varying termination of the local gradient-based optimization close to the optimum). For a number of $m = 9$ sample points, the design forms a mod. RGS, while for $m \in \{16, 25, 36\}$, the designs look like mod. RGS, where every second row of points is parallel and slightly shifted relative to the other parallel rows (or columns, since the value of the criterion is invariant to transposition of the design on a square). Hence the characteristics of the one-shot MD and RMSD designs are similar to mod. RGS. In contrast to mod. RGS, the one-shot MD and RMSD designs can be computed for any number of samples and are not limited to a number of samples that can form a grid (square numbers in 2d), as shown in Fig. 3.6 for $m \in \{5, 6, 7, 8\}$.

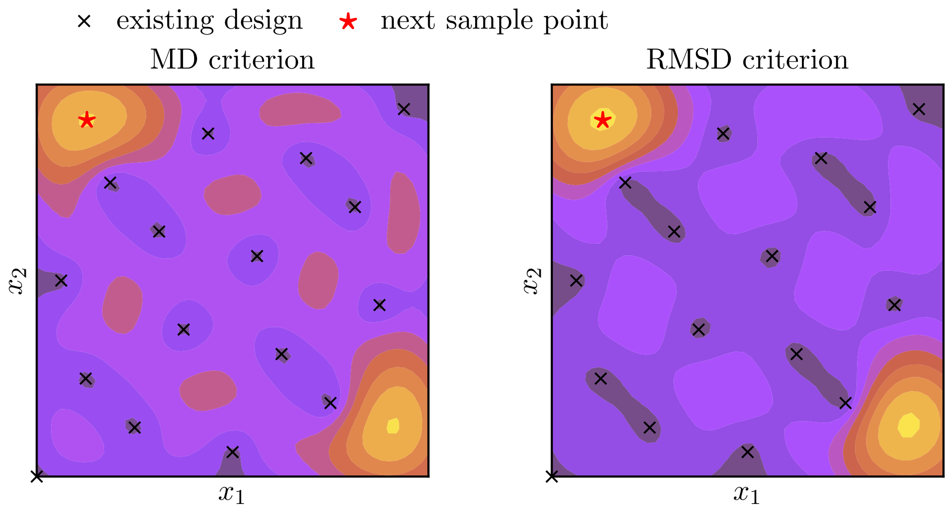

Figure 3.7: Existing (Hammersley) sample points and next sequential sample point on top of the sequential MD and RMSD criteria (high values are represented by yellow and low values by purple) evaluated using the existing sample points. The next sample point is located at the maximum of the respective criterion.

The sequential designs in Fig. 3.5 differ for the MD and RMSD criterion. However, their distribution of sample points is similar. This results from the sequential MD and RMSD criteria having similar characteristics, which is illustrated in Fig. 3.7 and Fig. 3.8. When the existing samples are evenly distributed across the design space (Fig. 3.7), the maxima of the two criteria are located very close to each other. In case of unevenly distributed sample points, such as randomly generated points that leave gaps in the design space and cluster in other parts, the criteria still have the same characteristics but take the maximum value at different points in the design space, as shown in Fig. 3.8.

Fig. 3.9 shows the mean NMAE and RMSE of the one-shot and sequential MD and RMSD designs over all $N^{\text{runs}} = 100$ random black-box functions versus the number of sample points. For both error measures, the curves of the sequential designs and the one-shot designs are nearly identical, respectively. Accordingly, no difference in the performance of the MD and RMSD criteria can be found for the two error measures. However, it is noticeable that the difference between the sequential and one-shot designs is larger for the NRMSE than for the NMAE. For the NMAE the sequential designs are even slightly below the one-shot designs for a small number of sample points. In general, the one-shot designs lead to smaller model errors, which is due to the global optimization of all sample points. Since the MD and RMSD designs yield the same performance we will use only the MD design in the following sections for better clarity. In addition, we will only use the NRMSE to evaluate the model error in the remainder of the thesis.

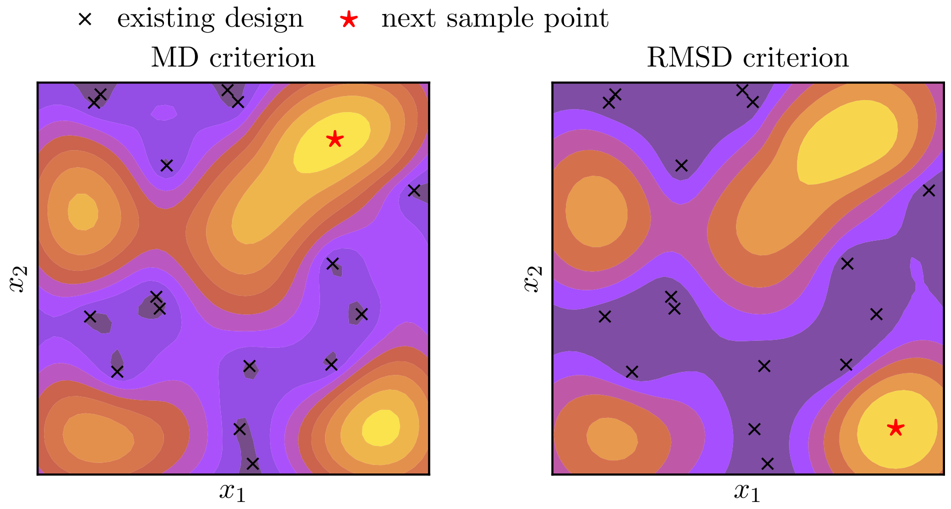

Figure 3.8: Existing (randomly generated) sample points and next sequential sample point on top of the sequential MD and RMSD criteria (high values are represented by yellow and low values by purple) evaluated using the existing sample points. The next sample point is located at the maximum of the respective criterion.

3.2.3 Numerical Comparison to Literature Methods

In this section, we compare the performance of the sequential MD, mod. maximin, and mod. RGS designs from Sec. 3.1 with the literature methods from Sec. 2.2. To limit the number of curves per figure, Fig. 3.10 and Fig. 3.11 first compare the one-shot and sequential designs from the literature in terms of NRMSE. The best literature methods are then compared with the MD and mod. maximin designs in Fig. 3.13 and Fig. 3.14.

In Fig. 3.10, CVT, LHS, minimax design, RGS, Hammersley sampling, maximin design, and RGS are evaluated. It should be noted that despite the extensive global optimization of the designs (except Hammersley sampling, RGS, and mod. RGS, as these are constructed) it cannot be guaranteed that a global optimum has been found. This is especially true for the minimax design, which is visibly suboptimal even for $m = 16$ sample points and can no longer be meaningfully used for evaluation for $m > 25$ sample points.

In the sequential designs in Fig. 3.11, random sampling is outperformed by sequential maximin and sequential Voronoi. Sequential maximin and sequential Voronoi yield almost identical results. Therefore, only sequential maximin is used in the further course of the evaluation.

When comparing the one-shot designs from the literature with the MD, mod. maximin, and mod. RGS in Fig. 3.13, the NRMSE of mod. maximin and LHS show an almost identical course. For a few sample points, mod. maximin is slightly below LHS. In

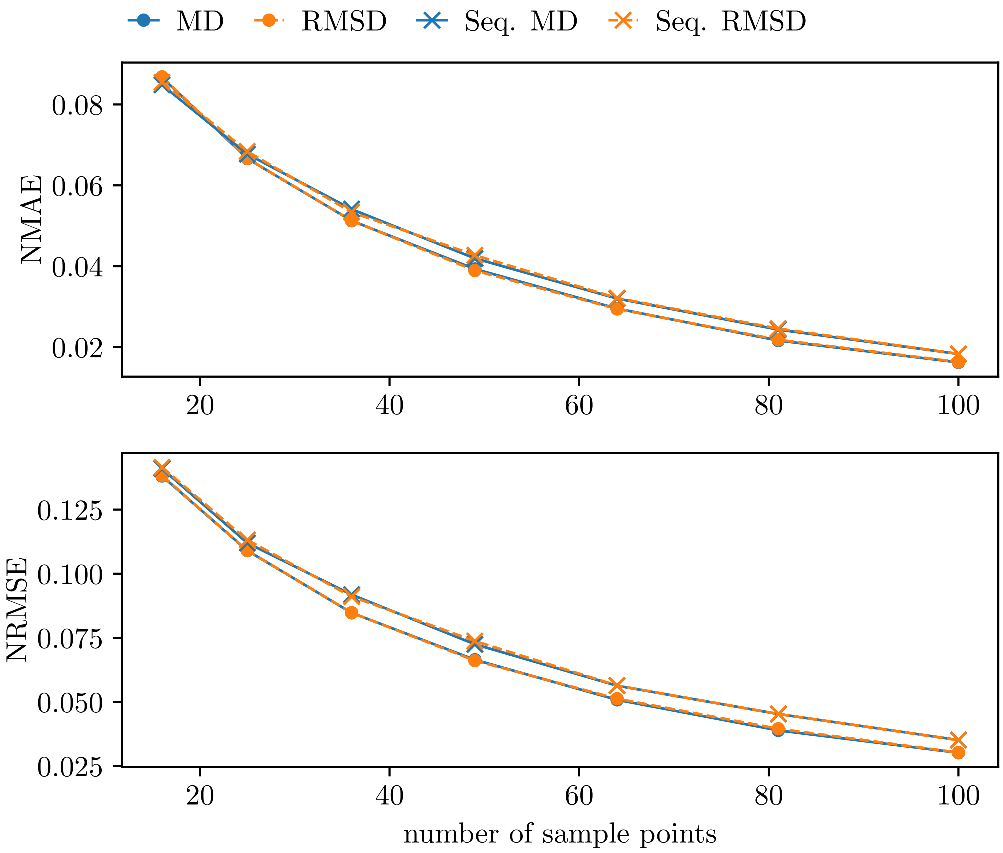

Figure 3.9: The mean of the error measures NMAE and NRMSE over $N^{\text{runs}} = 100$ randomly generated black-box functions versus the number of sample points.

the range of $36 \leq m \leq 64$, mod. maximin and LHS are close to each other and for $m > 64$ mod. maximin diverges upward. This is because as the number of sample points increases, no optimum was found in the global optimization and no sufficient local optimum was found. For illustration, mod. maximin designs for 36 and 81 sample points are shown in Fig. 3.12. It can be easily seen that with a larger number of sample points, gaps appear in the design space and thus uniform coverage by sample points is no longer guaranteed. The One-shot MD design and mod. RGS provide an almost identical NRMSE in Fig. 3.13 and outperform both mod. maximin as well as LHS.

The sequential MD and mod. maximin designs also yield an almost identical NRMSE and outperform the sequential maximin design, as shown in Fig. 3.14.

3.3 Summary

In this chapter, we presented three novel space-filling designs that minimize the expected global model error when approximating a Lipschitz-continuous black-box

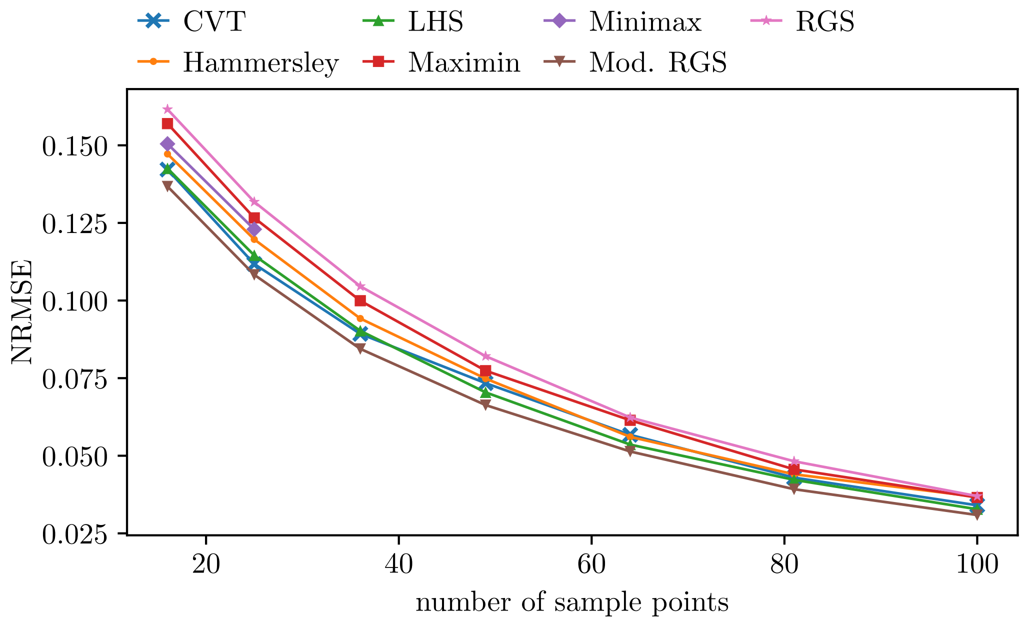

Figure 3.10: The mean of the NRMSE over $N^{\text{runs}} = 100$ randomly generated black-box functions of space-filling one-shot designs from the literature versus the number of samples.

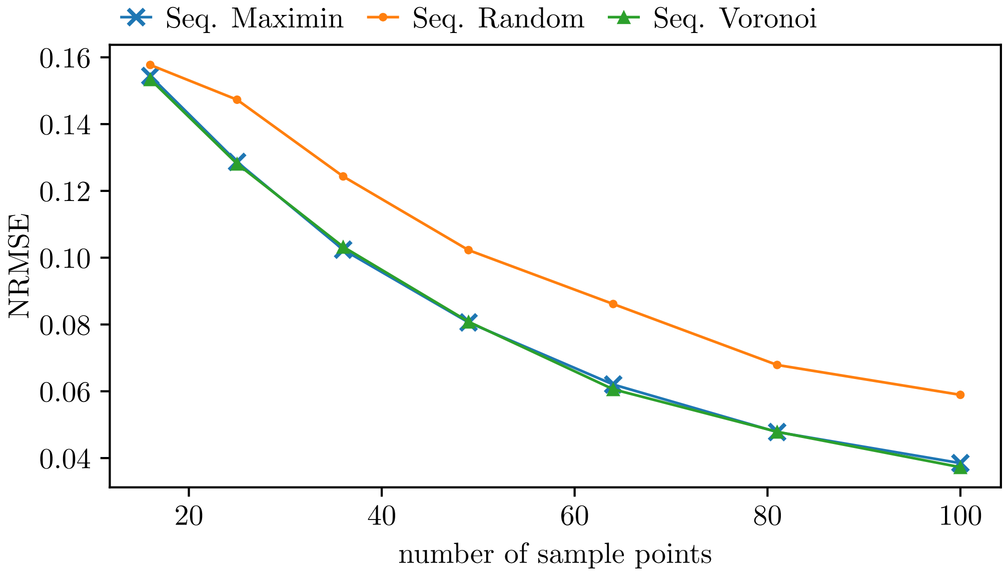

Figure 3.11: The mean of the NRMSE over $N^{\text{runs}} = 100$ randomly generated black-box functions of space-filling sequential designs from the literature versus the number of samples.

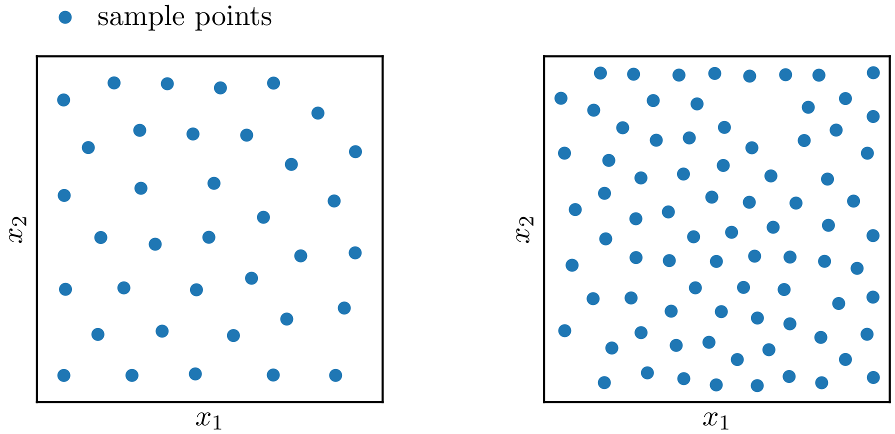

Figure 3.12: Mod. maximin designs of size 36 (left) and 81 (right) that have been optimized as described in Sec. 3.2.1. Both are suboptimal solutions to the global optimization problem, with the solution getting worse as the number of sample points increases (gaps in the design space due to an uneven distribution of sample points).

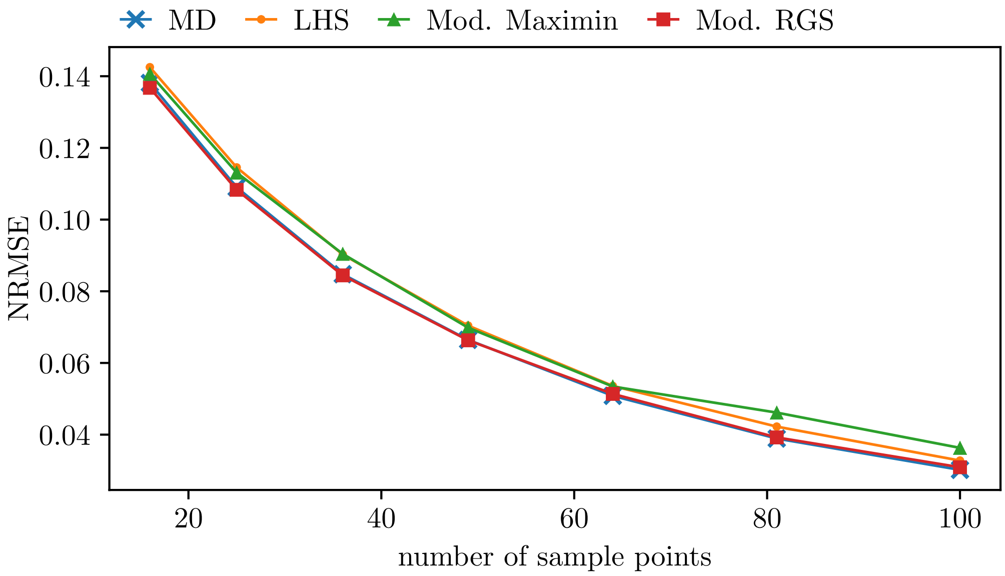

Figure 3.13: The mean of the NRMSE over $N^{\mathrm{runs}} = 100$ randomly generated black-box functions of the best space-filling one-shot designs from the literature and our novel one-shot MD and mod. maximin designs versus the number of samples.

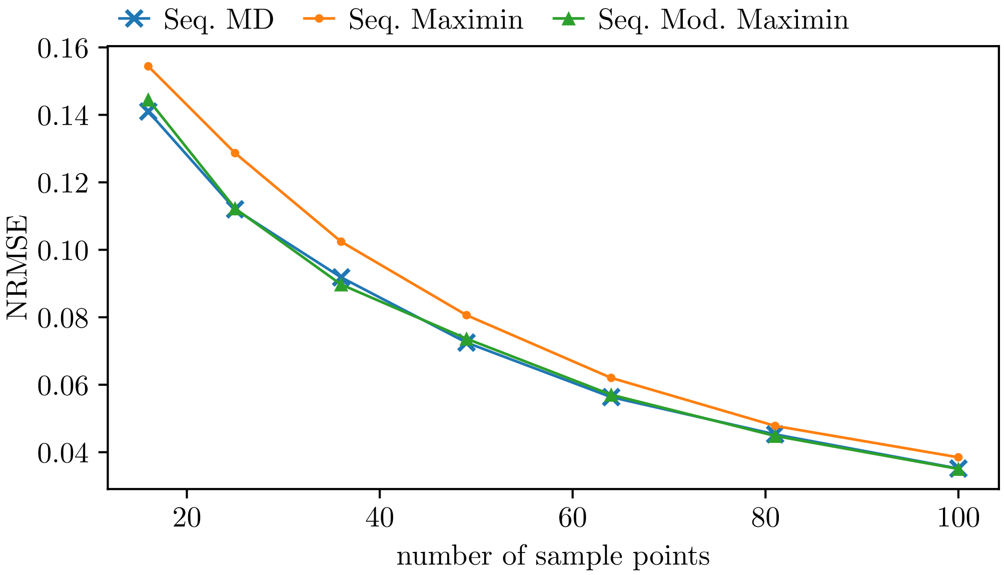

Figure 3.14: The mean of the NRMSE over $N^{\mathrm{runs}} = 100$ randomly generated black-box functions of the best space-filling sequential designs from the literature and our novel sequential MD and mod. maximin designs versus the number of samples.

function by a metamodel when no information about the black-box function is known and hence not considered for selection of the sample points. The MD and RMSD criteria were derived using the MEA and RMSE, respectively, between the metamodel and the black-box function. The mod. maximin criterion is a modification of the widely used maximin criterion, in which the distance to the boundary of the design space is taken into account and thus the disadvantages of the maximin design (points on the boundary of the design space and consequently larger gaps between the sample points in the design space) are prevented.

For all three criteria, both an one-shot and a sequential variant were defined. During the simulative evaluation it was shown that the proposed criteria outperform the methods commonly used in the literature. An exception is the mod. RGS, which provides the same NRMSE as the one-shot MD and RMSD design. However, the MD and RMSD design can be computed for any number of sample points, while the mod. RGS requires that a grid can be constructed using the sample points. The one-shot designs provide a smaller NRMSE between the model and the black-box function due to the global optimization (or construction) of the space-filling design.

Interestingly, the mean errors for the MD and RMSD criteria are almost identical regardless of the error measures NMAE and NRMSE used. Thus, the derivation based on different error measures is not directly reflected in the distribution of sample points. The minimax criterion is so challenging to optimize that no practically usable design can be found even for a small number of sample points after excessive global optimization. Global optimization of the other criteria cannot guarantee that an optimal design will be found as well – especially for a large number of sample points. However, it has been found that the optimization of the MD and RMSD criteria are more robust due to the integral included. More robust in the sense that a homogeneous distribution in the design space is achieved even for a larger number of sample points, although the design may only be a suboptimum of the global optimization.

CHAPTER 4

Weighted Explorative Sampling Criteria for Environmental Sensing

In this chapter we introduce a novel class of adaptive sampling criteria, specifically designed for adaptive sampling of non-negative Lipschitz-continuous environmental phenomena, such as concentration functions. In the following, the context of the problem considered in this chapter is first explained, followed by a formal definition of the problem in Sec. 4.1. Subsequently, our novel class of adaptive sampling criteria is introduced in Sec. 4.2 and simulatively evaluated in Sec. 4.3.

In Chap. 3, we investigated space-filling designs for the approximation of Lipschitz-continuous black-box functions. We assumed that no prior knowledge about the black-box function is available, which is why we used space-filling designs that select sample points without any information about the black-box function. The adaptive sampling methods presented in Sec. 2.2.3 iteratively add new sample points to the design, taking into account all available information about the black-box function to select the next sample point (or the next sample points, if multiple sample points are suggested per iteration [71]) [15, 21]. Since the choice of sample points in adaptive sampling methods is based on prior knowledge about the black-box function, adaptive sampling methods are usually warm-started with a space-filling initial design. As the knowledge about the black-box function improves with each iteration of the algorithm, the choice of sample points can also become more targeted as the number of samples increases [21]. In this way, adaptive sampling methods aim to collect more information about the black-box function per sample and thus achieve better model accuracy with the same number of samples than the space-filling methods from Chap. 3 [16].

In this chapter, we use these properties of adaptive sampling methods to reconstruct positive-valued environmental phenomena (e.g., concentration functions) as precisely as possible for a given number of sample points (or to achieve a good reconstruction of the phenomenon with as few sample points as possible). We examine the scenario in which a mobile robot undertakes sequential, expensive sampling operations.

Specifically, this requires the mobile robot to interrupt its travel in order to perform time-consuming and/or energy-intensive sampling. This necessitates maximizing the informational yield from each sample. Furthermore, the model should exhibit increased accuracy (more sample points should be placed) in regions of high function values than in regions of low function values of the phenomenon.

A prime example for this scenario is the project ROBDEKON [14], where an autonomous construction machine is used to carry out an expensive and time-consuming dynamic probing on a contaminated site. The resulting samples are then used to reconstruct the distribution of pollutants in the soil. Based on the metamodel of the pollutant distribution, regions that exceed a legal limit value are then to be precisely removed. Accordingly, it is important to achieve high model accuracy in regions with high concentration values, while in regions with low concentration values it is sufficient to know that there is little (or no) contamination. A formal definition of the considered problem can be found in Sec. 4.1.

As already described in Sec. 2.2.3, there are many adaptive sampling criteria in the literature that pursue different strategies for selecting particularly informative sample points [15]. The MV and IMSE criteria defined in Sec. 2.2.3 are based on the predictive uncertainty of the metamodel [16, 70] (usually the predictive standard deviation of a GP). While the MV criterion corresponds directly to the predictive variance evaluated at a candidate point $\underline{x}$ in the case of a GP, the GP needs to be conditioned on $\underline{x}$ when the IMSE criterion is evaluated. As a result, when optimizing the IMSE criterion in the design space, the GP must be reconditioned very frequently, which makes the use of the IMSE criterion for selecting the next sample point computationally expensive.

Other popular adaptive sampling methods are Lipschitz sampling [39], LOLA-Voronoi [23], and CVV [36], which exploit Voronoi tessellations based on the existing sample points. Lipschitz sampling [39] uses the Voronoi cells for estimation of the local Lipschitz constant. Based on this, a continuous criterion is defined that favors points that have both a large distance to the existing sample points and a large local Lipschitz constant (large gradient). In LOLA-Voronoi [23], the Voronoi cell with the largest estimated local gradient is identified and then the vertex of the cell furthest from the corresponding existing sample point is selected as the next sample point. The CVV criterion described in Sec. 2.2.3 is among best known adaptive sampling criteria in the context of global metamodeling. It owes its popularity to its performance and robustness for a variety of different black-box functions. There are several related methods based on the cross-validation error in the literature, such as [71, 81–84].

Approaches also exist that weight a criterion based on the prediction variance with the cross-validation error [84, 85]. This combines two different ways of estimating the model error (quantification of the model error by cross-validation and the prediction variance). Further work deals with correction factors and broad mechanisms to make such methods more robust [84]. In the area of conditional optimization and reliability analysis, the metamodel should often have a particularly high accuracy around a specified target value of the black-box function [69]. The authors in [69] present a modified variant of the IMSE criterion, which weights the predictive standard deviation within the integral over the design space with a Gaussian weighting function around the specified target value and thus achieves a greater density of sample points in the associated regions of the design space.

The aforementioned variance-based and cross-validation-based adaptive sampling methods have been evaluated in numerous publications for various black-box functions [15, 16, 21, 23, 24, 36, 71]. However, real-valued functions are always assumed, some of which may even have discontinuities (e.g., jumps) [27]. The only publication [69] (to the best of our knowledge) that considers the function value of the black-box function sets a known target value of the black-box around which the function is to be approximated with greater accuracy. For the problem considered in this chapter, neither a target value is relevant (nor would it be known), nor is the maximum value of the black-box function known. Rather, the model accuracy should increase with increasing value of the black-box function, and it should be taken into account that the black-box function is a non-negative continuous function with a limited rate of change. This problem is not addressed by any of the existing methods.

4.1 Problem Formulation

We consider a Lipschitz-continuous, non-negative, and bounded black-box function $f : \mathcal{X} \to \mathbb{R}$ which takes values in the interval $[0, f_{\max}]$, where $f_{\max} = \max_{\underline{x} \in \mathcal{X}} f(\underline{x})$ is a positive and finite real number. The black-box function is to be approximated within the bounded domain $\mathcal{X} \subset \mathbb{R}^n$ (the design space), where n is the number of dimensions. The approximation is based on noisy samples $y_l \in \mathbb{R}$, taken at sample points $\underline{x}_l \in \mathcal{X}$ for $l = 1, \dots, m$, where m is the number of existing samples. The samples

$$y_l = f(\underline{x}_l) + \epsilon_l \tag{4.1}$$

are given by the value of the black-box function $f(\underline{x})$ at sample points $\underline{x}_l$ disturbed by additive, zero-mean noise modeled by a Gaussian noise term $\epsilon_l \sim \mathcal{N}(0, \sigma_l^2)$. Furthermore, the approximation should be “more accurate" (according to a given metric) where the black-box function $f(\underline{x})$ takes large function values and should hence be sampled more densely in the corresponding regions of the design space. Given the existing design $\mathcal{D}_m = \{\underline{x}_l, y_l\}_{l=1}^m$, the task is to find the optimal choice of $\underline{x}_{m+1}$.

4.2 Weighted Explorative Sampling Criteria

In this section, we introduce our concept of *weighted explorative sampling criteria (WESC)* – a novel way of defining weighted adaptive sampling criteria, tailored to the problem statement in Sec. 4.1. For this, we first define the class of *explorative sampling criteria (ESC)* in Sec. 4.2.1 and show that most existing distance-based and variance-based sampling criteria can easily be transformed into such. Building on ESC we then present the novel concept of WESC in Sec. 4.2.2.

4.2.1 Explorative Sampling Criteria

First, we define the class of explorative sampling criterion (ESC), which will serve as a basis for the formulation of WESC in Sec. 4.2.2.

Definition 4.2.1. *Consider a sampling criterion $\varphi : \mathcal{X} \to \mathbb{R}$ on a design space $\mathcal{X} \subset \mathbb{R}^n$, where n is the number of dimensions. A sampling criterion $\varphi(\underline{x})$ is an explorative sampling criterion if it satisfies the following conditions:*

1. *It is an adaptive or a sequential sampling criterion.*
2. *It is a non-negative and bounded function on the design space $\mathcal{X}$ which takes values in the interval $[0, \varphi_{max}]$, where φ_{max} is the maximum value $\varphi(\underline{x})$ takes on the design space ($\varphi_{max} = \arg\max_{\underline{x} \in \mathcal{X}} \varphi(\underline{x})$).*
3. *It is to be maximized for determination of the next sample point, hence $\underline{x}_{m+1} = \arg\max_{\underline{x} \in \mathcal{X}} \varphi(\underline{x})$.*
4. *It exhibits at least one of the following properties*
   (a) *Uncertainty minimization: It effectively minimizes the uncertainty of the metamodel, measured by an uncertainty metric, such as the maximum predictive variance of the metamodel.*
   (b) *Space-filling: It ensures an even coverage of the design space according to a space-filling metric, such as the maximin distance.*

Def. 2.1.1 summarizes all space-filling sequential and uncertainty-based adaptive criteria that have certain properties to the class of ESC. Essential properties are that they are to be maximized, thus the next sample point is the maximizer of the criterion in the design space, and that the criterion is non-negative. The criteria maximin from Sec. 2.2.2 and mod. maximin from Sec. 3.1.3 as well as MMSE and MV from Sec. 2.2.3 are examples of ESC. Most distance-based space-filling sequential criteria and variance-based adaptive criteria can easily be transformed into ESC by negation and / or shifting. E.g., the sequential MD and RMSD criteria from

Sec. 3.1 have a positive and bounded range and must be minimized to determine the next sample point. By negating (the negated criterion is now to be maximized) and subtracting the minimum that the negated criterion takes in the design space (the shifted criterion has a positive value range), they can easily be transformed into an ESC. As an example, we state the transformation of the sequential MD criterion

$$\varphi^{\text{MD-ESC}}(\underline{x}) = -\varphi^{\text{MD}}(\underline{x}) + \max_{\underline{x}\in\mathcal{X}} \varphi^{\text{MD}}(\underline{x}) \ , \tag{4.2}$$

where $\max_{\underline{x}\in\mathcal{X}} \varphi^{\text{MD}}(\underline{x}) = -\min_{\underline{x}\in\mathcal{X}}(-\varphi^{\text{MD}}(\underline{x}))$ was used to turn the subtraction of the negated minimum into an addition of the maximum.

4.2.2 Weighted Explorative Sampling Criteria

The basic idea behind the WESC is the weighting of an ESC $\varphi^{\text{ESC}}(\underline{x})$ with the function value $f(\underline{x})$ of the black-box function. By combining an ESC with the (non-negative) function value of the black-box function, the explorative samples are placed in favor of high function values of the black-box function, thus addressing the problem stated in Sec. 4.1. While the ESC guarantees exploration and avoids clustering of samples, the weighting by the black-box function ensures more sample points in regions of high function values.

However, when formulating the according WESC $f(\underline{x}) \cdot \varphi^{\text{ESC}}(\underline{x})$, we encounter several challenges. First, the black-box function is not known, so we have to approximate it by the metamodel, resulting in $\hat{f}(\underline{x}) \cdot \varphi^{\text{ESC}}(\underline{x})$. The metamodel has large uncertainties in unexplored regions of the design space and in the case of a GP with a prior mean of zero (which is commonly used as metamodel in global metamodeling [15, 16] and environmental sensing [28, 46]), the function value of the metamodel vanishes in these regions of the design space. Consequently, this approach may result in certain regions remaining unexplored, potentially overlooking regions of interest. In order to ensure exploration of unseen regions, we clamp the function value of the metamodel to the lower bound $\alpha \cdot \hat{f}_{\max}$, where $\alpha \in [0, 1]$ is the *exploration parameter* and $\hat{f}_{\max} = \max_{\underline{x}\in\mathcal{X}} \hat{f}(\underline{x})$ is the maximum of the metamodel in the design space. Finally, we define the WESC

$$\varphi^{\text{WESC}}(\underline{x}) = \max(\alpha \hat{f}_{\max}, \hat{f}(\underline{x})) \cdot \varphi^{\text{ESC}}(\underline{x}) \tag{4.3}$$

as the product of the maximum of the lower bound and the function value of the metamodel (the weighting function) and the ESC. Note that the WESC is again an adaptive sampling criterion that is to be maximized for determination of the next sample point, hence $\underline{x}_{m+1} = \arg\max_{\underline{x}\in\mathcal{X}} \varphi^{\text{WESC}}(\underline{x})$. Fig. 4.1 shows the effect of the exploration parameter α on the resulting WESC.

Next, we present two examples of WESC. The first criterion is based on the MV criterion from Sec. 2.2.3 while the second criterion is based on the mod. maximin criterion from Sec. 3.1.3.

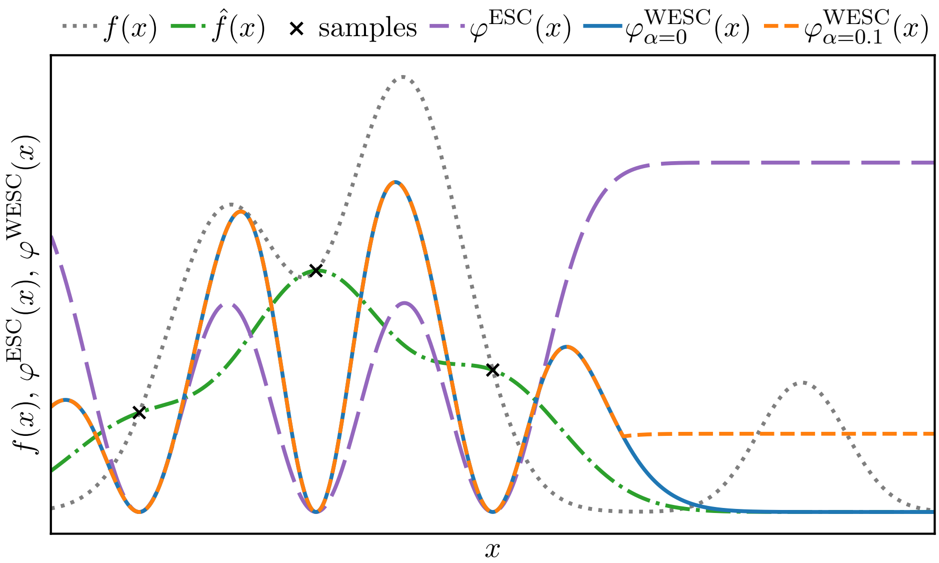

Figure 4.1: The effect of the exploration parameter α of the WESC is illustrated in a one-dimensional example. Without enforced exploration ($\alpha > 0$), regions where the metamodel $\hat{f}(\underline{x})$ vanishes but the black-box function $f(x)$ takes significant values are likely to be missed.

Weighted Maximum Variance Criterion The *weighted maximum variance (WMV) criterion* [31]

$$\varphi^{\mathrm{WMV}}(\underline{x}) = \max(\alpha \hat{f}_{\max}, \hat{f}(\underline{x})) \cdot \hat{\sigma}^2(\underline{x}) \tag{4.4}$$

uses the MV criterion as ESC and weights it by the clipped function value of the metamodel, as described in (4.3). Note we assume the metamodel to be a GP (analogous to the definition of the MV criterion in Sec. 2.2.3) with the predictive variance $\hat{\sigma}^2(\underline{x})$. By weighting the predictive variance, the adaptive sample points are placed in the design space where the model uncertainty and the metamodel take on high values.

Weighted Square Modified Maximin Criterion The WMV criterion requires the use of a GP as a metamodel. By using the distance-based mod. maximin criterion from Sec. 3.1.3 as ESC, the *weighted square modified maximin (WMM) criterion*

$$\varphi^{\mathrm{WMM}}(\underline{x}) = \max(\alpha \hat{f}_{\max}, \hat{f}(\underline{x})) \cdot \min\Big(\min_{\underline{x}_i \in X} ||\underline{x} - \underline{x}_i||, 2 \min_{\underline{b} \in \partial \mathcal{X}} ||\underline{x} - \underline{b}||\Big)^2 \tag{4.5}$$

is independent of the metamodel used. Utilizing the WMM criterion allows for the placement of sample points near the boundary of the design space, even though the mod. maximin design does not distribute sample points in these regions. This is because the weighting of the mod. maximin criterion by the clipped function value of the metamodel significantly influences the maximum of the WMM criterion and thus the placement of the sample points. However, it is not possible for sample points to be placed exactly on the boundary, as the mod. maximin criterion vanishes there.

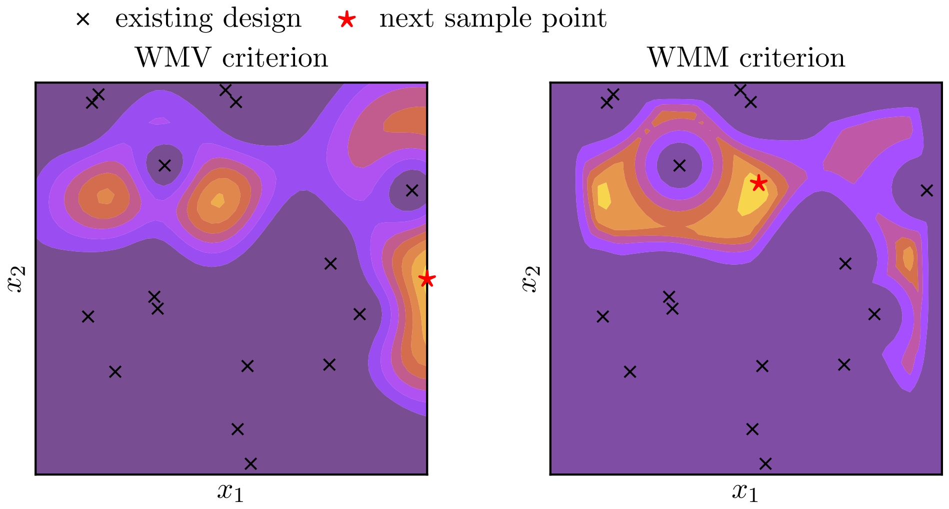

Figure 4.2: Existing sample points and next adaptive sample point on top of the adaptive WMV criterion (left) and WMM criterion (right), both computed for an exploration parameter $\alpha = 0.025$. The next sample point is located at the maximum of the respective criterion.

In contrast, the prediction variance of the GP increases at the boundary of the design space, since there are no sample points beyond the boundary, thereby increasing model uncertainty. Accordingly, the WMV criterion exhibits an inherent tendency to place sample points on the boundary of the design space if the function value of the metamodel is significant in these regions. Example evaluations of the two criteria WMV and WMM are shown in Fig. 4.2.

4.3 Evaluation

In this section, the two adaptive criteria WMV and WMM as representatives of the class of WESC are compared against the adaptive criteria CVV and the MV criterion from literature (defined in Sec. 2.2.3). We show that the WESC are better suited for metamodeling of non-negative, Lipschitz-continuous black-box functions (detailed in the problem statement in Sec. 4.1) than the literature methods. First, the experimental setup is described in Sec. 4.3.1, followed by an extensive numerical comparison of the WESC with the literature methods in Sec. 4.3.2.

4.3.1 Experimental Setup

For evaluation, the adaptive WMV, WMM, CVV, and MV designs were computed on the design space $\mathcal{X} = [0, 1] \times [0, 1]$ using the adaptive sampling procedure described in Fig. 2.19. LHS of 16 samples, computed as described in Sec. 3.2.1, was for each criterion used as space-filling initial design to warm-start the adaptive sampling algorithm.

All designs were computed for different numbers of samples and used to evaluate $N^{\text{runs}} = 100$ randomly generated black-box functions f_q, $q \in \{1, \ldots, N^{\text{runs}}\}$. Each of the black-box functions was modeled as weighted sum of ten squared exponential functions

$$f_q(\underline{x}) = \sum_{t=1}^{10} w_t \cdot e^{-\frac{1}{2}(\underline{x}-\bar{\underline{x}}_t)^\top \boldsymbol{S}_t^{-1}(\underline{x}-\bar{\underline{x}}_t)} \ , \tag{4.6}$$

where $w_t \in [0, 1]$ are uniformly distributed random weights, the locations of the squared exponentials $\bar{\underline{x}}_t \in \mathcal{X}$ are uniformly distributed on the design space $\mathcal{X}$, and $\boldsymbol{S}_t$ are covariance matrices (positive semidefinite, and symmetric matrices with non-negative diagonal elements). The covariance matrices were generated by rotating a diagonal matrix with uniformly distributed random entries between $[0.001, 0.01]$ by a uniformly distributed random angle using a rotation matrix. This resulted in smooth distributions that could already be well approximated with 100 samples. Sampling noise was modeled by adding a random number drawn from the distribution $\mathcal{N}(0, 0.01^2)$ to the value of the black-box function when evaluating the black-box function at the adaptive sample points according to (4.1).

A GP with an SE kernel was used as a metamodel and the hyperparameters of the kernel were obtained by optimizing the log marginal likelihood as described in Sec. 2.1. To evaluate the accuracy of the metamodel, the design space $\mathcal{X}$ was discretized into $\mathcal{X}^{\text{eval}} \subset \mathcal{X}$ by means of a 100 by 100 mod. RGS, resulting in $N^{\text{eval}} = 10,000$ evaluation points $\underline{x}_\eta \in \mathcal{X}^{\text{eval}}$, $\eta = 1, \ldots, N^{\text{eval}}$. Based on the evaluation points, the normalized root-mean-square error

$$\text{NRMSE} = \frac{1}{f_{\max} - f_{\min}} \sqrt{\frac{1}{N^{\text{eval}}} \sum_{\eta=1}^{N^{\text{eval}}} \left(\hat{f}(\underline{x}_\eta) - f(\underline{x}_\eta)\right)^2} \tag{4.7}$$

and the normalized weighted root-mean-square error

$$\text{NWRMSE} = \frac{1}{f_{\max} - f_{\min}} \sqrt{\frac{\frac{1}{N^{\text{eval}}} \sum_{\eta=1}^{N^{\text{eval}}} f(\underline{x}_\eta) \left(\hat{f}(\underline{x}_\eta) - f(\underline{x}_\eta)\right)^2}{\sum_{\eta=1}^{N^{\text{eval}}} f(\underline{x}_\eta)}} \tag{4.8}$$

were computed as a measure for the overall error between the black-box function and the metamodel, where $f_{\max} = \max\limits_{\eta} f(\underline{x}_\eta)$ and $f_{\min} = \min\limits_{\eta} f(\underline{x}_\eta)$. The NWRMSE weights the squared deviations of the metamodel from the black-box function and was introduced specifically to evaluate the WESC. In this way, it can be evaluated whether the model accuracy using the WESC as adaptive sampling criteria results in greater model accuracy in regions of high black-box function values compared to the model accuracy achieved with the criteria from the literature. This is possible as the black-box function is known when computing the error measures in the evaluation (while it is assumed unknown for the adaptive sampling algorithm when selecting the adaptive sample points).

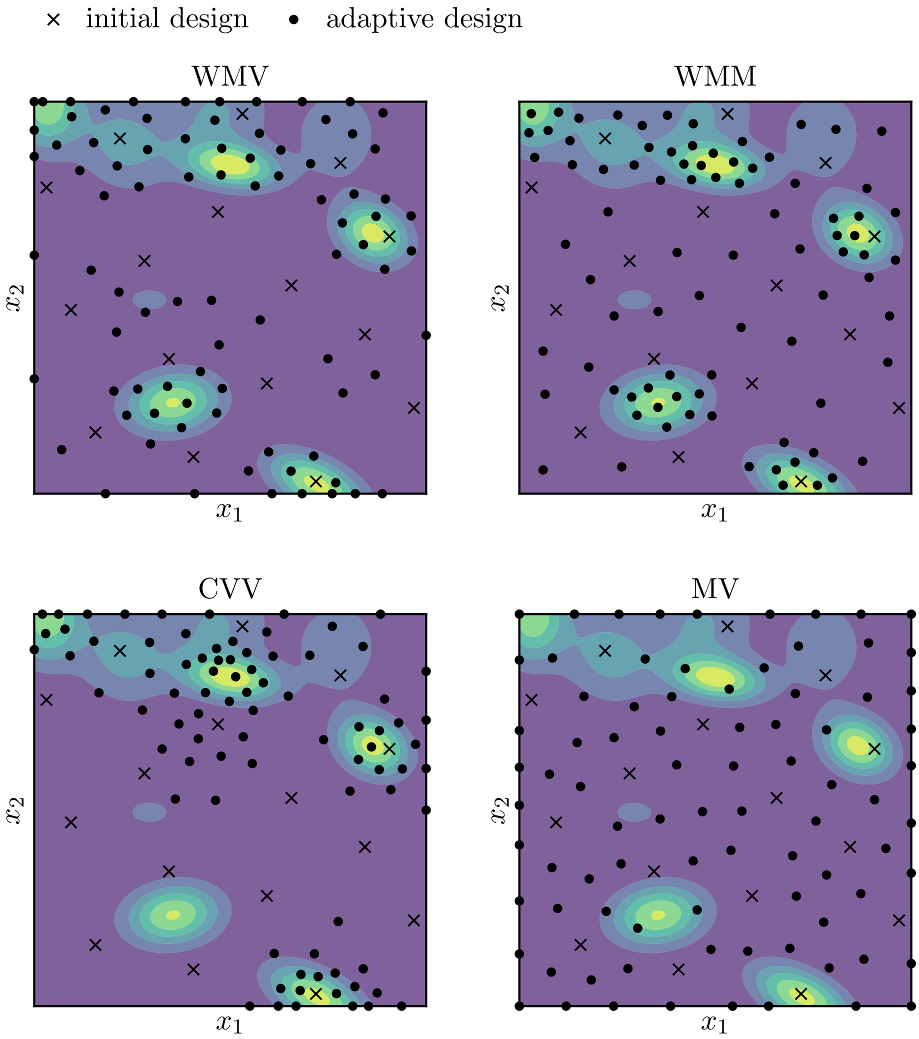

Figure 4.3: Space-filling initial design (16 samples) and adaptive design (84 samples) on top of a black-box function that was randomly generated as outlined in Sec. 4.3.1. For all criteria the adaptive sampling algorithm was warm-started with the initial design. The top row displays the adaptive designs generated based on our novel criteria WMV and WMM as examples of WESC. The bottom row shows the designs based on the criteria CVV and MV from literature. The exploration parameter was set to $\alpha = 0.025$ for the WMV criterion and $\alpha = 0.1$ for the WMM criterion.

4.3.2 Numerical Comparison to Literature Methods

In this section, the two criteria WMV and WMM as representatives of the class of WESC are compared against the criteria CVV and MV from literature (defined in Sec. 2.2.3).

Fig. 4.3 shows examples of the corresponding adaptive designs for a randomly generated black-box function as described in Sec. 4.3.1. It can be clearly seen that the WESC place more sample points in regions of high black-box function values than the literature methods. The MV criterion distributes the sample points more evenly in all regions of the design space but does not take into account the local complexity of the black-box function. Consequently, the regions with significant function values or rates of change are not approximated with greater accuracy than the remaining regions of the design space. Unlike the MV criterion, many sample points are placed in regions of high complexity when using CVV. However, this often leads to the situation that not all regions of interest are identified and sampled, as illustrated in Fig. 4.3.

This is also reflected in the results of the numerical simulations. The WESC provide a better approximation of the black-box function than CVV and MV for each number of samples, with CVV consistently performing better than MV. This applies to both the mean NRMSE and NWRMSE. WMV and WMM deliver comparable results, with WMM achieving a slightly lower mean NRMSE and WMV a slightly lower mean NWRMSE. The gap between the WESC and CVV is greater for the mean WNRMSE than for the mean NRMSE. This can be explained by the fact that the WESC generate a greater density of sample points in regions of high black-box function values and thus achieve better accuracy of the metamodel in these regions, as shown in Fig. 4.3.

4.4 Summary

In this chapter we introduced our new concept of weighted explorative criteria (WESC), specifically designed for metamodeling of non-negative, Lipschitz-continuous environmental phenomena (black-box functions), such as concentration distributions. For this purpose, we first defined the class of explorative criteria (ESC), which comprises all sequential and adaptive criteria that effectively generate space-filling or uncertainty-minimizing samples, are non-negative, and are to be maximized for determination of the next sample point. We showed that many space-filling sequential and adaptive sampling criteria already have these properties and how other criteria can easily be transformed to an ESC by negation and / or shifting. Building on ESC, we then introduced the class of WESC that weight ESC by the lower-bounded function value of the metamodel to achieve enhanced model accuracy in regions

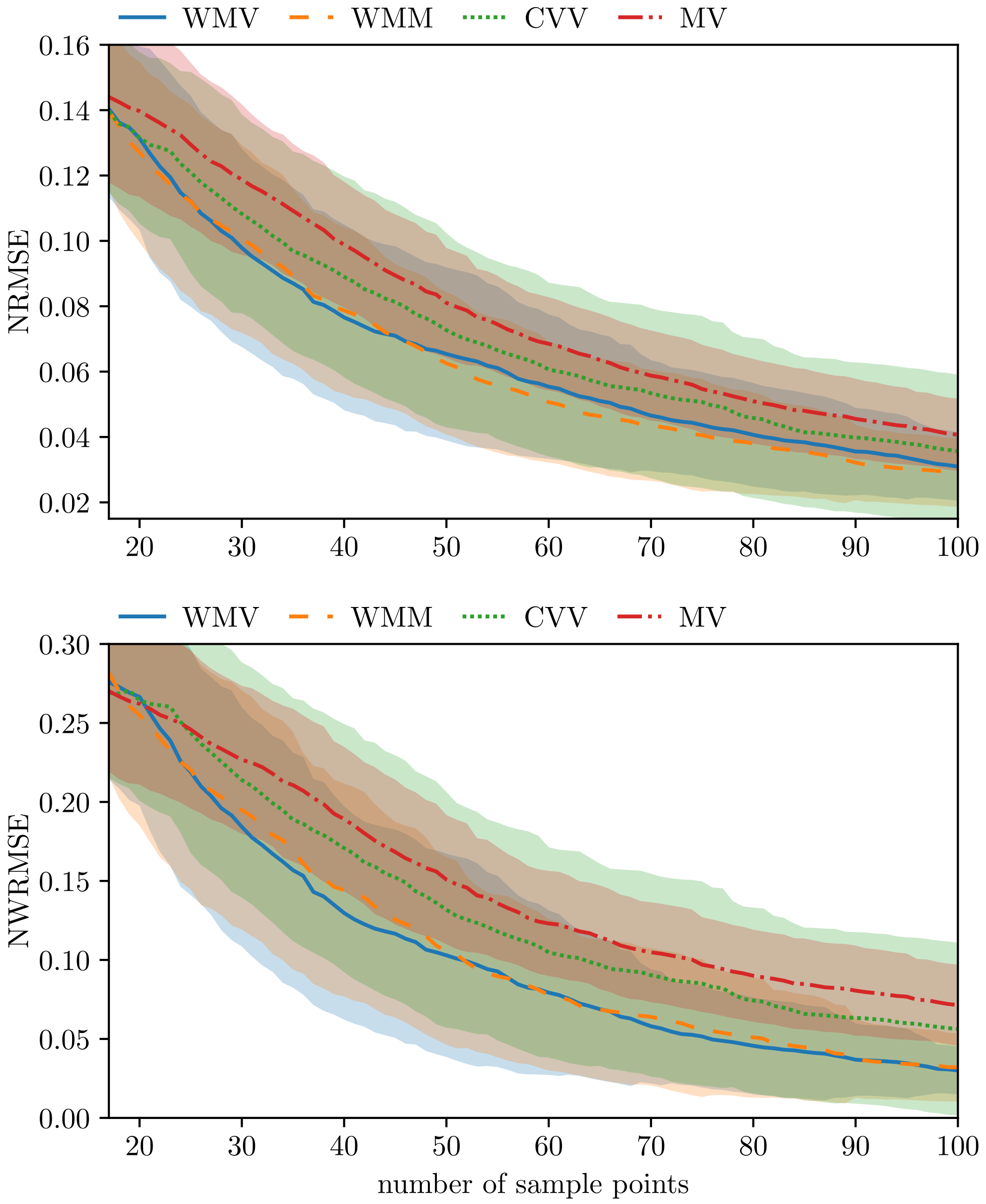

Figure 4.4: The mean (lines) and standard deviation (shaded area) of the NRMSE (top) and NWRMSE (bottom) over $N^{\mathrm{runs}} = 100$ randomly generated black-box functions vs the number of sample points.

of the design space where the black-box function takes high values, compared to regions where it takes low values. This addresses the problem of sensing non-negative, continuous phenomena with a limited rate of change (e.g., concentration distributions) as detailed in Sec. 4.1. We show that WESC vastly outperform the literature methods in the numerical evaluation by taking into account the properties of the black-box function.

CHAPTER 5

Cost-aware Adaptive Sampling for Environmental Sensing

In this chapter, we consider location-dependent (local) and travel costs that are important in the application of sequential and adaptive sampling methods for environmental sensing using mobile robots. Mobile robotic systems are widely used for various tasks in the field of environmental sensing and monitoring. Examples are the measurement of air quality in urban regions [86] or the detection of gas leaks indoors [9] using mobile robotic systems with gas sensors. Mobile robotic systems have also been equipped with an electronic nose (miniature sensor array) for soil sensing in agriculture in order to measure fertilization and moisture [10] or have been used for soil health sensing [11]. Others have used a stationary drilling robot for subsurface soil exploration [87]. Algorithms have been developed for phytoplankton peak capturing using autonomous underwater vehicles (AUVs) [12, 13]. Furthermore, an AUV has been used as autonomous water sampler for oil spill response [88].

The aforementioned and many other comparable applications use real-time capable sensors that record data along the robot's path while the robot is moving. This requirement is not met if the mobile robot has to interrupt its motion to carry out expensive sampling, as in the scenario considered in this thesis. One example is the ROBDEKON project [14], in which an autonomous construction machine is used to perform expensive dynamic probing on a contaminated site, on the basis of which the distribution of pollutants in the soil is then to be reconstructed. The sampling costs (time, energy) depend on the subsoil (different compositions (e.g., sand, clay, gravel), degree of compaction) and the accessibility (e.g., vegetation, slopes) of the sample points, resulting in location-dependent sampling costs.

In the previous chapters, we considered sampling methods that trade-off between exploration and exploitation in order to maximize the information content per sample. However, successive sample points often lie in entirely different regions of the design space. In the application, this would result in the mobile robot (in the case of ROBDEKON, a heavy construction machine) having to travel long distances between the sample points, which would incur high travel costs. Therefore, to achieve optimal reconstruction of an environmental phenomenon within a defined (finite) budget, it is essential to account for location-dependent sampling costs and travel costs in addition to the trade-off between exploration and exploitation anchored in the sampling criterion. An exclusive focus on the sampling criterion may provide informative samples but may lead to increased costs. Increased costs could deplete the budget more quickly, thereby reducing the number of possible samples and thus negatively impacting the quality of the reconstruction. Conversely, excessive cost avoidance can lead to insufficient information being obtained per sample, resulting in poor reconstruction despite a large number of samples.

Accordingly, compared to the previous chapters, competing objectives must be compromised. This can be achieved by means of multi-objective optimization, the required basics of which are introduced in Sec. 5.1. In the further course of this chapter, Sec. 5.3 and Sec. 5.4 define our novel methods for considering travel costs and location-dependent (local) sampling costs, followed by Sec. 5.5 which introduces a novel means of combining both types of cost concentration in a way that leverages their strengths. Each section of this chapter includes an excessive simulative evaluation of the methods, with the experimental setup detailed in Sec. 5.2. Finally, Sec. 5.6 summarizes this chapter.

5.1 Pareto-optimal Multi-objective Optimization

A *multi-objective optimization problem (MOP)* is a type of optimization problem that involves more than one objective function to be optimized simultaneously [89–91]. If the number of objective functions to be optimized gets large, this is usually referred to as a *many-objective optimization problem* [92, 93]. These objectives often conflict with each other, making the task of finding the best solution more complex than single-objective optimization. Mathematically, we define a vector of $2 \leq Q < \infty$ objectives

$$\underline{J}(\underline{x}) = [J_1(\underline{x}), \ldots, J_Q(\underline{x})]^\top \in \mathcal{J} \subset \mathbb{R}^Q \tag{5.1}$$

that are to be minimized

$$\min_{\underline{x}} \underline{J}(\underline{x}) \text{ s.t. } \underline{x} \in \mathcal{X} \ ,$$

where $J_i(\underline{x})$ are the competing objective functions, $\underline{x} \in \mathcal{X}$ is a vector of n decision variables, $\mathcal{X} \subset \mathbb{R}^n$ is the decision space, and $\mathcal{J}$ is the image of the mapping $\underline{J} : \mathcal{X} \to \mathcal{J}$ [89–91]. The decision space is a nonempty and compact set that contains all feasible decision vectors [89], which corresponds to the design space in the context of this thesis.

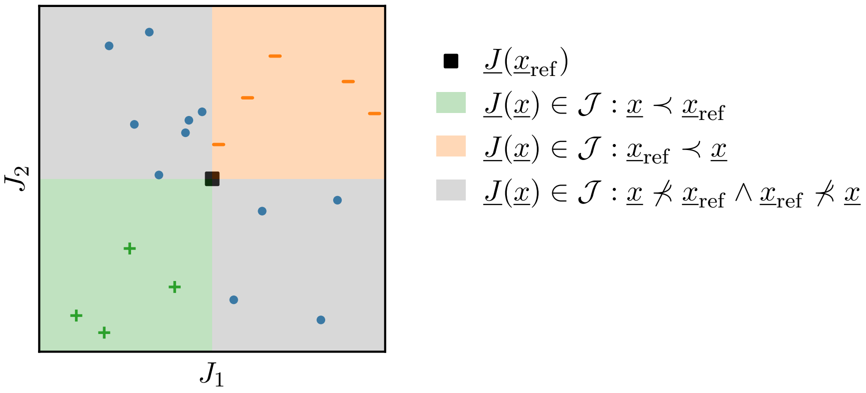

Figure 5.1: Dominance relationships w.r.t. a reference point (black square) illustrated in a two-dimensional objective space. The green area contains the dominating points, the orange area the dominated points and the gray area contains the points that cannot be compared by dominance relationship.

In general, there is no unique optimal solution for a MOP [91]. However, we can find *Pareto-optimal* (or *Pareto-efficient*) solutions that can be defined using dominance relationships. A solution $\underline{x} \in \mathcal{X}$ is said to dominate another solution $\underline{x}' \in \mathcal{X}$ ($\underline{x} \prec \underline{x}'$) if and only if $J_i(\underline{x}) \leq J_i(\underline{x}')$ $\forall i \in 1, 2, \ldots, Q$ and $\exists j \in 1, 2, \ldots, Q$ $J_j(\underline{x}) < J_j(\underline{x}')$ [89, 91]. If for a solution $\underline{x} \in \mathcal{X}$, there does not exist a solution $\underline{x}' \in \mathcal{X}$ that dominates x, then x is Pareto-optimal. The *Pareto set* $\mathcal{X}^* = \{\underline{x} \in \mathcal{X} : \nexists \underline{x}' \in \mathcal{X}, \underline{x}' \prec \underline{x}\}$ is the set of all Pareto-optimal solutions. Fig. 5.1 illustrates the dominance relationships in the objective space.

When applying the mapping $\underline{J}(\cdot)$ to the Pareto set, we obtain the *Pareto front* $\mathcal{J}^* = \{\underline{J}(\underline{x}) \in \mathcal{J} : \underline{x} \in \mathcal{X}^*\}$ [92, 93]. The Pareto front is the set of all objective vectors whose components cannot be further optimized without worsening at least one other component [89], as shown in Fig. 5.2.

Note that there exist solutions that do not dominate each other ($\underline{x} \not\prec \underline{x}'$ and $\underline{x}' \not\prec \underline{x}$) and can hence not be compared by dominance relation [94].

Two characteristic points in the objective space are the *ideal point*

$$\underline{J}^{\text{ideal}} = [\min_{\underline{x} \in \mathcal{X}} J_1(\underline{x}) \quad \ldots \quad \min_{\underline{x} \in \mathcal{X}} J_Q(\underline{x})]^\top \tag{5.2}$$

representing the lower bound of the non-dominated points and the *nadir point*

$$\underline{J}^{\text{nadir}} = [\max_{\underline{x} \in \mathcal{X}} J_1(\underline{x}) \quad \ldots \quad \max_{\underline{x} \in \mathcal{X}} J_Q(\underline{x})]^\top \tag{5.3}$$

representing the upper bound of the dominated points [93, 95].

A common way to trade-off between multiple competing objectives is to transform the MOP into a *single optimization problem (SOP)*

$$\min_{\underline{x}} s(\underline{J}(\underline{x})) \text{ s.t. } \underline{x} \in \mathcal{X} \ ,$$

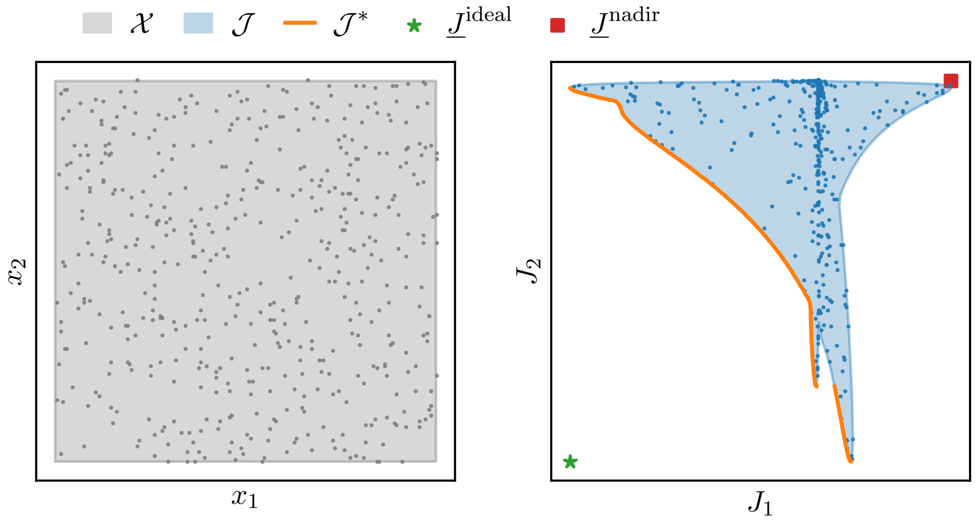

Figure 5.2: Decision space $\mathcal{X}$ (gray area) and scattered sample points on the left and the objective space on the right. The objective space contains the image $\mathcal{J}$ (blue area) of the mapping $\underline{J} : \mathcal{X} \to \mathcal{J}$ containing the mapped scattered sample points. In addition, the Pareto front (orange line) as well as the ideal and nadir points are shown.

using a real-valued *scalarization function* $s : \mathcal{X} \to \mathcal{S}$, where $\mathcal{S} \subset \mathbb{R}$ is the image of the scalarization function [96]. Most scalarization functions involve parameters to control the trade-off [89].

One of the most widely used scalarization functions is the *weighted sum*

$$s^{\mathrm{ws}}(\underline{J}(\underline{x})) = \sum_{i=1}^{Q} w_i J_i(\underline{x}) \tag{5.4}$$

that linearly combines the objective functions [96], where w_i are weights. However, the weighted sum has the major drawback that it cannot find the Pareto-optimal solutions in the non-convex parts of the Pareto front [79, 96]. The *Chebyshev* scalarization function

$$s^{\mathrm{cheby}}(\underline{J}(\underline{x})) = \max_{i \in \{1,\ldots,Q\}} (w_i |J_i(\underline{x}) - \underline{J}^{\mathrm{ideal}}[i]|) \tag{5.5}$$

is a simple means to overcoming this limitation [79, 94, 96]. Instead of summing the objective functions it minimizes the maximum weighted distance to the ideal point in the objective space. Augmented versions of the Chebyshev scalarization function exist that avoid weakly Pareto-optimal solutions [95]. Refer to e.g., [96] for an overview on scalarization functions.

5.2 Experimental Setup

This section describes the experimental setup used to evaluate the sampling methods presented in Sec. 5.3 – Sec. 5.5.

As in Sec. 3.2 and Sec. 4.3, the sampling methods were evaluated on the design space $\mathcal{X} = [0, 1] \times [0, 1]$. The adaptive sampling methods were warm-started with an initial design of 16 space-filling sample points. Since multiple initial designs were evaluated, the method used for generating the space-filling design is specified in the according section. However, if LHS was used, it was generated as described in Sec. 3.2.1.

All designs were computed for different numbers of samples and applied to $N^{\text{runs}} = 100$ randomly generated black-box functions $f_q(\underline{x})$, $q \in \{1, \dots, N^{\text{runs}}\}$. Each of the black-box functions was modeled as weighted sum of ten squared exponential functions

$$f_q(\underline{x}) = \sum_{t=1}^{10} w_t \cdot e^{-\frac{1}{2}(\underline{x}-\bar{\underline{x}}_t)^\top \boldsymbol{S}_t^{-1}(\underline{x}-\bar{\underline{x}}_t)} \ , \tag{5.6}$$

where $w_t \in [0, 1]$ are uniformly distributed random weights, the locations of the squared exponentials $\bar{\underline{x}}_t \in \mathcal{X}$ are uniformly distributed on the design space $\mathcal{X}$, and $\boldsymbol{S}_t$ are covariance matrices (positive semidefinite, and symmetric matrices with non-negative diagonal elements). The covariance matrices were generated by rotating a diagonal matrix with uniformly distributed random entries between $[0.001, 0.01]$ by a uniformly distributed random angle using a rotation matrix. This resulted in smooth distributions that could already be well approximated with 100 samples. Sampling noise was modeled by adding a random number drawn from the distribution $\mathcal{N}(0, 0.01^2)$ to the value of the black-box function when evaluating the black-box function at the adaptive sample points as defined in (4.1).

The location-dependent sampling costs $c_q^x(\underline{x})$ were modeled analogously to the black-box functions as a sum of 25 squared exponential functions, whose covariance matrices were generated by rotating a diagonal matrix with uniformly distributed random entries between $[0.0005, 0.005]$ by a uniformly distributed random angle using a rotation matrix. An offset of 0.1 has been added to all cost functions to account for a positive minimum sampling cost.

A GP with an SE kernel was used as a metamodel (and as a cost model for approximation of the location-dependent sampling costs, in case needed) and the hyperparameters of the kernel were obtained by optimizing the log marginal likelihood as described in Sec. 2.1. The metamodel was initialized with a zero mean prior and the cost model was initialized with a constant prior mean of 0.1 to account for the minimum location-dependent sampling cost. To evaluate the accuracy of the metamodel, the design space $\mathcal{X}$ was discretized into $\mathcal{X}^{\text{eval}} \subset \mathcal{X}$ by means of a 100 by 100 mod. RGS, resulting in $N^{\text{eval}} = 10,000$ evaluation points $\underline{x}_\eta \in \mathcal{X}^{\text{eval}}$, $\eta = 1, \dots, N^{\text{eval}}$. Based on the evaluation points, the normalized root-mean-square error

$$\text{NRMSE}_q = \frac{1}{f_{q,\max} - f_{q,\min}} \sqrt{\frac{1}{N^{\text{eval}}} \sum_{\eta=1}^{N^{\text{eval}}} \left(\hat{f}_q(\underline{x}_\eta) - f_q(\underline{x}_\eta)\right)^2} \tag{5.7}$$

was computed as a measure for the overall error between the black-box function and the metamodel, where $f_{q,\max}{=}\max\limits_{\eta} f_q(\underline{x}_\eta)$ and $f_{q,\min}{=}\min\limits_{\eta} f_q(\underline{x}_\eta)$.

The evaluation points were also used to numerically approximate the integrals needed to compute mean location-dependent sampling cost

$$\bar{c}_q^x = \frac{\int_{\underline{x}\in\mathcal{X}} c_q^x(\underline{x}) d\underline{x}}{\int_{\underline{x}\in\mathcal{X}} 1 d\underline{x}} \approx \frac{1}{N^{\text{eval}}} \sum_{\eta=1}^{N^{\text{eval}}} c_q^x(\underline{x}_\eta) \tag{5.8}$$

on the design space by means of Monte Carlo approximation. To evaluate the cost-effectiveness of the sampling methods, we considered the model accuracy as a function of the budget used. Since the cost functions were randomly generated and could therefore result in different average sampling costs, we defined a multiple of the average sampling cost as a budget

$$b_q^x = M \cdot \bar{c}_q^x \tag{5.9}$$

for each of the cost functions, where $M = 100$. Note that the budget b_q^x corresponds to the expected sampling costs when selecting M random sample points.

To illustrate how the costs per sample evolve for each of the sampling methods, we defined the *normalized mean cost per sample (NMCPS)*

$$\text{NMCPS}_q^x = \frac{1}{m \cdot \bar{c}_q^x} \underbrace{\sum_{i=1}^{m} c_q^x(\underline{x}_i)}_{c_{\text{cum},q}^x} = \frac{c_{\text{cum},q}^x}{m \cdot \bar{c}_q^x} \;, \tag{5.10}$$

which describes the average cost generated up to the number of m samples, normalized by dividing them by the expected cost $\bar{c}_q^x$ for the respective cost function. To analyze the model accuracy achieved per budget spent, we defined the *normalized cumulative cost (NCC)*

$$\text{NCC}_q^x = \frac{1}{M \cdot \bar{c}_q^x} \sum_{i=1}^{m} c_q^x(\underline{x}_i) = \frac{c_{\text{cum},q}^x}{b_q^x} = \frac{m}{M} \cdot \text{NMCPS}_q^x \tag{5.11}$$

which takes values in the interval $[0, 1]$. It can be interpreted as the normalized budget already spent and can be computed from NMCPS_q^x (and vice versa).

Without loss of generality, we defined the travel costs

$$c^{\Delta x}(\underline{x}_m, \underline{x}_{m+1}) = ||\underline{x}_{m+1} - \underline{x}_m|| \tag{5.12}$$

for the robot to travel from its current position $\underline{x}_m$ to the next sample point $\underline{x}_{m+1}$ as the Euclidean distance between the two points.

The measures NRMSE_q, NMCPS_q^x, and NCC_q^x were defined for a single run q of the simulation and hence a unique set of black-box function, cost function or metamodel. In order to evaluate these statistically, their mean values NRMSE, NMCPS^x, and NCC^x (and partly standard deviations) over all q were considered in the figures.

In order to compare the performance of a sampling method to a reference sampling method, we first defined the *mean cumulative travel cost (MCTC)*

$$\mathrm{MCTC} = \frac{1}{Q}\sum_{q=1}^{Q} \underbrace{\sum_{i=1}^{m-1} c^{\Delta x}(\underline{x}_{q,m}, \underline{x}_{q,m+1})}_{c^{\Delta x}_{\mathrm{cum},q}} = \frac{1}{Q}\sum_{q=1}^{Q} c^{\Delta x}_{\mathrm{cum},q} \tag{5.13}$$

as the mean cumulative travel cost over all Q runs of the simulation, where $\underline{x}_{q,m}$ is the m-th sample point in run q. Then, for each number of samples m, the *relative mean cumulative travel cost saving (Rel. MCTCS)*

$$\mathrm{Rel.\,MCTCS} = \frac{\mathrm{MCTC}^{\mathrm{ref}} - \mathrm{MCTC}}{\mathrm{MCTC}^{\mathrm{ref}}} = 1 - \frac{\mathrm{MCTC}}{\mathrm{MCTC}^{\mathrm{ref}}} \tag{5.14}$$

was defined as a measure of the relative travel cost savings of a sampling method over a reference sampling method, indicated by the superscript 'ref'.

For application of the Chebyshev scalarization function defined in (5.5), its parameters were applied to the normalized objectives

$$\underline{J}^{\mathrm{norm}}(\underline{x}) = \left[\frac{J_1(\underline{x}) - \underline{J}^{\mathrm{ideal}}[1]}{\underline{J}^{\mathrm{nadir}}[1] - \underline{J}^{\mathrm{ideal}}[1]} \quad \cdots \quad \frac{J_Q(\underline{x}) - \underline{J}^{\mathrm{ideal}}[Q]}{\underline{J}^{\mathrm{nadir}}[Q] - \underline{J}^{\mathrm{ideal}}[Q]}\right]^{\top} \tag{5.15}$$

to make the parameters more comparable across various objectives.

5.3 Travel Costs in Adaptive Sampling of Expensive-to-evaluate Environmental Phenomena Using Mobile Robots

As described in the beginning of the chapter, the use of mobile robots to monitor environmental phenomena incurs both location-dependent costs and travel costs. In this section, we first address the travel costs of the mobile robot. As in the previous chapters, we consider the case where sampling (evaluation of the black-box function) is expensive, but sampling costs are independent of the sample location and hence constant across the design space.

The literature contains a number of works that use adaptive sampling methods for environmental sensing. Bin Zhang et al. [29] presented an adaptive sampling method that minimizes the integrated mean squared error of a local linear model while optimizing the path w.r.t. the energy available to the robot. To facilitate path optimization, the design space was discretized and then path planning was performed using the Breadth First Search Algorithm. Stankiewicz et al. [30] developed the Gaussian Process Adaptive Sampling (GPAS) method, which, depending on the available computational capacity, uses branch-and-bound techniques or cross-entropy optimization on a receding horizon to generate a trajectory with particularly informative sample points in order to obtain the maximum information with limited energy and runtime. In each iteration of the algorithm, the control inputs are optimized to maximize the accumulated values of the UCB acquisition function within the horizon, considering vehicle dynamics and obstacles as constraints. Then the first control input is applied, and the optimization is restarted. The Gaussian process (metamodel) is updated after each sampling of the phenomenon. Thus, highly localized regions of interest (ROI) could be resolved with lower energy consumption and higher accuracy than with a meandering coverage of the design space.

The aforementioned use real-time capable sensors that record data along the robot's path while the robot is moving. This requirement is not met if the mobile robot has to interrupt its motion to carry out expensive sampling, as in scenario considered in this thesis. In contrast to algorithms like GPAS, we require the samples to be taken at points in the design space where the criterion takes large values to gain most information per (expensive) sample. However, following the sequential and adaptive sampling algorithms used in the previous chapters would lead to the successive sample points mostly being far apart and thus the cumulative travel costs become large.

Choi et al. [17] proposed cost-aware adaptive sampling criteria for consideration of costs generated by parameter changes in wind tunnel experiments. In their work, cost was incorporated by dividing variance- and distance-based criteria by the cost function. They found, that division by the cost function made the combined criterion excessively sensitive to changes of the cost function which potentially leads to the cost function dominating the criterion [17]. Similar approaches can also be found in the field of Bayesian optimization for the consideration of parameter-dependent costs [25] and will be discussed in more detail in Sec. 5.4.

However, these methods are outperformed by the cost-aware sampling criterion *contextual expected improvement (CEI)* [97] (also for Bayesian optimization)

$$\varphi^{\mathrm{CEI}}(\underline{x}) = \begin{cases} -c^{x}(\underline{x}) & \text{if } \varphi^{\mathrm{EI}}(\underline{x}) \geq (1-\lambda)\max_{\underline{x}\in\mathcal{X}} \varphi^{\mathrm{EI}}(\underline{x}) \\ -\infty & \text{otherwise} \end{cases} \tag{5.16}$$

based on the EI criterion, where $c^{x}(\underline{x})$ is a cost function describing the parameter-dependent sampling costs. When applying the CEI criterion, the cheapest sample point is selected within the regions of the design space where the EI criterion takes values greater than $(1-\lambda)$ times its maximum value. This approach minimizes cost while ensuring the information gain specified by the parameter $\lambda \in [0,1]$, thereby guaranteeing a Pareto-optimal solution.

To summarize, there are no adaptive sampling methods from the field of environmental sensing that take expensive sampling into account. The most promising approach is CEI, which is used to account for parameter-dependent costs in global optimization of mixed-domain parameter spaces using Bayesian optimization (mainly hyperparameter optimization of machine learning methods). It has not yet been investigated how well comparable Pareto-optimal sampling strategies can be transferred to the consideration of travel costs of mobile robots in adaptive sampling methods for environmental sensing.

In this chapter, we close this gap by presenting two approaches that use different techniques to account for travel costs. The first approach sets up an MOP in Sec. 5.3.2 as motivated by the literature and uses a scalarization function to define a cost-aware sampling criterion. Depending on the choice of parameters this turns into the equivalent of the CEI criterion. Subsequently, in Sec. 5.3.3, we present our novel model predictive approach that predicts future sample points on a receding horizon and places them in a cost-optimal order to drastically reduce travel costs while maximizing information per sample. This approach is tailored to the application of mobile robots in environmental sensing of expensive-to-evaluate phenomena and can also be combined with other approaches, such as Pareto-optimal criteria. The two approaches represent solutions to the problem specified in Sec. 5.3.1 and are extensively evaluated in Sec. 5.3.4.

5.3.1 Problem Formulation

We consider a Lipschitz-continuous, non-negative, and bounded black-box function $f : \mathcal{X} \to \mathbb{R}$ which takes values in the interval $[0, f_{\max}]$, where $f_{\max} = \max_{\underline{x} \in \mathcal{X}} f(\underline{x})$ is a positive and finite real number. The black-box function is to be approximated within the bounded domain $\mathcal{X} \subset \mathbb{R}^n$ (the design space), where n is the number of dimensions. The approximation is based on noisy samples $y_l \in \mathbb{R}$, taken at sample points $\underline{x}_l \in \mathcal{X}$ for l=1,…,m, where m is the number of existing samples. The samples

$$y_l = f(\underline{x}_l) + \epsilon_l \tag{5.17}$$

are given by the value of the black-box function $f(\underline{x})$ at sample points $\underline{x}_l$ disturbed by additive, zero-mean noise modeled by a Gaussian noise term $\epsilon_l \sim \mathcal{N}(0, \sigma_l^2)$. For reconstruction of the phenomenon f, a metamodel $\hat{f} : \mathcal{X} \to \mathbb{R}$ is trained on the samples of the phenomenon.

In order to obtain the best possible reconstruction of the phenomenon for each number M of samples, the sample points are to be selected optimally according to a sampling criterion $\varphi(\underline{x})$ in an adaptive manner. Moreover, the cost to move the robot from one sample point to the next is described by the cost function $c^{\Delta x} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$. Depending on the application, the cost function may include the distance to be traveled, the time or energy required to travel the path, the accessibility of the terrain, etc. The goal is to sequentially suggest a new sample point to the robot, such that the accumulated path costs

$$c_{\text{cum}} = \sum_{l=1}^{M-1} c^{\Delta x}(\underline{x}_l, \underline{x}_{l+1}) \tag{5.18}$$

become minimal (with each $\underline{x}_l$ being optimal w.r.t. $\varphi(\underline{x})$).

5.3.2 Pareto-optimal Consideration of Travel Cost

In this section we formulate a cost-aware sampling criterion that takes into account the robots travel costs, utilizing Pareto-optimal multi-objective optimization (detailed in Sec. 5.1). For this, we first state the MOP

$$\min_{\underline{x}} \underbrace{\begin{bmatrix} -\varphi(\underline{x}) \\ c^{\Delta x}(\underline{x}_m, \underline{x}) \end{bmatrix}}_{\underline{J}^{\mathrm{TC}}(\underline{x})} \quad \text{s.t. } \underline{x} \in \mathcal{X} \ , \tag{5.19}$$

where $\varphi(\underline{x})$ is a sequential or adaptive sampling criterion that is to be maximized, $c^{\Delta x}(\underline{x}_m, \underline{x})$ is the travel cost incurred by the robot when travelling from its current position $\underline{x}_m$ to $\underline{x}$ that is to be minimized, and $\underline{J}^{\mathrm{TC}}(\underline{x})$ is the vector of the two competing objectives (TC indicates travel cost consideration). Since we formulate the MOP as a minimization problem, the sampling criterion $\varphi(\underline{x})$ is negated to turn its maximization into a minimization.

Next, we turn the MOP in (5.19) into the SOP

$$\min_{\underline{x}} \ s^{\mathrm{cheby}}(\underline{J}^{\mathrm{TC}}(\underline{x})) \quad \text{s.t. } \underline{x} \in \mathcal{X} \tag{5.20}$$

by applying the Chebyshev scalarization function from (5.5) to the competing objectives. Using the parameters of the scalarization function, we can now set the trade-off between the criterion and the travel costs. This trade-off favors solutions that cause lower travel costs (which often implies that they are closer to the current robot position). However, the parameters do not allow any conclusions to be drawn about the specific value of the sampling criterion in the optimum of the optimization. This contradicts the problem statement in Sec. 5.3.1, where it is required that the sample points must be optimal w.r.t. the sampling criterion used. In fact, any compromise is a violation of this requirement.

To enable a suboptimal solution with quantifiable guarantees, we add a condition to the optimization problem in (5.20)

$$\min_{\underline{x}} \ s^{\mathrm{cheby}}(\underline{J}^{\mathrm{TC}}(\underline{x})) \quad \text{s.t. } \underline{x} \in \mathcal{X} : \varphi(\underline{x}) \geq (1-\lambda) \max_{\underline{x} \in \mathcal{X}} \varphi(\underline{x}) \tag{5.21}$$

ensuring the criterion to be greater than $(1-\lambda)$ times its maximum for any possible solution, where $\lambda \in [0, 1]$. Note that when applying (5.21) to sequential or adaptive sampling, the minimizer of (5.21) will be the next sample point. Hence, we can formulate the corresponding sampling criterion

$$\varphi^{\mathrm{PoTC}}(\underline{x}) = \begin{cases} -s^{\mathrm{cheby}}(\underline{J}^{\mathrm{TC}}(\underline{x})) & \text{if } \varphi(\underline{x}) \geq (1-\lambda) \max_{\underline{x} \in \mathcal{X}} \varphi(\underline{x}) \\ -\infty & \text{otherwise} \end{cases} \tag{5.22}$$

that is to be maximized on the design space to obtain the next sample point. When specific sequential or adaptive sampling criteria are plugged into $\varphi^{\mathrm{PoTC}}(\underline{x})$ we use the following notation $\varphi^{\text{PoTC-}(\cdot)}(\underline{x})$ (e.g., $\varphi^{\text{PoTC-WMV}}(\underline{x})$ for the WMV criterion or $\varphi^{\text{PoTC-MD}}(\underline{x})$ for the MD criterion).

Remark 1. If the parameter in the scalarization function that weights $\varphi(\underline{x})$ is set to zero, the formulation in (5.22) aligns with that of CEI in (5.16) (except with a different criterion and the travel costs instead of the parameter-dependent costs). This simplifies the evaluation of the criterion, since the whole trade-off is reduced to a constrained optimization of the cost function. We recommend using this formulation for most scenarios. However, if the cost function contains (almost) flat regions we encourage to set a small scalarization weight on the sampling criterion to shift the solution towards more informative points without introducing (significant) additional cost.

5.3.3 Receding Horizon Cost-aware Adaptive Sampling

In this section, we present our novel *receding horizon cost-aware adaptive sampling (RHCaAS)* [34] algorithm for reducing the cumulative travel costs of the mobile robot. In contrast to the Pareto-optimal criterion in Sec. 5.3.2, no trade-off is made between the sequential or adaptive sampling criterion used, but the cumulative travel costs are approximated and optimized over a prediction horizon.

The idea behind the algorithm is that if we knew the future sample points, we were able to find the most cost-effective path that connects all points. However, the number M of all samples is often unknown. In addition, the exact prediction of future sample points is hindered by their dependence on the (unknown) black-box function. Consequently, the algorithm predicts future sample points in each iteration over a defined horizon based on the metamodel. These predicted points are then arranged in a cost-optimal sequence, with the first point in the sequence selected as the next sample point. This approach significantly reduces cumulative travel costs. RHCaAS is composed of an outer loop (real sampling) and an inner loop (virtual sampling) which are detailed in the following sections.

Real Sampling (Outer Loop)

In real sampling, sequential sampling is performed on the (real) phenomenon as shown in Fig. 5.3. It is warm-started with a non-empty initial data set $\mathcal{D}_0 = \{(\underline{x}_l, y_l)\}_{l=1}^{M_0}$, where M_0 is the number of initial samples which are taken in a (precomputed) cost-optimal order. A space-filling initial design is used as a basis for an initial training of the metamodel since this results in the least model error [21] when reconstructing unknown functions [21, 27]. Then, in each iteration i, virtual sampling is used to determine the next sample point

$$\underline{x}_{m+1} = \text{VirtualSampling}(\mathcal{X}, \mathcal{D}_i, \hat{f}, H) \ , \tag{5.23}$$

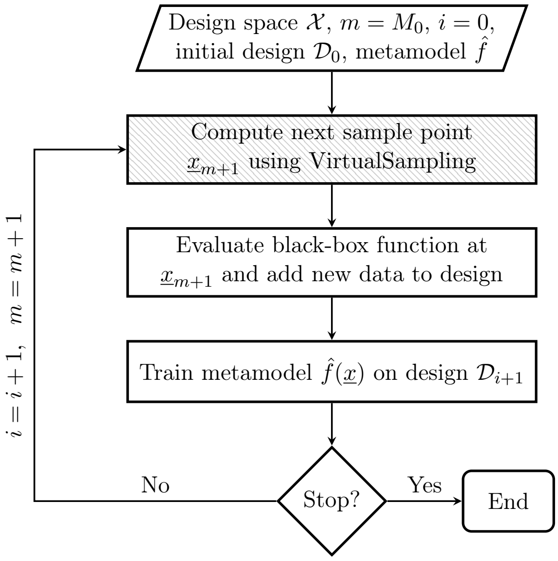
Design space $\mathcal{X}$, $m = M_0$, $i = 0$, initial design $\mathcal{D}_0$, metamodel $\hat{f}$
Compute next sample point $\underline{x}_{m+1}$ using VirtualSampling
Evaluate black-box function at $\underline{x}_{m+1}$ and add new data to design
Train metamodel $\hat{f}(\underline{x})$ on design $\mathcal{D}_{i+1}$
Stop?
Yes
End
No
$i = i + 1,\ m = m + 1$

Figure 5.3: Real sampling analogous to the adaptive sampling procedure in Sec. 2.2.3. Virtual sampling is performed to determine the next sample point in each iteration.

the phenomenon is sampled at the next sample point $\underline{x}_{m+1}$ to obtain the next sample y_{m+1}, the new data $\{\underline{x}_{m+1}, y_{m+1}\}$ is added to the design

$$\mathcal{D}_{i+1} = \mathcal{D}_i \cup \{\underline{x}_{m+1}, y_{m+1}\} \ , \tag{5.24}$$

and the metamodel $\hat{f}$ is trained based on the current design $\mathcal{D}_{i+1}$. The shading in Fig. 5.3 indicates the step of real sampling where virtual sampling is performed.

Virtual Sampling (Inner Loop)

As described in (5.23), virtual sampling determines the next sample point based on the design space $\mathcal{X}$, the current design $\mathcal{D}_i$, the metamodel $\hat{f}$, and the horizon length H. For this, adaptive sampling is again performed as described in Fig. 5.3. However, virtual samples are taken from the metamodel instead of the (real) phenomenon. The algorithm is warm-started with the current design $\mathcal{D}_i$ as the initial design and the metamodel $\hat{f}$. In addition, the virtual design $\tilde{\mathcal{D}}_h = \emptyset$ for $h = 0$ is initialized as an empty set. In each iteration h, the next virtual sample point

$$\underline{\tilde{x}}_{h+1} = \arg \max_{\underline{\tilde{x}} \in \mathcal{X}} \ \varphi(\underline{\tilde{x}}) \tag{5.25}$$

is determined by maximizing the used sampling criterion φ, the metamodel is evaluated at the next virtual sample point $\underline{\tilde{x}}_{h+1}$ to obtain the next virtual sample $\tilde{y}_{h+1}$, the virtual data $\{\underline{\tilde{x}}_{h+1}, \tilde{y}_{h+1}\}$ is added to the virtual design

$$\tilde{\mathcal{D}}_{h+1} = \tilde{\mathcal{D}}_h \cup \{\underline{\tilde{x}}_{h+1}, \tilde{y}_{h+1}\} \ , \tag{5.26}$$

and the metamodel $\hat{f}$ is then trained based on the existing real and virtual samples $\mathcal{D}_i \cup \tilde{\mathcal{D}}_{h+1}$.

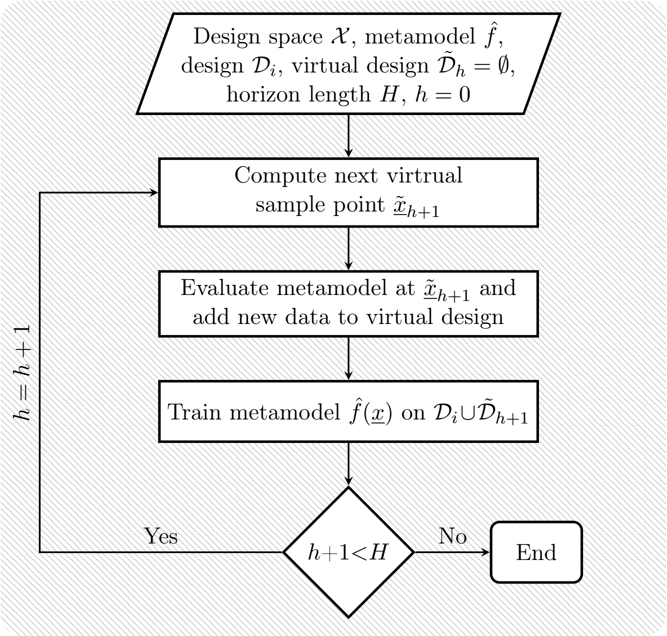
Design space $\mathcal{X}$, metamodel $\hat{f}$, design $\mathcal{D}_i$, virtual design $\tilde{\mathcal{D}}_h = \emptyset$, horizon length H, $h = 0$
Compute next virtrual sample point $\underline{\tilde{x}}_{h+1}$
Evaluate metamodel at $\underline{\tilde{x}}_{h+1}$ and add new data to virtual design
Train metamodel $\hat{f}(\underline{x})$ on $\mathcal{D}_i \cup \tilde{\mathcal{D}}_{h+1}$
$h+1<H$
Yes
No
End
$h = h + 1$

Figure 5.4: Virtual sampling procedure of RHCaAS with a horizon length of H virtual samples, starting with the current design $\mathcal{D}_i$ as initial design.

The loop is repeated until H virtual sample points are determined, where H denotes the length of the prediction horizon. Next, the optimal permutation

$$\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} \left(c^{\Delta x} \left(\underline{x}_m, \underline{x}_{\boldsymbol{\pi}_1} \right) + \sum_{l=1}^{H-1} c^{\Delta x} \left(\underline{x}_{\boldsymbol{\pi}_l}, \underline{x}_{\boldsymbol{\pi}_{l+1}} \right) \right) \tag{5.27}$$

of all permutations $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_H)$ is determined, which, starting from the current sample point $\underline{x}_m$, minimizes the costs to visit all virtual sample points $\{\underline{\tilde{x}}_1, \ldots, \underline{\tilde{x}}_H\}$. Virtual sampling returns the virtual sample point $\underline{\tilde{x}}_{\boldsymbol{\pi}_1^*}$ indexed by the first element $\boldsymbol{\pi}_1^*$ of the optimal permutation $\boldsymbol{\pi}^*$ as the next sample point

$$\underline{x}_{m+1} = \underline{\tilde{x}}_{\boldsymbol{\pi}_1^*} \tag{5.28}$$

for evaluation of the (real) phenomenon to real sampling. The virtual sampling algorithm is illustrated in Fig. 5.4.

Remark 2. Consider the case where the mean value of the metamodel equals the real phenomenon on the entire design space (e.g., the virtual samples all have the value that would also be measured in real at that location, neglecting measurement noise) and the horizon length H is chosen such that $M = M_0 + H$ corresponds to the desired total number of samples. Then the virtual sample points would be equal to those that would have been determined even without virtual sampling. The optimal permutation would then correspond to the cost-optimal solution according to (5.18) for an initial design with M_0 samples and H adaptive samples. However, since the metamodel is error-prone, RHCaAS provides only a suboptimal solution. Moreover,

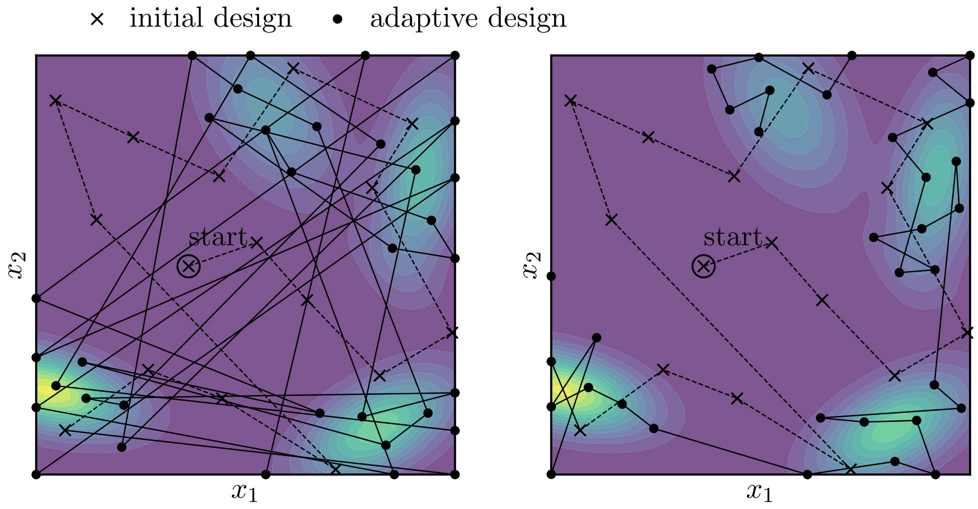

Figure 5.5: Initial design (crosses) and adaptive design (dots) of RHCaAS-WMV for a horizon length of $H = 1$ (no cost optimization) on the left and $H = 8$ on the right on top of a randomly generated black-box function. The initial design was traversed in a cost-optimal order, starting from a randomly selected initial sample point (indicated by the circle).

predictions over large horizons H are inaccurate, which is why the horizon practically includes only a few next points.

Remark 3. RHCaAS is independent of the criterion used which extends beyond the sequential and adaptive sampling criteria presented in Chap. 2 – Chap. 4. In particular, this includes criteria such as $\varphi^{\text{PoTC}}(\underline{x})$ defined in Sec. 5.3.2. As a result, the consideration of travel costs in the scalarization function in (5.22) can be combined with the minimization of cumulative travel costs over a horizon of RHCaAS (which will be included in the evaluation in Sec. 5.3.4 and utilized in Sec. 5.5.1). When applying RHCaAS with a specific sampling criterion, we use the notation $\varphi^{\text{RHCaAs-}(\cdot)}(\underline{x})$ (e.g., $\varphi^{\text{RHCaAs-WMV}}(\underline{x})$ for the WMV criterion).

5.3.4 Evaluation

In this section, the following three methods for consideration of the mobile robot's travel costs are compared with each other.

PoTC-WMV: PoTC presented in Sec. 5.3.2 was used with the WMV criterion ($\alpha = 0.025$) in Sec. 4.2.2 and evaluated for different parameters λ. The weight for the sampling criterion in the scalarization function was set to zero.

RHCaAS-WMV: RHCaAS from Sec. 5.3.3 was used along with the WMV criterion (α=0.025) in Sec. 4.2.2 and evaluated for different horizon lengths H.

RHCaAS-PoTC-WMV: PoTC-WMV ($\alpha = 0.025$, $\lambda = 0.1$) was used as a sampling criterion for RHCaAS and evaluated for different horizon lengths H.

OLHS of 16 sample points (computed as described in Sec. 3.2.1) was used in all simulations as space-filling initial design. The experimental setup for the numerical evaluation is described in Sec. 5.2.

In Fig. 5.6 and Fig. 5.7 the NRMSE and the Rel. MCTCS are shown versus the number of samples. It can be seen that the parameter λ for PoTC-WMV has a major impact on the NRMSE and the travel costs. For $\lambda = 0.1$, over 40 % of the total travel costs can already be saved over 100 samples, while only a minimal degradation of the model accuracy can be observed. With increasing λ, significant cost savings can be achieved, leading to a drastic degradation of the NRMSE for all numbers of samples. This can be explained by the fact that the robot is always given the opportunity to select sample points close to its current position for large λ and thus, due to the dominance of path costs in the optimization, mostly non-informative sample points (w.r.t. the WMV criterion) are selected.

With RHCaAS-WMV, there is also a direct correlation between the horizon length H and the savings in travel costs. However, a considerable degradation in model accuracy can only be observed with a small number of samples, which decreases with an increasing number of samples until the horizon length H no longer has a significant impact on the NRMSE from about 70 samples. This has two reasons:

- The model error of the metamodel decreases with an increasing number of samples, which leads to suboptimal predictions of the next sample points in the virtual sampling loop, especially with a small number of samples.
- The collection of sample points in a more travel-cost-optimal order results in the robot initially enhancing the local model accuracy before moving to other regions of the design space. This implies that sample points, which would be selected in future iterations without virtual sampling, are prioritized due to cost optimization (and vice versa). As illustrated in Fig. 5.5, RHCaAS-WMV selects globally informative sample points in a sequence that facilitates piecewise local exploration of the black-box function.

RHCaAS-PoTC-WMV combines PoTC-WMV and RHCaAS-WMV by using PoTC-WMV as a sampling criterion for RHCaAS. The combination is interesting because PoTC-WMV for $\lambda = 0.1$ only achieves a small increase in NRMSE, but already a comparable cost saving to RHCaAS-WMV with a horizon length of $H = 6$. As can be seen in Fig. 5.6, the NRMSE of RHCaAS-PoTC-WMV is almost identical to that of RHCaAS-WMV. However, RHCaAS-PoTC-WMV already achieves greater cost savings than RHCaAS-WMV at a horizon length of $H = 1$, as shown in Fig. 5.7. As the horizon length H increases, however, this effect decreases, resulting in RHCaAS-PoTC-WMV achieving a comparable cost saving to RHCaAS-WMV from $H = 6$.

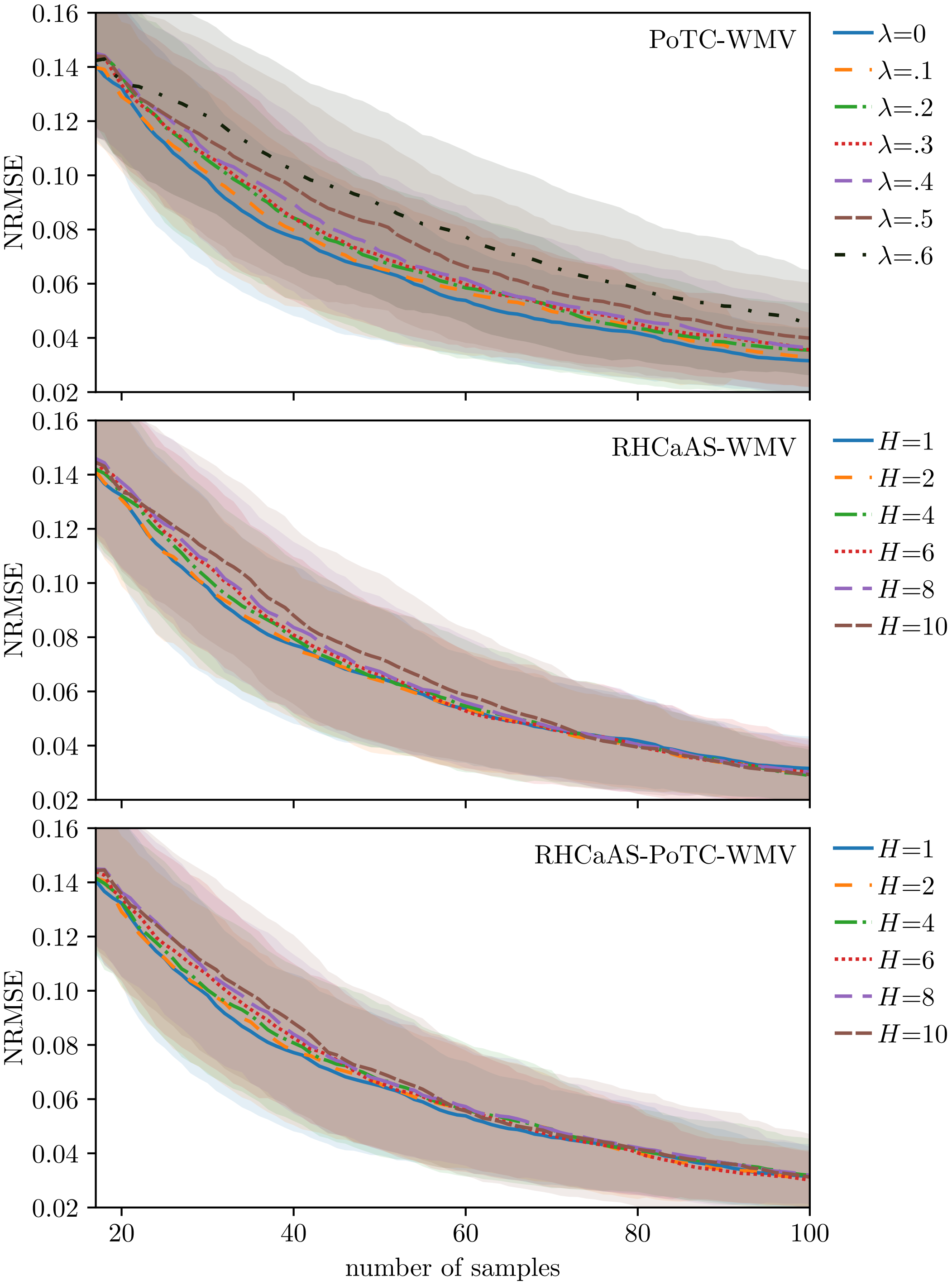

Figure 5.6: The mean (lines) and standard deviation (shaded area) of the NRMSE over $N^{\text{runs}} = 100$ randomly generated black-box functions vs the number of sample points per method.

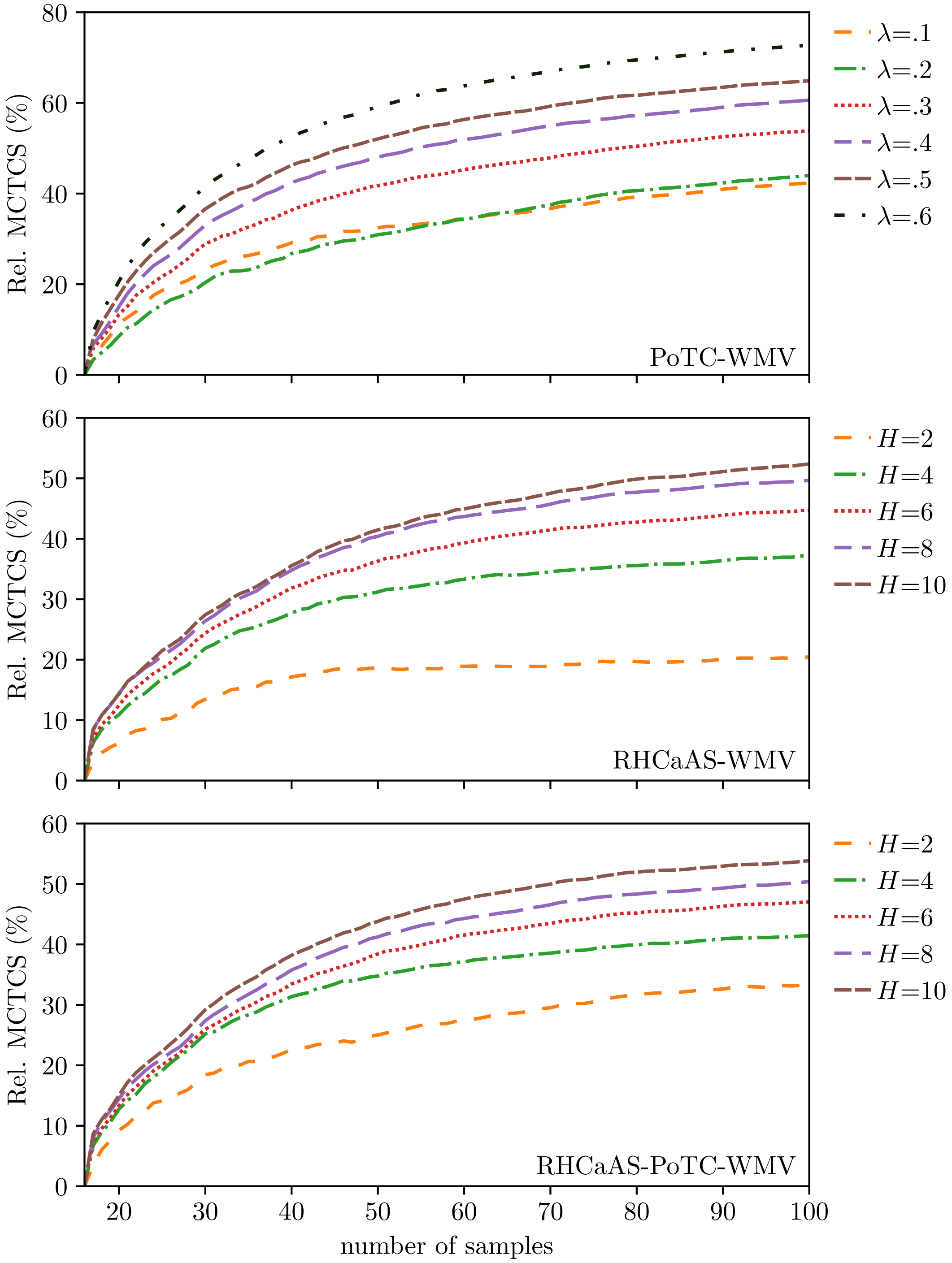

Figure 5.7: The relative mean cumulative travel cost savings (Rel. MCTCS) over $N^{\text{runs}} = 100$ randomly generated black-box functions vs the number of sample points per method.

5.4 Local Sampling Costs in Adaptive Sampling of Expensive-to-evaluate Environmental Phenomena Using Mobile Robots

While the previous Sec. 5.3 focused on the robots travel costs, this section is concerned with the consideration of location-dependent costs in adaptive sampling for environmental sensing using mobile robots. By considering location-dependent costs in the choice of sample points, better model accuracy can be achieved for a given budget. However, to the best of our knowledge, there is no literature on location-dependent sampling costs in the context of adaptive sampling for environmental sensing using mobile robots and little literature from global metamodeling and Bayesian (global) optimization to account for parameter-dependent sampling costs.

As mentioned in Sec. 5.3, there is one work by Choi et al. [17] in global metamodeling considering the cost of altering the parameters in wind tunnel testing by dividing distance- and variance-based criteria by the cost function. Similar approaches are pursued in the context of Bayesian Optimization (BO) described in Sec. 2.2.3, where the acquisition function EI (see Sec. 2.2.3) is divided by the cost function

$$\varphi^{\mathrm{EIpu}}(\underline{x}) = \frac{\varphi^{\mathrm{EI}}(\underline{x})}{c^{x}(\underline{x})} \tag{5.29}$$

to obtain the expected improvement per unit (EIpu) [40]. EIpu was modified by Lee et al. in their Cost Appointed BO (CArBO) [25] method

$$\varphi^{\mathrm{CArBO}}(\underline{x}) = \frac{\varphi^{\mathrm{EI}}(\underline{x})}{c^{x}(\underline{x})^{\tau}} \;, \tau \geq 0 \tag{5.30}$$

in such a way that the influence of the cost function decreases with the number of samples (cost cooling). Division by the cost function often leads to it having too great an influence on the choice of the next sample point [17]. This effect is particularly significant if the cost function has a big range or the costs can become very small on the design space.

In addition, in their CArBO paper [25] the authors proposed a cost-efficient initial design, which we refer to as CeID in the following. To generate the CeID, the design space is discretized, the cost function is evaluated at all discrete points and the results are stored in a data set. In each iteration of the algorithm, the most expensive point and the point closest to the existing sample points are then alternately removed from the data set until only one point remains. This is then used as the next sample point. The process is repeated until the cumulative sampling costs are below a certain budget (or a sufficient number of sample points have been generated). In the first iteration, the cheapest of the discrete points is used as the next sample point, as there are no existing sample points.

In one of our previous works, we presented the cost-aware Voronoi sampling algorithm CAV [32], which performs a local optimization of the cost function within the neighborhood of the most informative sample point. Unlike the methods that divide by the cost function, the algorithm poses no requirements on the course of the cost function [32]. It has also been shown that this approach clearly outperforms CeID when used as an initial design.

The solutions of CArBO always lie on the Pareto front, whose position on the Pareto front can be varied with the parameter τ [97]. In the CArBO method, the parameter is not set to a fixed value, but is constantly reduced depending on the remaining budget (cost-cooling), which leads to exploration of more costly regions of the design space with increasing number of samples.

Guinet et al. introduce the cost-aware sampling criterion *contextual EI (CEI)* [97] based on the EI acquisition function as defined in (5.16) in Sec. 5.3. By minimizing $\varphi^{\text{CEI}}(\underline{x})$, the minimizer of the cost function in the regions of the design space in which the criterion is greater than $(1-\lambda)$ times its maximum value on the design space is obtained as next sample point, where $\lambda \in [0,1]$. CEI outperforms CArBO in various global optimization tasks (hyperparameter optimization of machine learning methods) since it dynamically adjusts the compromise between the objectives on the Pareto front in each iteration of the adaptive sampling algorithm and is less sensitive to the type and range of cost function [97].

Note that if the cost function is not known a cost model needs to be trained in addition to the metamodel of the black-box function [25, 40] in all methods that take into account parameter-dependent sampling costs.

None of the methods mentioned are tailored for use in adaptive sampling for environmental sensing. As already mentioned in Sec. 5.3.2, CEI outperforms other methods in various global optimization tasks. Pareto-optimal methods following the same principle are therefore promising candidates for cost consideration when using mobile robots for environmental sensing. After defining the problem under consideration in Sec. 5.4.1, in this section we will use the Pareto-optimal sampling criterion introduced in Sec. 5.3.2 to consider location-dependent sampling costs. For this purpose, we define two specific variants: one for the generation of a cost-aware initial design based on the mod. maximin criterion from Sec. 3.1.3 and one based on the WMV criterion from Sec. 4.2.2 for adaptive sampling of environmental phenomena. Finally, the methods are evaluated in Sec. 5.4.3.

5.4.1 Problem Formulation

We consider a Lipschitz-continuous, non-negative, and bounded black-box function $f : \mathcal{X} \to \mathbb{R}$ which takes values in the interval $[0, f_{\max}]$, where $f_{\max} = \max_{\underline{x} \in \mathcal{X}} f(\underline{x})$

is a positive and finite real number. The black-box function is to be approximated within the bounded domain $\mathcal{X} \subset \mathbb{R}^n$ (the design space), where n is the number of dimensions. The approximation is based on noisy samples $y_l \in \mathbb{R}$, taken at sample points $\underline{x}_l \in \mathcal{X}$ for $l = 1, \ldots, m$, where m is the number of existing samples.

The samples $y_l = f(\underline{x}_l) + \epsilon_l$ are given by the value of the black-box function $f(\underline{x})$ at sample points $\underline{x}_l$ disturbed by additive, zero-mean noise modeled by a Gaussian noise term $\epsilon_l \sim \mathcal{N}(0, \sigma_l^2)$. For reconstruction of the phenomenon f, a metamodel $\hat{f} : \mathcal{X} \to \mathbb{R}$ is trained on the samples of the phenomenon.

Furthermore, the sampling costs depend on the location $\underline{x}$ in the design space $\mathcal{X}$ and are given by the non-negative cost function $c_l^x : \mathcal{X} \to \mathbb{R}_0^+$. For a series of m sequential samples, the accumulated costs

$$c_{\text{cum}}^x = \sum_{l=1}^{m} c_l^x(\underline{x}_l) \tag{5.31}$$

can be calculated as the sum of costs per iteration.

The goal is to sequentially determine the sample points $\underline{x}_l$ such that, for a given budget $c_{\max}^x$, the global error between the metamodel $\hat{f}$ and the black-box function f becomes minimal w.r.t. a given metric. Accordingly, the budget represents the upper bound for the accumulated costs $c_{\text{cum}}^x \leq c_{\max}^x$.

5.4.2 Pareto-optimal Cost-aware Adaptive Sampling

In this section, we use the results from Sec. 5.3.2 to define a sampling criterion for the Pareto-optimal consideration of location-dependent sampling costs. Analogous to (5.19), we set up the MOP

$$\min_{\underline{x}} \underbrace{\begin{bmatrix} -\varphi(\underline{x}) \\ c^x(\underline{x}) \end{bmatrix}}_{\underline{J}^{\text{LC}}(\underline{x})} \quad \text{s.t. } \underline{x} \in \mathcal{X} \ , \tag{5.32}$$

where $\varphi(\underline{x})$ is a sequential or adaptive sampling criterion that is to be maximized and $c^x(\underline{x})$ denotes the sampling costs at the point $\underline{x}$. The abbreviation LC indicates the consideration of location-dependent (local) costs. Using the Chebyshev scalarization function s^{cheby} from (5.5), we transform the MOP in (5.32) into an SOP, where the parameters of the scalarization function determine the trade-off between the two objectives in $\underline{J}^{\text{LC}}(\underline{x})$.

Analogous to the criterion $\varphi^{\text{PoTC}}(\underline{x})$ in (5.22), we define the Pareto-optimal cost-aware criterion

$$\varphi^{\text{PoLC}}(\underline{x}) = \begin{cases} -s^{\text{cheby}}(\underline{J}^{\text{LC}}(\underline{x})) & \text{if } \varphi(\underline{x}) \geq (1-\lambda) \max_{\underline{x} \in \mathcal{X}} \varphi(\underline{x}) \\ -\infty & \text{otherwise,} \end{cases} \tag{5.33}$$

where $\lambda \in [0,1]$. When inserting sequential or adaptive sampling criteria into $\varphi^{\text{PoLC}}(\underline{x})$, we use the notation $\varphi^{\text{PoTC-}(\cdot)}(\underline{x})$. Specifically, for use of the WMC criterion and the sequential mod. maximin criterion, we define the notation $\varphi^{\text{PoLC-WMV}}(\underline{x})$, $\varphi^{\text{PoLC-ModMM}}(\underline{x})$.

5.4.3 Evaluation

In this section, the effectiveness of PoLC is examined in simulation using the experimental setup described in Sec. 5.2. The following three scenarios are considered:

- **Scenario 1** – A space-filling sequential design is generated, assuming the cost function to be known (Fig. 5.8)
- **Scenario 2** – An adaptive design is generated, assuming the cost function to be known (Fig. 5.9)
- **Scenario 3** – An adaptive design is generated, assuming the cost function to be unknown (Fig. 5.10).

In the adaptive sampling methods, we employed an initial design of 16 space-filling sample points. For scenarios involving an unknown cost function, we used OLHS of 16 sample points, as detailed in Sec. 3.2.1. If different sampling methods were used for the initial design and the adaptive design, we use the notation A | B, where A denotes the initial design and B the adaptive design. The sequential mod. maximin criterion is abbreviated to ModMM in this section (for reasons of space in the figure legends). PoLC was applied with $\lambda = 0.3$ and the weight of the criterion in the scalarization function was set to zero.

Scenario 1 Fig. 5.8 compares space-filling sampling methods. As known from Sec. 3.2, ModMM delivers an NRMSE similar to that of LHS, with LHS slightly outperforming ModMM. PoLC-ModMM achieves a comparable model accuracy in terms of NRMSE but performs marginally worse than ModMM. Conversely, CeID yields a significantly higher NRMSE than the other sampling methods for any number of samples.

When considering the NMCPS, it is evident that CeID generates the lowest costs per sample by a significant margin. LHS and ModMM do not account for sampling costs, thereby incurring the average costs per sample for each cost function (as they are space-filling sampling methods). As discussed in Sec. 3.2, LHS, as a one-shot design, exhibits superior space-filling properties compared to ModMM, and consequently results in an NMCPS slightly closer to 1. PoLC-ModMM starts with comparable NMCPS to CeID, but unlike CeID, shows a steady increase in NMCPS with the number of samples. This can be explained by the fact that regions where ModMM

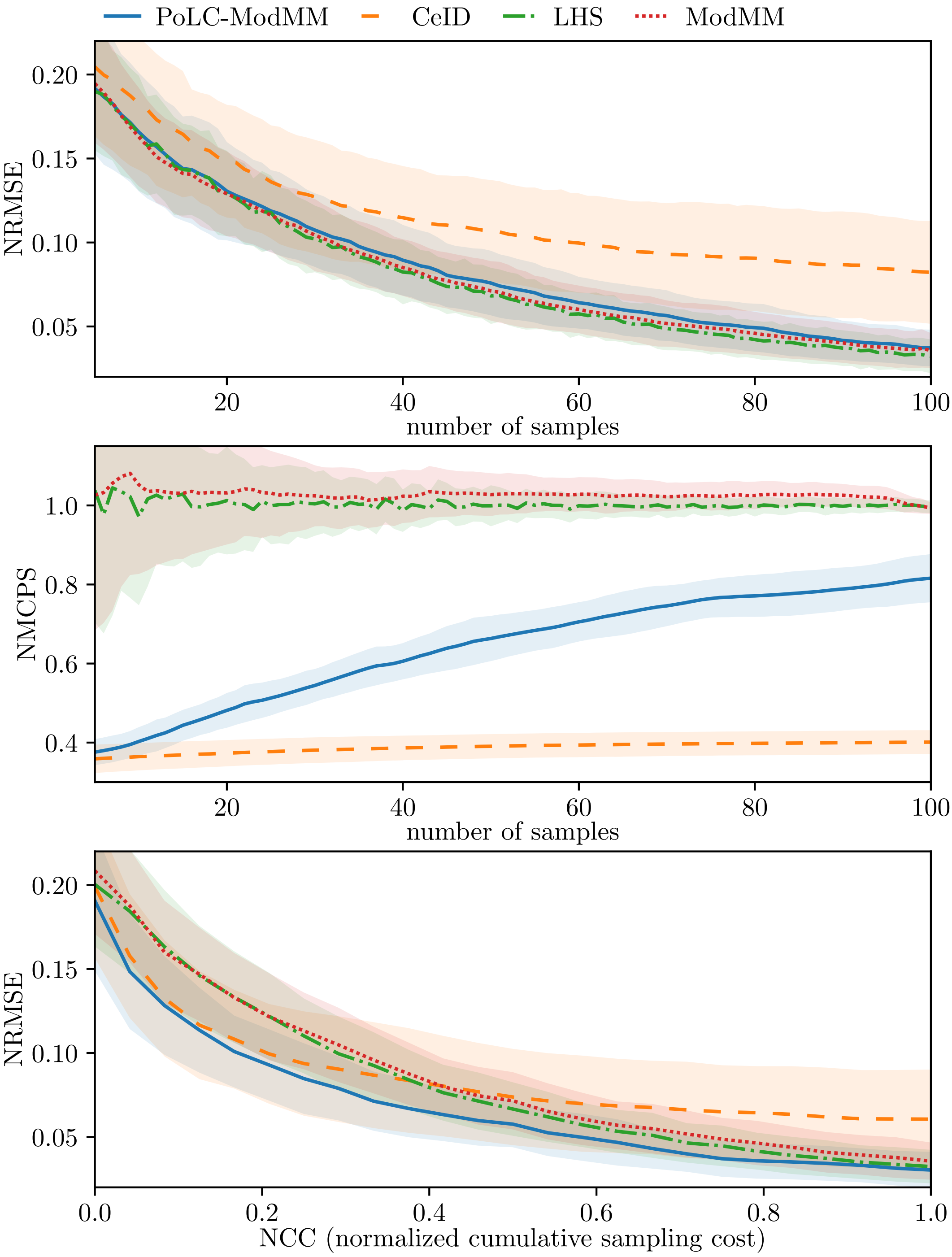

Figure 5.8: The mean (lines) and standard deviation (shaded area) of the measures NRMSE and NMCPS over $N^{\text{runs}} = 100$ randomly generated black-box and cost function pairs vs the number of sample points or NCC, respectively. The cost function is known to the sampling algorithms.

takes values close to its current maximum value (above the threshold defined by λ), and within which cost can be minimized, become smaller with an increasing number of samples (cost cooling).

PoLC-ModMM offers the lowest NRMSE per NCC (NCC can be considered normalized invested budget). ModMM and LHS yield comparable NRMSEs, with LHS causing a slightly smaller NRMSE. As the NCC increases, the difference between PoLC-ModMM and the methods without cost consideration diminishes. For small NCCs, CeID results in a slightly higher NRMSE than PoLC-ModMM. As the NCCs increase, CeID performs progressively worse compared to the other sampling methods, delivering poorer model accuracy than LHS and ModMM for NCC$>$ 0.4. While CeID ensures low costs per sample, it fails to maintain the space-filling properties of the resulting design, leading to decreased model accuracy compared to other sampling methods. PoLC-ModMM maintains space-filling properties for any number of samples or NCC, thereby reducing the ability of cost consideration with an increasing number of samples (cost cooling), and thus the model accuracy regarding the NCC approaches that of LHS and ModMM.

Scenario 2 In Fig. 5.9, the comparison of adaptive sampling methods is presented. Since the cost function is assumed to be known, cost-aware strategies can be employed to generate a space-filling initial design. To assess the impact of a cost-aware initial design on the adaptive designs WMV and PoLC-WMV, simulations were conducted using both LHS and PoLC-ModMM as the initial designs.

As illustrated in Fig. 5.9, all examined sampling methods achieve a comparable NRMSE per number of samples, with the adaptive design of PoLC-ModMM | PoLC-WMV slightly underperforming. When considering the NMCPS over the number of samples, the progression of PoLC-WMV varies depending on the initial design used. PoLC-ModMM | PoLC-WMV incurs the lowest costs per sample for any number of samples. The NMCPS of LHS | PoLC-WMV start at those of LHS and approach the NMCPS of PoLC-ModMM | PoLC-WMV as the number of samples increases. Similarly, the adaptive design of PoLC-ModMM | WMV starts with the NMCPS of PoLC-ModMM and approaches the NMCPS of LHS with an increasing number of samples.

This is also reflected in the assessment of the NRMSE as a function of the NCC. PoLC-ModMM | PoLC-WMV outperforms LHS | PoLC-WMV and PoLC-ModMM | WMV for small budgets, while the curves of PoLC-ModMM | WMV and PoLC-ModMM | PoLC-WMV overlap for small budgets, with PoLC-ModMM | WMV resulting in a larger NRMSE than PoLC-ModMM | PoLC-WMV for larger budgets. The difference in NRMSE among the studied sampling methods diminishes with increasing NCC, leading to comparable NRMSE for NCC$>$ 0.8, with the NRMSE of PoLC-ModMM | PoLC-WMV and LHS | PoLC-WMV slightly below that of PoLC-ModMM | WMV and

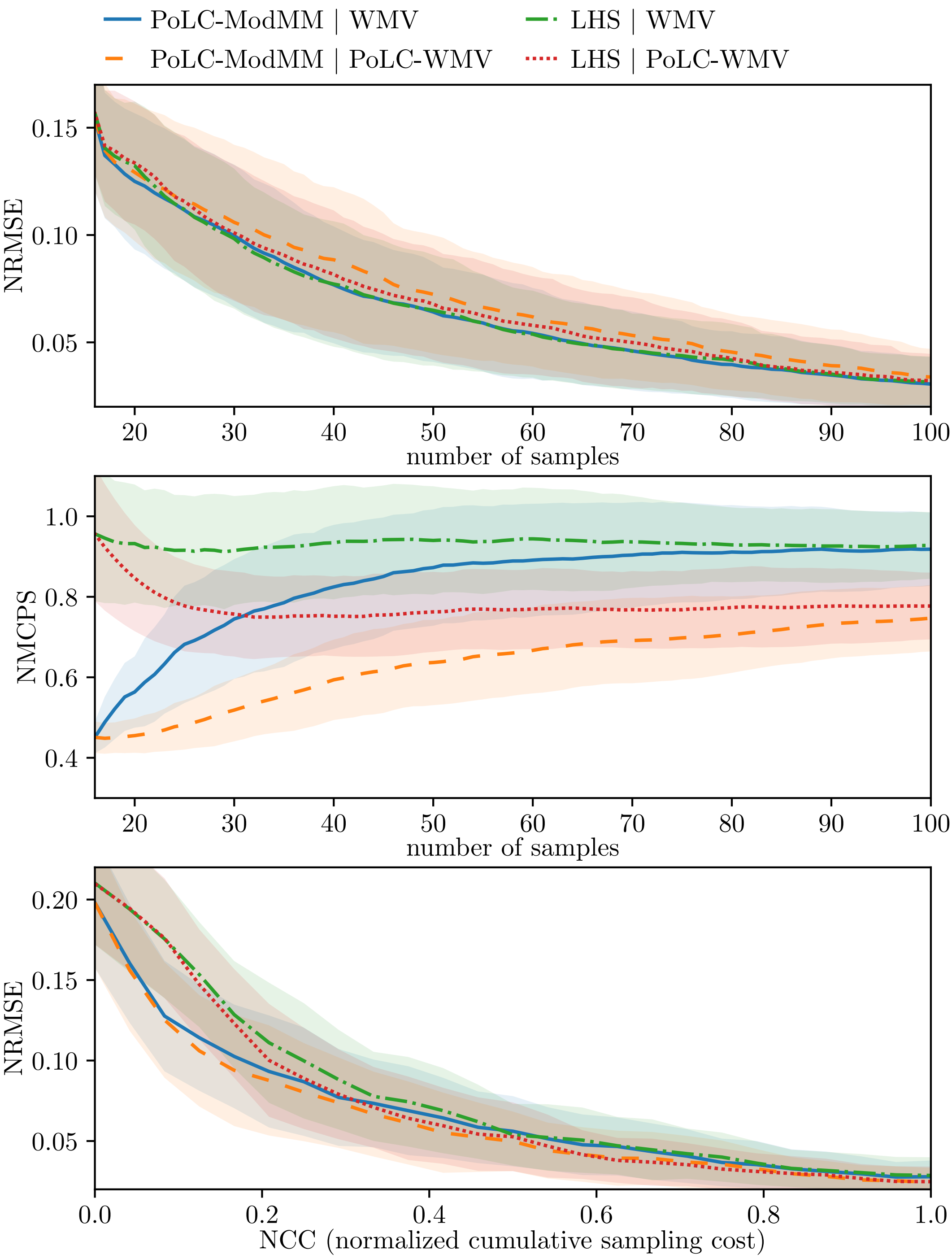

Figure 5.9: The mean (lines) and standard deviation (shaded area) of the measures NRMSE and NMCPS over $N^{\text{runs}} = 100$ randomly generated black-box and cost function pairs vs the number of sample points or NCC, respectively. The cost function is known to the sampling algorithm.

LHS | WMV. Hence, the consideration of location-dependent sampling costs has a more significant impact on the resulting model accuracy when only small budgets are available. This is due to the fact that, beyond a certain number of samples, more expensive samples must be selected to maintain the information gain per sample.

Scenario 3 In Scenario 3, the sampler lacks prior information about the cost function. Consequently, in addition to the metamodel, it is necessary to train a cost model based on the sampling costs incurred at the sample points, as described in Sec. 5.2. This cost model is then used for consideration of sampling costs in the selection of the next sample points. Since the cost function is unknown, a cost-aware initial design cannot be implemented, leading to the adaptive algorithms being warm-started with LHS, as outlined at the beginning of the section.

Fig. 5.10 illustrates that LHS | PoLC-WMV incurs lower NMCPS than LHS | WMV by considering the approximated sampling costs based on the cost model. Unlike Fig. 5.9, where LHS | PoLC-WMV demonstrates a significantly lower NRMSE for NCC > 0.2 compared to LHS | WMV, the NRMSE of the two methods in Fig. 5.10 are nearly identical. Although the NRMSE for LHS | PoLC-WMV is slightly higher w.r.t. the number of samples and marginally lower w.r.t. NCC compared to that of LHS | WMV, the differences are not significant. To effectively deploy the PoLC, it is advisable to incorporate prior knowledge about the course of the cost function into the cost model in order to be able to use it for the generation of a cost-aware initial design and to achieve good model accuracy of the cost model even with a smaller number of sample points.

5.5 Local and Travel Costs in Adaptive Sampling of Expensive-to-evaluate Environmental Phenomena Using Mobile Robots

In this section, we combine the insights and results from the previous sections Sec. 5.3 and Sec. 5.4 to address both problem statements in Sec. 5.3.1 and Sec. 5.4.1 equally. In this way, we enable the simultaneous consideration of travel costs and location-dependent sampling costs, which (as discussed at the beginning of this chapter) is relevant for the application of adaptive sampling in environmental sensing using mobile robots. To the best of our knowledge, there are no adaptive sampling methods in the literature that integrate both parameter-dependent (location-dependent) costs and the costs associated with modifying parameters (travel costs) within the context of global metamodeling, related domains such as Bayesian Optimization, or environmental sensing using mobile robots. Our novel method presented in Sec. 5.5.1 closes this gap and is extensively evaluated in Sec. 5.5.2.

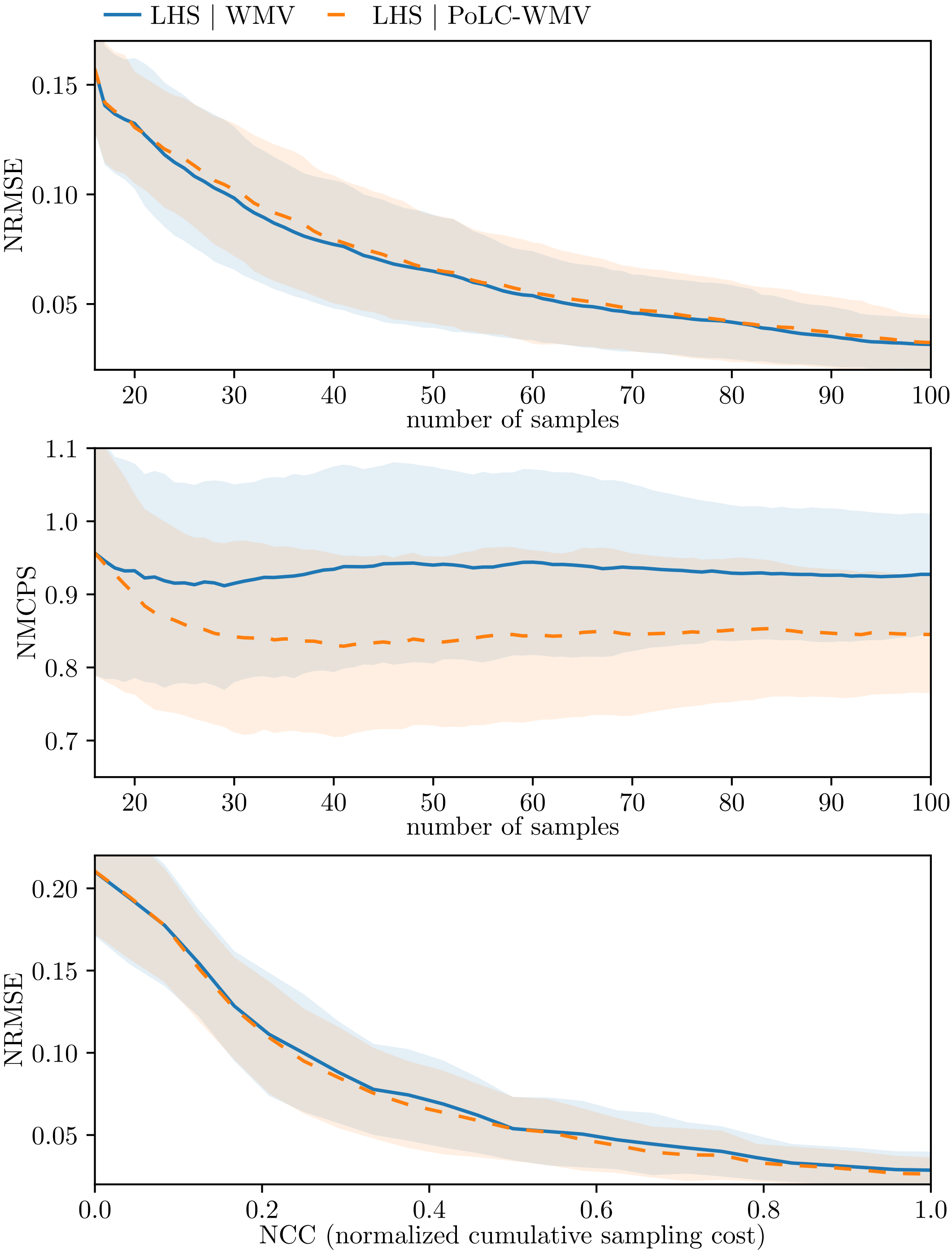

Figure 5.10: The mean (lines) and standard deviation (shaded area) of the measures NRMSE and NMCPS over $N^{\text{runs}} = 100$ randomly generated black-box and cost function pairs vs the number of sample points or NCC, respectively. The cost function is unknown and approximated by a cost model.

5.5.1 Receding Horizon Pareto-optimal Cost-aware Adaptive Sampling

In Sec. 5.3.4 it was demonstrated that RHCaAS in combination with $\varphi^{\mathrm{PoTC}}(\underline{x})$ results in greater savings of travel costs without significant impact on model accuracy than the mere consideration of $\varphi(\underline{x})$, if the travel costs are weakly weighted against the criterion. We make use of this principle in this section by defining a Pareto-optimal sampling criterion that compromises between the travel costs and location-dependent (local) costs which is then used as criterion for RHCaAS. First, we define the vector of competing objectives for the according MOP

$$\min_{\underline{x}} \underbrace{\begin{bmatrix} c^{x}(\underline{x}) \\ c^{\Delta x}(\underline{x}_m, \underline{x}) \end{bmatrix}}_{\underline{J}^{\mathrm{LTC}}(\underline{x})} \quad \text{s.t. } \underline{x} \in \mathcal{X} : \varphi(\underline{x}) \geq (1-\lambda) \max_{\underline{x} \in \mathcal{X}} \varphi(\underline{x}) \tag{5.34}$$

to exclusively encompass the cost functions while the criterion is only reflected in the constraint, where $\varphi(\underline{x})$ is a sequential or adaptive sampling criterion and $\lambda \in [0, 1]$. This allows for a better trade-off between the cost functions in regions of high values of the sampling criterion while avoiding an additional trade-off (or tuning) parameter. Hence, the resulting Pareto-optimal sampling criterion

$$\varphi^{\mathrm{PoLTC}}(\underline{x}) = \begin{cases} -s^{\mathrm{cheby}}(\underline{J}^{\mathrm{LTC}}(\underline{x})) & \text{if } \varphi(\underline{x}) \geq (1-\lambda) \max_{\underline{x} \in \mathcal{X}} \varphi(\underline{x}) \\ -\infty & \text{otherwise} \end{cases} \tag{5.35}$$

can be defined as in (5.22) and (5.33). The abbreviation LTC indicates that both travel costs and local sampling costs are considered. Following the convention from Sec. 5.3.2 and Sec. 5.4.2, we use the notation $\varphi^{\text{PoLTC-}(\cdot)}(\underline{x})$ when inserting sequential or adaptive sampling criteria into (5.35) (e.g., $\varphi^{\text{PoLTC-WMV}}(\underline{x})$ for the WMV criterion or $\varphi^{\text{PoLTC-ModMM}}(\underline{x})$ for the sequential mod. maximin criterion). We define *receding horizon Pareto-optimal cost-aware adaptive sampling (RHPoCaAS)* to be RHCaAS where any $\varphi^{\text{PoLTC-}(\cdot)}(\underline{x})$ is used as sampling criterion. Accordingly, we denote RHPoCaAS with a particular Pareto-optimal criterion $\varphi^{\text{PoLTC-}(\cdot)}(\underline{x})$ RHCaAS-PoLTC-$(\cdot)$ (e.g., RHCaAS-PoLTC-WMV).

5.5.2 Evaluation

In this section, the WMV criterion is used to compare the simultaneous consideration of the location-dependent sampling costs and the robot's travel costs with the separate consideration of each of the two cost types. The WMV criterion without cost consideration is also examined as a baseline. The experimental setup is described in Sec. 5.2. We used PoLC-ModMM to generate a cost-aware space-filling initial design of 16 sample points as a basis for the adaptive designs, assuming the cost function to be known. The following parameters were used for the algorithms:

PoLC-ModMM: $\lambda = 0.3$, no weight on the sampling criterion in the scalarization function

WMV: $\alpha = 0.025$ (in all methods)

PoTC-WMV: $\lambda = 0.1$, no weight on the sampling criterion in the scalarization function

PoLC-WMV: $\lambda = 0.3$, no weight on the sampling criterion in the scalarization function

RHCaAS-PoLTC-WMV: $H = 4$, $\lambda = 0.3$, in the scalarization function the weight for the location-dependent sampling cost is set to 0.9 and the weight for the travel cost is set to 0.1.

In Fig. 5.11 it can be seen that WMV results in a lower NRMSE than the cost-aware methods. The cost-aware methods have a comparable NRMSE, with PoLC-WMV having the highest NRMSE. RHCaAS-PoLTC-WMV and PoLC-WMV incur comparable NMCPS and Rel. MCTCS (relative mean cumulative travel cost savings), with the Rel. MCTCS computed relative to WMV. The same applies to RHCaAS-PoTC-WMV and WMV. Accordingly, RHCaAS-PoLTC-WMV provides comparable results to taking one of the cost types into account separately without additional loss of model accuracy. This underscores its potential as an effective method for simultaneously addressing both location-dependent sampling costs and travel costs of the mobile robot with minimal impact on model accuracy.

5.6 Summary

In this chapter we introduced novel methods that account for location-dependent costs and travel costs, advancing the methods from Chap. 3 and Chap. 4 for their application in environmental sensing using mobile robots. For this purpose, we first investigated the consideration of travel costs and location-dependent costs separately in Sec. 5.3 and Sec. 5.4. In both cases, we set up an MOP that included the sampling criterion and the considered cost function. By applying a scalarization function, we transformed each of the MOPs into an SOP, which enables a trade-off between the two objective functions and guarantees the position on the Pareto front. Furthermore, motivated by the CEI criterion [97], we formulated additional constraints for the optimization of the scalarization function, which only allow solutions in regions of the design space where the sampling criterion takes values above a threshold, where the threshold is a tuning parameter. The corresponding family of Pareto-optimal cost-aware criteria is denoted φ^{PoTC} or φ^{PoLC} in case of consideration of travel costs and location-dependent costs, respectively.

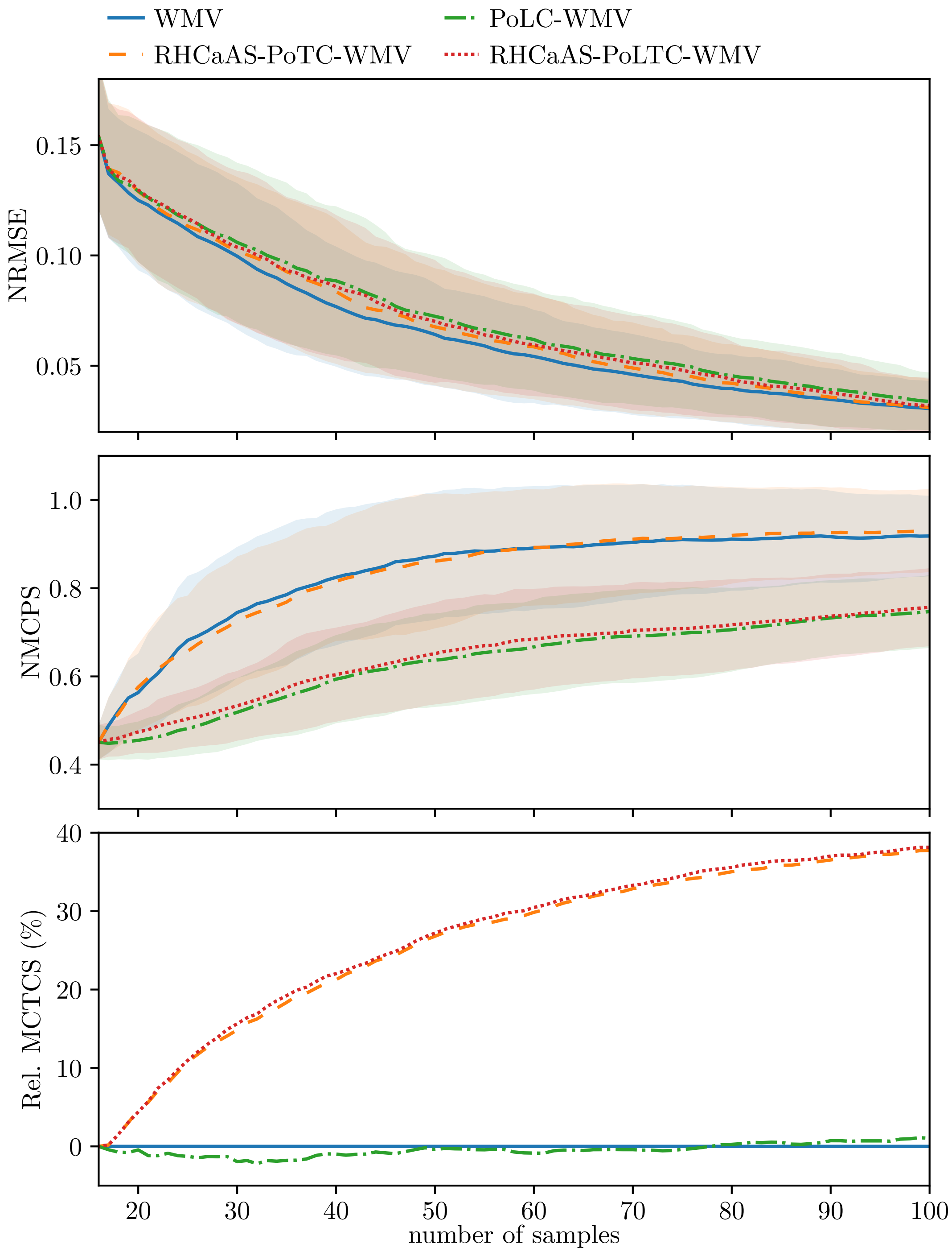

Figure 5.11: The mean (lines) and standard deviation (shaded area) of the measures NRMSE, NMCPS, and Rel. MCTCS over $N^{\text{runs}} = 100$ randomly generated black-box and cost function pairs vs the number of sample points. The cost function is known to the sampling algorithms.

In Sec. 5.4.3, our simulations demonstrated that consideration of location-dependent sampling costs in the Pareto-optimal criterion φ^{PoLC} drastically reduces the cumulative costs without having a significant impact on metamodel accuracy. We have also considered the scenario where the location-dependent sampling costs are not known in advance and must be approximated by a cost model. The simulations demonstrated that reliance on the error-prone cost model in the optimization process reduces cost savings, thereby diminishing the achieved model accuracy per cost incurred. Therefore, it is advisable to incorporate any prior knowledge of the cost function into the cost model.

The Pareto-optimal criterion φ^{PoTC} for consideration of the robot's travel cost did not perform well in simulation in Sec. 5.3.4, as cost reduction is achieved by selecting the next sample point close to the robot's current position. Hence, significant consideration of travel costs results in the robot barely moving from its current position.

In addition to balancing the criterion and the cost function, in Sec. 5.3 we presented a novel cost-aware adaptive sampling (RHCaAS) algorithm that predicts the robot's cumulative travel costs over a horizon. The next sample point is in each iteration obtained by minimization of the predicted cumulative travel costs. Unlike the Pareto-optimal approach, the sample points in each iteration are selected in the (predicted) optimum of the sampling criterion and thus no trade-off has to be made, resulting in drastic cost savings without significant impact on model accuracy. Large horizon lengths result in initial losses in global model accuracy, as the robot engages in local exploitation along its route, leading to uneven exploitation across the design space.

RHCaAS is independent of the sampling criterion used, allowing the use of Pareto-optimal cost-aware sampling criteria. This enabled us to use RHCaAS in combination with PoLC in Sec. 5.5, which led to greater cost savings for small horizon lengths without negatively impacting model accuracy. Moreover, we exploited this property to use RHCaAS in combination with the Pareto-optimal criterion φ^{PoLTC}, which trades-off between the location-dependent sampling costs and the travel costs. In simulation, this resulted in a drastic reduction of the cumulative location-dependent sampling costs and the cumulative travel costs of the mobile robot without significant impact on model accuracy.

CHAPTER 6

Conclusions

Rapidly advancing climate change and increasing environmental pollution require solutions for sensing and monitoring the environment, enabling the implementation of targeted and effective interventions. Besides satellite data (e.g., high resolution imagery and radar data) and stationary sensors such as weather stations, there is an increasing deployment of mobile robots for this purpose. A prime example is the ROBDEKON project [14], in which an autonomous construction machine is used to perform expensive dynamic probing on a contaminated site, on the basis of which the distribution of pollutants in the soil is then to be reconstructed. This example underscores that in certain applications, each individual measurement incurs significant costs (e.g., time and/or energy demands), necessitating the employment of data-efficient machine learning (ML) methods for modeling environmental phenomena, coupled with intelligent strategies for the selection of informative sample points.

As described in Chap. 1, this thesis initially develops space-filling and adaptive sampling methods, specifically designed for environmental sensing of expensive-to-evaluate environmental phenomena. Subsequently, cost-aware sampling strategies are developed, taking into account the location-dependency of sampling costs and the travel costs of the mobile robot. These strategies build on the methods previously introduced to achieve the most accurate reconstruction of the environmental phenomenon within a given budget.

In the further course of the chapter, Sec. 6.1 first summarizes the contents of this thesis. An outlook on future research opportunities is then given in Sec. 6.2.

6.1 Summary

In Chap. 3, we investigated the metamodeling of an unknown function (black-box function) using space-filling designs. Existing methods in the domain of metamodeling address mixed-domain (continuous, discrete, categorical, various scales, etc.) and often high-dimensional design spaces, as well as black-box functions that may exhibit complex behaviors, including discontinuities. We exploited the fact that environmental sensing typically involves a low-dimensional design space with uniformly scaled dimensions. Furthermore, motivated by characteristics of environmental phenomena (e.g., it is subject to diffusion processes), we assumed that the black-box function is Lipschitz continuous, implying a bounded rate of change. Building on these premises, we derived three novel sampling criteria: mean distance (MD), root mean squared distance (RMSD), and modified maximin (mod. maximin). For each criterion, we defined both a one-shot and a sequential variant. Additionally, we introduced modified regular grid sampling (mod. RGS), which constructs an inward-shifted regular grid. Unlike factorial designs, the proposed methods enable the generation of designs with any number of samples. Our simulation results demonstrated three key findings: First, mod. RGS performed best in terms of global metamodel error. Second, our proposed methods produced comparable results to mod. RGS but have the advantage of being usable for any number of samples. Third, our methods outperformed other space-filling designs from the literature.

In Chap. 4, we applied adaptive sampling methods. These methods iteratively select the next sample point(s) based on the available information about the black-box function, leading to improved model accuracy. Initially, we defined the class of explorative sampling criteria (ESC) and showed that most distance-based or variance-based sampling criteria can be easily transformed into an ESC. Building on ESC, we introduced the novel class of weighted explorative sampling criteria (WESC), specifically developed for sensing positive-valued environmental phenomena such as concentration distributions. Using two concrete WESCs, we showed that these outperform existing methods in simulation when used for metamodeling of positive-valued environmental phenomena.

Chap. 3 and Chap. 4 focused on the optimal placement of sample points w.r.t. global model accuracy. However, in practical scenarios, sampling costs can be location-dependent. Additionally, the use of mobile robots introduces travel costs, which are significantly influenced by the sequence of sample points. Hence, in Chap. 5, we investigated cost-aware sampling methods that take these costs into account, thereby achieving an increase in model accuracy for a given budget. For this purpose, we developed a Pareto-optimal sampling criterion for each scenario: considering location-dependent costs, travel costs, and both costs combined. Moreover, we introduced a receding horizon cost-aware adaptive sampling (RHCaAS) algorithm, which drastically reduces travel costs without significantly impacting metamodel accuracy. Finally, we combined the Pareto-optimal criterion with the RHCaAS algorithm in such a way that a significant reduction in both types of costs could be achieved with minimal impact on model accuracy. The proposed algorithms were evaluated through extensive simulations, demonstrating their superiority over existing methods.

The criteria presented in this work enable a superior selection of sample points for the reconstruction of environmental phenomena concerning information yield compared to existing methods. In conjunction with the methods developed in Chap. 5 for considering location-dependent sampling costs and the robot's travel costs, the results of this work have the potential for significant cost savings in the deployment of mobile robots for sensing of expensive-to-evaluate environmental phenomena.

6.2 Future Research

As detailed in Sec. 6.1, this thesis was concerned with metamodeling of expensive-to-evaluate black-box functions, with specific application to environmental sensing. A significant emphasis was placed on the incorporation of location-dependent sampling costs and the travel costs associated with the deployment of mobile robots for sample collection. In this section, we provide an outlook on promising avenues for future research. We propose connections to state-of-the-art ML techniques, which hold the potential to overcome several limitations of existing methods.

Reinforcement Learning Reinforcement learning (RL) presents a promising alternative for determining the next sample point in sequential or adaptive sampling methods. Instead of optimizing a sampling criterion, RL enables the learning of a policy (the RL agent) that suggests the next sample point based on the existing design [98]. Current research in this emerging and active field primarily focuses on Bayesian approaches to maximize the expected information gain associated with the next sample point [98–100]. To avoid the necessity of retraining the metamodel in each iteration of the algorithm, some approaches consider the metamodel as a black-box, which is solely used for training the policy. Thereby the characteristics of the black-box function are learned implicitly. When applied, the next sample point is then generated based on the existing design.

RL requires that the transitions between successive states can be modeled as a Markov decision process (MDP) [101]. This can be achieved by transforming the pairs of sample and sample point into an embedding space of fixed size, where the current design (the state) in the embedding space is recursively defined as the sum of the embedding of the previous design and the embedding of the new data point (sample point, sample) [98].

We consider the use of RL for adaptive sampling to be promising and propose the following research questions that we consider interesting for further exploration:

> **Utilization of Novel AI Technologies for Handling Dynamic State Sizes**: RL on graph neural networks (GNNs), as well as its integration with transformer models, represent novel and active fields of research (e.g., for

collaboration of multiple agents [102] (e.g., swarm of robots), solving routing problems in graphs [103], or efficiently encode multi-modal high-dimensional data into low-dimensional embeddings [104]). We suggest exploring whether technologies such as GNNs or transformer models are suitable for RL-based sequential or adaptive sampling involving dynamic state sizes, as they can process inputs of varying lengths. For instance, each sample point could be represented as a node of a GNN, with the corresponding sample as its attribute, and the distances between the sample points encoded in the edges of the graph.

Use of Application-specific Metamodels: Rather than implicitly training the characteristics of the metamodel into the models used (e.g., encoder, policy), we propose investigating whether an application-specific metamodel, optimized through careful selection of parameters (e.g., hyperparameters and prior of a Gaussian process) to achieve high model accuracy with little data, can enhance the data efficiency of the RL algorithm. Furthermore, it should be examined whether the policy generalizes better across different applications when application-specific knowledge is explicitly included in the metamodel instead of the policy and encoder models. This leads to another interesting question: How can the state of the metamodel be represented and integrated with the existing design as a state input to the policy?

Distance-based Space-filling Sequential Designs: These represent a special case as they do not use a metamodel for selecting the next sample point. This category includes designs such as the minimax design, which is particularly challenging to optimize. We consider investigating an RL policy capable of generating optimal minimax samples to be highly beneficial.

Optimal Initial Sample Point in Sequential Sampling As illustrated in Fig. 2.14, sequential designs are strongly dependent on the choice of the first sample point. Although this fact is regularly mentioned in the literature, no solution is proposed. Therefore, we consider the investigation of strategies for the optimal selection of initial sample points for various sequential sampling criteria to be an important next step.

Optimal Initial Design for Adaptive Sampling The same observation noted in the selection of the initial sample point for sequential designs is evident in the choice of initial designs for adaptive sampling methods. The size of the initial design and the need for its space-filling characteristics are discussed a lot in the literature. However, it is unclear how the type of initial design effects the resulting adaptive design including its implications on model accuracy. Therefore, we propose a closer examination of how the characteristics of adaptive designs depend on their initial setups, including the implications for model accuracy.

Avoidence of Initial Design through Multi-criteria Sampling Adaptive sampling methods use existing information about the black-box function in each iteration to determine the next sample point. Since no information about the black-box function is available initially, a space-filling initial design is often employed to gather data from various regions of the design space. This data forms the basis for performing adaptive sampling. The choice and size of the initial design are parameters that must be specified by the user. Some approaches integrate exploration and exploitation to avoid this distinction. Essentially, these approaches can be divided into three categories [15]:

1. **Decreasing strategy**: A weighted sum is formed between an explorative and an exploitative criterion. As the algorithm iterates, the weight assigned to exploration decreases while the weight assigned to exploitation increases.

2. **Greedy strategy**: Beginning with pure exploration, a switching condition is employed across the iterations of the algorithm to transition between pure exploration and pure exploitation.

3. **Switching strategy**: A weighted sum is formed between an explorative and an exploitative criterion. The weights are dynamically and inversely adjusted (e.g., in a sinusoidal pattern), ensuring that either exploration or exploitation predominates over the iterations.

Note that the weighted sum serves as a scalarization function, and the weights influence the positioning of the solution on the Pareto front. We believe that exploring more sophisticated multi-criteria adaptive sampling criteria, which combine multiple sampling criteria with varying characteristics (e.g., through multi-criteria optimization or machine learning techniques), holds significant potential. This is especially true for the deployment of mobile robots, as their travel distance could be substantially reduced if, starting from an initial point, it could be decided during the exploration of the design space that a location requires closer examination, rather than first exploring the entire design space and subsequently investigating specific locations in detail.

Parallelization of Sampling Approaches By employing methods such as delayed feedback sampling (or its analog for Voronoi-based techniques [71]) or Thompson sampling, adaptive sampling methods can propose multiple next sample points simultaneously [67]. This allows for the parallel (or batch) evaluation of the black-box function at these points, thereby enhancing efficiency. To facilitate parallel sampling in environmental sensing, multiple mobile robots are required. Considerable research is directed towards collaboration between multiple robots (robot swarms) [105–107], including their application in environmental sensing and monitoring [108–110].

However, as described in Chap. 5, existing methods are only partially suitable for sensing expensive-to-evaluate phenomena. Therefore, we believe it is worthwhile to develop methods that are suited for cost-aware sensing of expensive-to-evaluate environmental phenomena using robot swarms.

Cost-aware Space-filling One-shot Design In Sec. 5.4, we have investigated cost-aware sequential space-filling designs. Furthermore, in Chap. 3, it was noted that one-shot designs demonstrate superior space-filling properties compared to space-filling designs. Accordingly, we propose the exploration of multi-objective optimization for generating cost-aware space-filling designs. Note that this approach requires prior knowledge of the cost function.

Time-variant Black-box Funcitons Mobile robots are also employed for spatiotemporal sampling of time-varying environmental phenomena [111, 112] and methods for regression of spatiotemporal data have been proposed [113–115]. Similar to the methods discussed in Chap. 5, data is collected during the robots motion. We consider the extension of the methods from this thesis for suitability in spatiotemporal sampling of time-varying environmental phenomena to be an exciting research direction.

Generative AI for Sample Generation Generally speaking, generative AI comprises methods that can learn distributions from data and then generate new data instances (not contained in the training data) following the same distribution [116–119]. Moreover, generative AI methods can be conditioned to influence the output [120, 121]. We consider the exploration of the suitability of generative AI for generating one-shot or adaptive designs to be a fascinating research direction. For example, it could be examined to what extent existing samples and the sampling method used can be accounted for through conditioning.

Transfer to Other Applications Community sensing explores how a group of people, such as the residents of a city, can be used to monitor specific metrics. The methods developed in this thesis for cost-aware space-filling and adaptive sampling, combined with batch sampling techniques, hold the potential to maximize the amount of information obtained per measurement for a given budget, particularly when samples are expensive in terms of effort, duration, or payment to the individual.

Another potential application is quality monitoring in production lines. In some cases, quality assessment can be time-consuming, such as when a large component must be removed with a crane to evaluate its quality characteristics on a test bench. The methods developed in this thesis could be used to create a metamodel of the quality characteristics based on features of the process data and use a cost-aware adaptive sampling criterion to decide whether a component should be removed for quality assessment. This approach could prevent unnecessary removals when the model confidently predicts the component's quality as satisfactory.

The cost-aware adaptive sampling methods presented in this thesis should, from our perspective, be evaluated for their suitability for use in robot-based extraterrestrial observation, such as with Mars rovers. Such robots operate under constrained resources (time, energy), and individual samples can be expensive depending on the metric being assessed (wear and tear, measurement capacity, energy consumption, duration).

APPENDIX A

Contextualization of Our Research within Current Developments in Artificial Intelligence

Machine learning (ML) methods have become indispensable in the modern era. Recent years have been marked by significant breakthroughs in a variety of ML methods across numerous application domains. The trend is evident: more data and larger models yield better results [122–124]. To date, no limits have been identified beyond which additional data and larger models do not continue to enhance performance. However, the resources freely available on the internet are limited [123, 125], prompting artificial intelligence (AI) providers to actively seek partnerships and licensing agreements with content holders [126, 127]. The challenges are not limited to the quantity and quality of the training data, but also extend to the associated computational resources [128]. These are coupled with high energy demands and considerable time expenditure [129–131]. Consequently, there is currently significant research effort being directed towards various means to enhance the efficiency of computing units [131–134], optimize technical implementations [123, 135, 136], and develop innovative methods [131, 137] that allow for more efficient solutions to achieve comparable results.

According to current estimates, the deployment of AI will determine the competitiveness of enterprises in the future and will be utilized across all societal sectors and industries [135, 138, 139]. In this context, enhancing the efficiency of AI methods is of strategic importance for humanity. This is essential to prevent or at least mitigate the negative environmental impacts of AI [128, 129, 131].

In practice, particularly within the industrial context, the application of AI for specific purposes is often constrained by the availability of data. A solution to this issue is transfer learning, in which models are pre-trained (e.g., through a multi-stage process involving self-supervised, then semi-supervised, then supervised learning) on large datasets with comparable characteristics (such as their probabilistic distribution) and

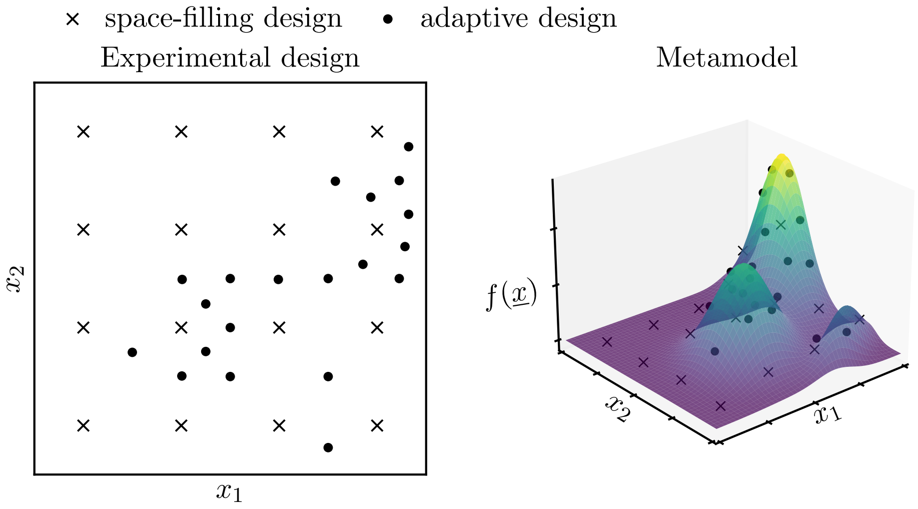

Figure A.1: Metamodeling of an unknown function $f(\underline{x})$ (black-box function) involving a space-filling initial design (crosses) and an adaptive design (dots) on a two-dimensional parameter space (design space). The adaptive design iteratively exploits the existing knowledge about the black-box function to intensify sampling in regions of interest.

subsequently fine-tuned on small, application-specific datasets. Examples include the analysis of MRI or CT scans [140, 141], the segmentation and classification of camera data in autonomous driving [142], land use classification based on satellite data [2, 3], and the assessment of environmental damage using image data [1, 4].

The size of the model, especially in the context of embedded AI [135], poses limitations when AI is implemented on microcontrollers for integration into products that have restricted space, energy, or memory capacities [143–146]. This often requires the use of specialized techniques for model compression and optimization, which may include application-specific logics based on expert knowledge [147].

Reinforcement learning (RL) is a machine learning technique wherein an agent learns to maximize a reward function over the long term through interaction with its environment [101]. The agent performs actions based on the state of its environment, which requires learning through trial and error. This makes the generation of sufficient training data more challenging, expensive, and thus unfeasible for many practical applications compared to other ML methods [148, 149]. Such difficulty arises particularly because an RL agent requires numerous trials to learn an effective strategy. Consequently, a variety of methods are being explored and ingeniously combined to enhance the data efficiency of RL [148, 150]. Examples include collecting past experiences of an agent in a memory and randomly replaying them to stabilize the learning process and maximize the use of available data (experience replay) [151], transferring knowledge from a previously trained model to a new problem (transfer learning) [152], progressively increasing the difficulty of the agent's tasks for

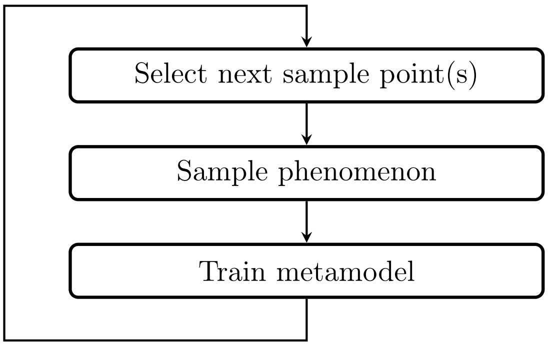

Figure A.2: Simplified adaptive sampling procedure for environmental sensing.

more effective and faster convergence (curriculum learning) [153], encoding desired behaviors into the reward function (reward shaping) [154], or reinterpreting an agent's failures as successes of an alternative reward function (hindsight experience replay) [155]. In some cases, off-policy learning can also be employed to learn from the experiences of existing solutions, other agents, or past experiences [101, 156, 157].

Another approach is model-based RL, which, similar to model predictive control, explicitly incorporates a model of the environment (which can be learned from data, predefined, or a combination of both) [158–160]. Model-based RL illustrates a fundamental issue common to all data-driven learning methods: the inefficient representation of the physical relationships in the environment by ML models [161, 162].

Physics-informed ML addresses this issue by trying to learn low-dimensional physical representations from high-dimensional data [161, 163]. This approach focuses on the design of the model architecture and the loss function to accurately represent the low-dimensional (and often sparse) physical relationships [164, 165]. The term "physics" in this context is abstracted to encompass all laws inherent in the data. However, physics-informed ML is still in its infancy and, to date, has been practically applied only in specialized cases [166].

Instead of physically modeling the environment in full detail, metamodels – often referred to as digital twins [19] – are commonly used to represent the relationships between target variables (outputs) and parameters (inputs) in physical experiments (e.g., wind tunnel tests [17]) and computer experiments (e.g., numerical simulations [18]) [15, 16]. The rapidly evaluable metamodel mimics the physical relationships across the entire parameter space and serves as a surrogate for the experiments [21]. In this process, active learning (also known as adaptive sampling [21]) techniques are utilized to select the parameter combinations for training the metamodel in such a way that it achieves maximum model accuracy or identifies the optimum of the underlying causal relationships [15, 25], as shown in Fig. A.1. In some cases, cost-aware methods are applied, aiming to achieve the best possible results within a given budget by considering the parameter-dependent experimental costs [17, 26].

Thanks to advances in mobile robotics, battery technology, and sensor technology, adaptive sampling methods are becoming increasingly important in environmental sensing. As illustrated in Fig. A.2, the adaptive sampling procedure iterates by selecting the next sample point, sampling the environmental phenomenon at the selected point, and then training the new data points into the metamodel. This sequential approach allows for the selection of subsequent sample points to be adapted based on the most recent knowledge about the phenomenon, thereby gaining more information per sample than would be obtained using non-adaptive sampling methods (e.g., distance-based space-filling designs [21, 27]) [15].

Adaptive sampling was employed in environmental sensing and monitoring for the exploration and reconstruction of scalar fields (such as depth maps or concentration distributions) using unmanned surface vehicles (USV) [28–30]. For this, real-time capable sensors were employed, which recorded measurements at regular intervals along the robot's route and while it was in motion. However, there are also applications where the robot has to interrupt its motion to perform expensive sampling (in terms of energy and/or time) and the number of samples is limited by constrained resources (e.g., battery capacity, time limit).

A prominent example is the ROBDEKON project [14], which investigates the use of autonomous construction machines for the decontamination of contaminated sites. The autonomous construction machine performs expensive and time-consuming dynamic probing to collect samples, which are then used to reconstruct the unknown distribution of pollutants in the soil. This reconstruction allows for the precise identification and removal of regions exceeding legal contamination limits, ensuring effective site decontamination.

The project illustrates that there may only be a limited number of expensive samples available for modeling the phenomenon, necessitating the preference for data-efficient methods over large artificial intelligence (AI) models, such as those e.g., used in satellite data processing. Additionally, there may be location-dependent differences in sampling costs due to variations in soil composition, slopes, vegetation, or even obstacles. Furthermore, travel costs incurred by the mobile robot between sample points can vary significantly depending on the sequence of the points.

In contrast, in the context of global metamodeling, (adaptive) sample points are chosen to maximize the information gained about the black-box function per sample. Furthermore, global metamodeling typically addresses high-dimensional mixed-domain (continuous, discrete, categorical, various scalings) parameter spaces and highly nonlinear or even discontinuous black-box functions (e.g., with jumps) [27].

However, in this thesis we consider the scenario of environmental sensing motivated by the ROBDEKON project, where a continuous black-box function (e.g., a concentration distribution) is to be reconstructed on the (low-dimensional and equally scaled) location space based on expensive samples. In addition, incurred costs are to be taken into account which include the robot's travel costs as well as the potentially location-dependent sampling costs.

Analogous to the concept of the digital twin, this thesis focuses on the digital representation of unknown environmental phenomena based on costly samples. In contrast to existing methods, we examine the problem of global metamodeling under the constraints of environmental sensing with mobile robots. By considering the characteristics of the spatial domain, environmental phenomena, and the costs associated with deploying a mobile robot, we have succeeded in developing methods at the conjunction of global metamodeling and mobile robotics for environmental sensing that significantly outperform existing approaches from both fields in this scenario.

Bibliography

[1] Lei Ma et al. "Deep Learning in Remote Sensing Applications: A Meta-Analysis and Review". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 152 (June 2019), pp. 166–177. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2019.04.015. (Visited on 04/18/2024).

[2] Deyuan Zhang, Zhenghong Liu, and Xiangbin Shi. "Transfer Learning on EfficientNet for Remote Sensing Image Classification". In: *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*. Harbin, China: IEEE, Dec. 2020, pp. 2255–2258. ISBN: 978-1-66542-314-4. DOI: 10.1109/ICMCCE51767.2020.00489. (Visited on 04/18/2024).

[3] Yuchi Ma, Shuo Chen, Stefano Ermon, and David B. Lobell. "Transfer Learning in Environmental Remote Sensing". In: *Remote Sensing of Environment* 301 (2024), p. 113924. ISSN: 0034-4257. DOI: 10.1016/j.rse.2023.113924.

[4] Yassine Himeur, Bhagawat Rimal, Abhishek Tiwary, and Abbes Amira. "Using Artificial Intelligence and Data Fusion for Environmental Monitoring: A Review and Future Perspectives". In: *Information Fusion* 86–87 (2022), pp. 44–75. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2022.06.003.

[5] Silvia Nittel. "A Survey of Geosensor Networks: Advances in Dynamic Environmental Monitoring". In: *Sensors* 9.7 (July 2009), pp. 5664–5678. ISSN: 1424-8220. DOI: 10.3390/s90705664. (Visited on 05/17/2024).

[6] Suprava Ranjan Laha, Binod Kumar Pattanayak, and Saumendra Pattnaik. "Advancement of Environmental Monitoring System Using IoT and Sensor: A Comprehensive Analysis". In: *AIMS Environmental Science* 9.6 (2022), pp. 771–800. ISSN: 2372-0352. DOI: 10.3934/environsci.2022044. (Visited on 05/17/2024).

[7] Nitin Agarwala. "Monitoring the Ocean Environment Using Robotic Systems: Advancements, Trends, and Challenges". In: *Marine Technology Society Journal* 54.5 (Sept. 2020), pp. 42–60. ISSN: 0025-3324. DOI: 10.4031/MTSJ.54.5.7. (Visited on 05/17/2024).

[8] Bin Liu, Jinyin Zhuang, and Gang Wei. "Recent Advances in the Design of Colorimetric Sensors for Environmental Monitoring". In: *Environmental Science: Nano* 7.8 (2020), pp. 2195–2213. DOI: 10.1039/D0EN00449A.

[9] Jorge Sánchez-Sosa, Juan Castillo-Mixcóatl, Georgina Beltrán-Pérez, and Severino Muñoz-Aguirre. "An Application of the Gaussian Plume Model to Localization of an Indoor Gas Source with a Mobile Robot". In: *Sensors* 18.12 (2018), p. 4375. ISSN: 1424-8220. DOI: 10.3390/s18124375.

[10] Theerapat Pobkrut and Teerakiat Kerdcharoen. "Soil Sensing Survey Robots Based on Electronic Nose". In: *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*. Gyeonggi-do, South Korea: IEEE, 2014, pp. 1604–1609. ISBN: 978-89-93215-07-6 978-89-93215-06-9. DOI: 10.1109/ICCAS.2014.6987829.

[11] Devin Bourgeois, Anu G. Bourgeois, and Ashwin Ashok. "Demo: RoSS: A Low-Cost Portable Mobile Robot for Soil Health Sensing". In: *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. Bangalore, India: IEEE, 2022, pp. 436–437. ISBN: 978-1-66542-104-1. DOI: 10.1109/COMSNETS53615.2022.9668355.

[12] Yanwu Zhang et al. "Targeted Sampling by Autonomous Underwater Vehicles". In: *Frontiers in Marine Science* 6 (2019), p. 415. ISSN: 2296-7745. DOI: 10.3389/fmars.2019.00415.

[13] Yanwu Zhang, Robert S. McEwen, John P. Ryan, and James G. Bellingham. "An Adaptive Triggering Method for Capturing Peak Samples in a Thin Phytoplankton Layer by an Autonomous Underwater Vehicle". In: *OCEANS 2009*. Biloxi, MS: IEEE, 2009, pp. 1–5. ISBN: 978-1-4244-4960-6. DOI: 10.23919/OCEANS.2009.5422436.

[14] J. Petereit et al. "ROBDEKON: Robotic Systems for Decontamination in Hazardous Environments". In: *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. Sept. 2019.

[15] Jan N. Fuhg, Amélie Fau, and Udo Nackenhorst. "State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging". en. In: *Archives of Computational Methods in Engineering* 28.4 (2021), pp. 2689–2747. ISSN: 1134-3060, 1886-1784. DOI: 10.1007/s11831-020-09474-6.

[16] Haitao Liu, Yew-Soon Ong, and Jianfei Cai. "A Survey of Adaptive Sampling for Global Metamodeling in Support of Simulation-Based Complex Engineering Design". en. In: *Structural and Multidisciplinary Optimization* 57.1 (2018), pp. 393–416. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-017-1739-8.

[17] Uihwan Choi, Junhong Kim, and Jaemyung Ahn. "Cost-Aware Adaptive Design of Experiment with Nonstationary Surrogate Model for Wind Tunnel Testing". en. In: *International Journal of Aeronautical and Space Sciences* 21.3 (2020), pp. 670–680. ISSN: 2093-274X, 2093-2480. DOI: 10.1007/s42405-020-00250-1.

[18] T. Braconnier, M. Ferrier, J.-C. Jouhaud, M. Montagnac, and P. Sagaut. "Towards an Adaptive POD/SVD Surrogate Model for Aeronautic Design". In: *Computers & Fluids* 40.1 (2011), pp. 195–209. ISSN: 00457930. DOI: 10.1016/j.compfluid.2010.09.002.

[19] Sylvain Chabanet, Hind Bril El-Haouzi, and Philippe Thomas. "Coupling Digital Simulation and Machine Learning Metamode through an Active Learning Approach in Industry 4.0 Context". In: *Computers in Industry* 133 (Dec. 2021), p. 103529. ISSN: 01663615. DOI: 10.1016/j.compind.2021.103529. (Visited on 04/20/2024).

[20] Flavia Pires, Ana Cachada, Jose Barbosa, Antonio Paulo Moreira, and Paulo Leitao. "Digital Twin in Industry 4.0: Technologies, Applications and Challenges". In: *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*. Helsinki, Finland: IEEE, July 2019, pp. 721–726. ISBN: 978-1-72812-927-3. DOI: 10.1109/INDIN41052.2019.8972134. (Visited on 05/20/2024).

[21] K. Crombecq, I. Couckuyt, D. Gorissen, and T. Dhaene. "Space-Filling Sequential Design Strategies for Adaptive Surrogate Modelling". In: *The First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*. Funchal, Madeira, Portugal, 2009.

[22] Ruichen Jin, Wei Chen, and Agus Sudjianto. "On Sequential Sampling for Global Metamodeling in Engineering Design". In: *Volume 2: 28th Design Automation Conference*. Montreal, Quebec, Canada, 2002.

[23] Karel Crombecq, Dirk Gorissen, Dirk Deschrijver, and Tom Dhaene. "A Novel Hybrid Sequential Design Strategy for Global Surrogate Modeling of Computer Experiments". en. In: *SIAM Journal on Scientific Computing* 33.4 (2011), pp. 1948–1974. ISSN: 1064-8275, 1095-7197. DOI: 10.1137/090761811.

[24] Han-Lim Choi, Jaemyung Ahn, and Doo-Hyun Cho. "Information-Maximizing Adaptive Design of Experiments for Wind Tunnel Testing". en. In: *Engineering Optimization 2014*. Ed. by Aurelio Araujo. CRC Press, 2014, pp. 329–334. ISBN: 978-1-138-02725-1 978-1-315-73210-7. DOI: 10.1201/b17488-59.

[25] Eric Hans Lee, Valerio Perrone, Cedric Archambeau, and Matthias Seeger. *Cost-Aware Bayesian Optimization*. en. 2020. arXiv: 2003.10870 [cs, stat].

[26] Phuc Luong, Dang Nguyen, Sunil Gupta, Santu Rana, and Svetha Venkatesh. "Adaptive Cost-Aware Bayesian Optimization". In: *Knowledge-Based Systems* 232 (2021), p. 107481. ISSN: 09507051. DOI: 10.1016/j.knosys.2021.107481.

[27] K. Crombecq, E. Laermans, and T. Dhaene. "Efficient Space-Filling and Non-Collapsing Sequential Design Strategies for Simulation-Based Modeling". en. In: *European Journal of Operational Research* 214.3 (2011), pp. 683–696. ISSN: 03772217. DOI: 10.1016/j.ejor.2011.05.032.

[28] Yew Teck Tan, Abhinav Kunapareddy, and Marin Kobilarov. "Gaussian Process Adaptive Sampling Using the Cross-Entropy Method for Environmental Sensing and Monitoring". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, 2018, pp. 6220–6227. ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8460821.

[29] Bin Zhang and Gaurav S. Sukhatme. "Adaptive Sampling for Estimating a Scalar Field Using a Robotic Boat and a Sensor Network". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. Rome, Italy: IEEE, 2007, pp. 3673–3680. ISBN: 978-1-4244-0602-9 978-1-4244-0601-2. DOI: 10.1109/ROBOT.2007.364041.

[30] Paul Stankiewicz, Yew T. Tan, and Marin Kobilarov. "Adaptive Sampling with an Autonomous Underwater Vehicle in Static Marine Environments". In: *Journal of Field Robotics* 38.4 (2021), pp. 572–597. ISSN: 1556-4959, 1556-4967. DOI: 10.1002/rob.22005.

[31] Johannes Westermann, Antonio Zea, and Uwe D. Hanebeck. "Adaptive Sampling for Global Meta Modeling Using a Gaussian Process Variance Measure". In: *Proceedings of the 2021 European Control Conference (ECC 2021)*. Virtual, 2021.

[32] Johannes Westermann and Lucas Alber. "Cost-aware Adaptive Sampling for Global Meta Modeling Using a Voronoi Tessellation". In: *Proceedings of the 2022 European Control Conference (ECC 2022)*. Virtual, 2022.

[33] Johannes Westermann, Jana Mayer, Janko Petereit, and Benjamin Noack. "Receding Horizon Cost-aware Adaptive Sampling for Environmental Monitoring". In: *Proceedings of the 2023 American Control Conference (ACC 2023)*. San Diego, CA, USA, 2023.

[34] Johannes Westermann, Jana Mayer, Janko Petereit, and Benjamin Noack. "Receding Horizon Cost-aware Adaptive Sampling for Environmental Monitoring". In: *IEEE Control Systems Letters* 7 (2023), pp. 1069–1074. DOI: 10.1109/LCSYS.2022.3230058.

[35] Maliki Moustapha, Bruno Sudret, Jean-Marc Bourinet, and Benoît Guillaume. "Metamodeling for Crashworthiness Design: Comparative Study of Kriging and Support Vector Regression". In: *Proceedings of the 2nd International Symposium on Uncertainty Quantification and Stochastic Modeling*. 2014, p. 9.

[36] Shengli Xu, Haitao Liu, Xiaofang Wang, and Xiaomo Jiang. "A Robust Error-Pursuing Sequential Sampling Approach for Global Metamodeling Based on Voronoi Diagram and Cross Validation". en. In: *Journal of Mechanical Design* 136.7 (2014), p. 071009. ISSN: 1050-0472, 1528-9001. DOI: 10.1115/1.4027161.

[37] Hao Tong, Changwu Huang, Jialin Liu, and Xin Yao. "Voronoi-Based Efficient Surrogate-Assisted Evolutionary Algorithm for Very Expensive Problems". In: *2019 IEEE Congress on Evolutionary Computation (CEC)*. 2019, pp. 1996–2003. DOI: 10.1109/CEC.2019.8789910.

[38] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. en. 3. print. Adaptive Computation and Machine Learning. Cambridge, Mass.: MIT Press, 2008. ISBN: 978-0-262-18253-9.

[39] Alberto Lovison and Enrico Rigoni. "Adaptive Sampling with a Lipschitz Criterion for Accurate Metamodeling". In: *Communications in Applied and Industrial Mathematics* 1.2 (2011), pp. 110–126. ISSN: 2038-0909. DOI: 10.1685/2010CAIM545.

[40] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. "Practical Bayesian Optimization of Machine Learning Algorithms". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems*. Vol. 2. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, 2951—2959.

[41] Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. "Safe Controller Optimization for Quadrotors with Gaussian Processes". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 491–496. DOI: 10.1109/ICRA.2016.7487170.

[42] Miao Liu, Girish Chowdhary, Bruno Castra da Silva, Shih-Yuan Liu, and Jonathan P. How. "Gaussian Processes for Learning and Control: A Tutorial with Examples". In: *IEEE Control Systems* 38.5 (2018), pp. 53–86. ISSN: 1066-033X, 1941-000X. DOI: 10.1109/MCS.2018.2851010.

[43] Marc Peter Deisenroth. "Efficient Reinforcement Learning Using Gaussian Processes". PhD thesis. KIT Scientific Publishing, 2010. ISBN: 978-3-86644-569-7. DOI: 10.5445/KSP/1000019799.

[44] Yaakov Engel. *Algorithms and Representations for Reinforcement Learning*. Hebrew University of Jerusalem, Israel. PhD thesis. Citeseer, 2005.

[45] Malte Kuss and Carl Rasmussen. “Gaussian Processes in Reinforcement Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 16. MIT Press, 2003.

[46] Varun Suryan and Pratap Tokekar. “Learning a Spatial Field with Gaussian Process Regression in Minimum Time”. In: *Algorithmic Foundations of Robotics XIII*. Ed. by Marco Morales, Lydia Tapia, Gildardo Sánchez-Ante, and Seth Hutchinson. Vol. 14. Cham: Springer International Publishing, 2020, pp. 301–317. ISBN: 978-3-030-44050-3 978-3-030-44051-0. DOI: 10.1007/978-3-030-44051-0_18.

[47] D.G. Krige. “A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand.” In: *Journal of the Chemical, Metallurgical and Mining Society of South Africa* 52.6 (1951), pp. 119–139.

[48] M. A. Oliver and R. Webster. “Kriging: A Method of Interpolation for Geographical Information Systems”. In: *International journal of geographical information systems* 4.3 (1990), pp. 313–332. ISSN: 0269-3798. DOI: 10.1080/02693799008941549.

[49] Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K. Sriperumbudur. *Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences*. 2018. arXiv: 1807.02582 [cs, stat].

[50] David Kristjanson Duvenaud. *Automatic Model Construction with Gaussian Processes (Doctoral thesis)*. 2014. DOI: 10.17863/CAM.14087.

[51] Jie Wang. “An Intuitive Tutorial to Gaussian Processes Regression”. In: *Computing in Science & Engineering* 25.4 (2023), 4–11. ISSN: 1558-366X. DOI: 10.1109/MCSE.2023.3342149. URL: http://dx.doi.org/10.1109/MCSE.2023.3342149.

[52] Henri Faure and Christiane Lemieux. “Improvements on the Star Discrepancy of (t,s)-Sequences”. In: *Acta Arithmetica* 154.1 (2012), pp. 61–78. ISSN: 0065-1036, 1730-6264. DOI: 10.4064/aa154-1-4.

[53] Colas Schretter, Leif Kobbelt, and Paul-Olivier Dehaye. “Golden Ratio Sequences for Low-Discrepancy Sampling”. In: *Journal of Graphics Tools* 16.2 (2012), pp. 95–104. ISSN: 2165-347X, 2165-3488. DOI: 10.1080/2165347X.2012.679555.

[54] Abdalla G. M. Ahmed et al. “Low-Discrepancy Blue Noise Sampling”. In: *ACM Transactions on Graphics* 35.6 (2016), pp. 1–13. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/2980179.2980218.

[55] Shuhei Kimura and Koki Matsumura. “Genetic Algorithms Using Low-Discrepancy Sequences”. In: *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*. Washington DC USA: ACM, 2005, pp. 1341–1346. ISBN: 978-1-59593-010-1. DOI: 10.1145/1068009.1068225.

[56] Harald Niederreiter. “Low-Discrepancy and Low-Dispersion Sequences”. In: *Journal of Number Theory* 30.1 (1988), pp. 51–70. ISSN: 0022314X. DOI: 10.1016/0022-314X(88)90025-X.

[57] Roswitha Hofer. “Generalized Hofer–Niederreiter Sequences and Their Discrepancy from an (U , e , s) -Point of View”. In: *Journal of Complexity* 31.2 (2015), pp. 260–276. ISSN: 0885064X. DOI: 10.1016/j.jco.2014.10.002.

[58] Tien-Tsin Wong, Wai-Shing Luk, and Pheng-Ann Heng. “Sampling with Hammersley and Halton Points”. In: *Journal of Graphics Tools* 2 (Jan. 1997). DOI: 10.1080/10867651.1997.10487471.

[59] Luc Pronzato. “Minimax and Maximin Space-Filling Designs: Some Properties and Methods for Construction”. In: *Journal de la Societe Française de Statistique* 158.1 (2017), pp. 7–36.

[60] Matthias H. Y. Tan. “Minimax Designs for Finite Design Regions”. In: *Technometrics* 55.3 (2013), pp. 346–358. ISSN: 0040-1706, 1537-2723. DOI: 10.1080/00401706.2013.804439.

[61] Lili Ju, Qiang Du, and Max Gunzburger. “Probabilistic Methods for Centroidal Voronoi Tessellations and Their Parallel Implementations”. In: *Parallel Computing* 28.10 (2002), pp. 1477–1500. ISSN: 01678191. DOI: 10.1016/S0167-8191(02)00151-5.

[62] James C. Hateley, Huayi Wei, and Long Chen. “Fast Methods for Computing Centroidal Voronoi Tessellations”. In: *Journal of Scientific Computing* 63.1 (2015), pp. 185–212. ISSN: 0885-7474, 1573-7691. DOI: 10.1007/s10915-014-9894-1.

[63] Qiang Du, Maria Emelianenko, and Lili Ju. “Convergence of the Lloyd Algorithm for Computing Centroidal Voronoi Tessellations”. In: *SIAM Journal on Numerical Analysis* 44.1 (Jan. 2006), pp. 102–119. ISSN: 0036-1429. DOI: 10.1137/040617364. (Visited on 11/24/2024).

[64] Qiang Du, Vance Faber, and Max Gunzburger. “Centroidal Voronoi Tessellations: Applications and Algorithms”. In: *SIAM Review* 41.4 (1999), pp. 637–676. ISSN: 0036-1445, 1095-7200. DOI: 10.1137/S0036144599352836.

[65] Jeong-Soo Park. “Optimal Latin-hypercube Designs for Computer Experiments”. In: *Journal of Statistical Planning and Inference* 39.1 (1994), pp. 95–111. ISSN: 0378-3758. DOI: 10.1016/0378-3758(94)90115-5.

[66] Felipe A. C. Viana, Gerhard Venter, and Vladimir Balabanov. “An Algorithm for Fast Optimal Latin Hypercube Design of Experiments: AN ALGORITHM FOR FAST OPTIMAL LHD”. In: *International Journal for Numerical Methods in Engineering* 82.2 (2010), pp. 135–156. ISSN: 00295981. DOI: 10.1002/nme.2750.

[67] Sayak Ray Chowdhury and Aditya Gopalan. “On Batch Bayesian Optimization”. In: (2019). arXiv: 1911.01032 [cs, stat].

[68] M. Rahimi, M. Hansen, W.J. Kaiser, G.S. Sukhatme, and D. Estrin. “Adaptive Sampling for Environmental Field Estimation Using Robotic Sensors”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Edmonton, Alta., Canada: IEEE, 2005, pp. 3692–3698. ISBN: 978-0-7803-8912-0. DOI: 10.1109/IROS.2005.1545070.

[69] Victor Picheny, David Ginsbourger, Olivier Roustant, Raphael T. Haftka, and Nam-Ho Kim. “Adaptive Designs of Experiments for Accurate Approximation of a Target Region”. In: *Journal of Mechanical Design* 132.7 (2010), p. 071008. ISSN: 1050-0472, 1528-9001. DOI: 10.1115/1.4001873.

[70] Bertrand Gauthier and Luc Pronzato. “Spectral Approximation of the IMSE Criterion for Optimal Designs in Kernel-Based Interpolation Models”. In: *SIAM/ASA Journal on Uncertainty Quantification* 2.1 (2014), pp. 805–825. ISSN: 2166-2525. DOI: 10.1137/130928534.

[71] Andrew L. Kaminsky, Yi Wang, and Kapil Pant. "An Efficient Batch K-Fold Cross-Validation Voronoi Adaptive Sampling Technique for Global Surrogate Modeling". en. In: *Journal of Mechanical Design* (2020), pp. 1–14. ISSN: 1050-0472, 1528-9001. DOI: 10.1115/1.4047155.

[72] Ruichen Jin, Wei Chen, and Agus Sudjianto. "An Efficient Algorithm for Constructing Optimal Design of Computer Experiments". In: *Journal of Statistical Planning and Inference* 134.1 (2005), pp. 268–287. ISSN: 0378-3758. DOI: 10.1016/j.jspi.2004.02.014.

[73] Tom Van Steenkiste, Joachim van der Herten, Dirk Deschrijver, and Tom Dhaene. "ALBATROS: Adaptive Line-Based Sampling Trajectories for Sequential Measurements". In: *Engineering with Computers* 35.2 (2019), pp. 537–550. ISSN: 0177-0667, 1435-5663. DOI: 10.1007/s00366-018-0614-6.

[74] Junghun Suh and Songhwai Oh. "Efficient Environmental Monitoring Using Cost-Aware Path Planning". In: *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*. Gwangju, Korea (South): IEEE, 2013, pp. 1362–1365. ISBN: 978-89-93215-05-2. DOI: 10.1109/ICCAS.2013.6704169.

[75] John J. Borkowski and Greg F. Piepel. "Uniform Designs for Highly Constrained Mixture Experiments". In: *Journal of Quality Technology* 41.1 (2009), pp. 35–47. ISSN: 0022-4065, 2575-6230. DOI: 10.1080/00224065.2009.11917758.

[76] *IDAES – Institute for the Design of Advanced Energy Systems*. https://idaes.org. (Visited on 10/05/2023).

[77] Paul Saves et al. *SMT 2.0: A Surrogate Modeling Toolbox with a Focus on Hierarchical and Mixed Variables Gaussian Processes*. 2023. DOI: https://doi.org/10.48550/arXiv.2305.13998. eprint: 2305.13998 (cs, math, stat).

[78] Tim Head, Manoj Kumar, Holger Nahrstaedt, Gilles Louppe, and Iaroslav Shcherbatyi. *scikit-optimize/scikit-optimize*. Version v0.9.0. 2021. DOI: 10.5281/zenodo.5565057. URL: https://doi.org/10.5281/zenodo.5565057.

[79] Julian Blank and Kalyanmoy Deb. "Pymoo: Multi-objective Optimization in Python". In: *IEEE Access* 8 (2020), pp. 89497–89509. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2990567.

[80] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17.3 (2020), pp. 261–272. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0686-2.

[81] Hossein Mohammadi, Peter Challenor, Daniel Williamson, and Marc Goodfellow. "Cross-Validation–based Adaptive Sampling for Gaussian Process Models". In: *SIAM/ASA Journal on Uncertainty Quantification* 10.1 (2022), pp. 294–316. ISSN: 2166-2525. DOI: 10.1137/21M1404260.

[82] Liye Lv, Xueguan Song, and Wei Sun. "Modify Leave-One-Out Cross Validation by Moving Validation Samples around Random Normal Distributions: Move-One-Away Cross Validation". In: *Applied Sciences* 10.7 (2020), p. 2448. ISSN: 2076-3417. DOI: 10.3390/app10072448.

[83] V. Aute, K. Saleh, O. Abdelaziz, S. Azarm, and R. Radermacher. "Cross-Validation Based Single Response Adaptive Design of Experiments for Kriging Metamodeling of Deterministic Computer Simulations". In: *Structural and Multidisciplinary Optimization* 48.3 (2013), pp. 581–605. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-013-0918-5.

[84] Aikaterini P. Kyprioti, Jize Zhang, and Alexandros A. Taflanidis. "Adaptive Design of Experiments for Global Kriging Metamodeling through Cross-Validation Information". In: *Structural and Multidisciplinary Optimization* 62.3 (2020), pp. 1135–1157. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-020-02543-1.

[85] Loic Le Gratiet and Claire Cannamela. "Cokriging-Based Sequential Design Strategies Using Fast Cross-Validation Techniques for Multi-Fidelity Computer Codes". In: *Technometrics* 57.3 (2015), pp. 418–427. ISSN: 0040-1706, 1537-2723. DOI: 10.1080/00401706.2014.928233.

[86] M. Trincavelli et al. "Towards Environmental Monitoring with Mobile Robots". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nice: IEEE, 2008, pp. 2210–2215. ISBN: 978-1-4244-2057-5 978-1-4244-2058-2. DOI: 10.1109/IROS.2008.4650755.

[87] T. Kubota, I. Nakatani, K. Watanabe, and S. Shimoda. "Study on Mole-Typed Deep Driller Robot for Subsurface Exploration". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. Barcelona, Spain: IEEE, 2005, pp. 1297–1302. ISBN: 978-0-7803-8914-4. DOI: 10.1109/ROBOT.2005.1570294.

[88] Daniel Gomez-Ibanez et al. "Autonomous Water Sampler for Oil Spill Response". In: *Journal of Marine Science and Engineering* 10.4 (2022), p. 526. ISSN: 2077-1312. DOI: 10.3390/jmse10040526.

[89] Kaisa Miettinen and Marko M. Mäkelä. "On Scalarizing Functions in Multiobjective Optimization". In: *OR Spectrum* 24.2 (2002), pp. 193–213. ISSN: 0171-6468, 1436-6304. DOI: 10.1007/s00291-001-0092-9.

[90] I. Giagkiozis and P.J. Fleming. "Methods for Multi-objective Optimization: An Analysis". In: *Information Sciences* 293 (2015), pp. 338–350. ISSN: 00200255. DOI: 10.1016/j.ins.2014.08.071.

[91] Mariette Awad and Rahul Khanna. "Multiobjective Optimization". In: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Ed. by Mariette Awad and Rahul Khanna. Berkeley, CA: Apress, 2015, pp. 185–208. ISBN: 978-1-4302-5990-9. DOI: 10.1007/978-1-4302-5990-9_10.

[92] Michael T. M. Emmerich and André H. Deutz. "A Tutorial on Multiobjective Optimization: Fundamentals and Evolutionary Methods". In: *Natural Computing* 17.3 (2018), pp. 585–609. ISSN: 1567-7818, 1572-9796. DOI: 10.1007/s11047-018-9685-y.

[93] Yicun Hua, Qiqi Liu, Kuangrong Hao, and Yaochu Jin. "A Survey of Evolutionary Algorithms for Multi-objective Optimization Problems With Irregular Pareto Fronts". In: *IEEE/CAA Journal of Automatica Sinica* 8.2 (2021), pp. 303–318. ISSN: 2329-9266, 2329-9274. DOI: 10.1109/JAS.2021.1003817.

[94] Marlon Alexander Braun. "Scalarized Preferences in Multi-objective Optimization". PhD thesis. Karlsruher Institut für Technologie (KIT) / Karlsruher Institut für Technologie (KIT), 2018. DOI: 10.5445/IR/1000082700.

[95] Kerstin Dächert, Jochen Gorski, and Kathrin Klamroth. "An Augmented Weighted Tchebycheff Method with Adaptively Chosen Parameters for Discrete Bicriteria Optimization Problems". In: *Computers & Operations Research* 39.12 (2012), pp. 2929–2943. ISSN: 03050548. DOI: 10.1016/j.cor.2012.02.021.

[96] Tinkle Chugh. "Scalarizing Functions in Bayesian Multiobjective Optimization". In: *2020 IEEE Congress on Evolutionary Computation (CEC)*. Glasgow, United Kingdom: IEEE, 2020, pp. 1–8. ISBN: 978-1-72816-929-3. DOI: 10.1109/CEC48606.2020.9185706.

[97] Gauthier Guinet, Valerio Perrone, and Cédric Archambeau. "Pareto-Efficient Acquisition Functions for Cost-Aware Bayesian Optimization". In: (2020). arXiv: 2011.11456 [cs, stat].

[98] Tom Blau, Edwin V. Bonilla, Iadine Chades, and Amir Dezfouli. *Optimizing Sequential Experimental Design with Deep Reinforcement Learning*. 2022. arXiv: 2202.00821 [cs, stat].

[99] Wanggang Shen, Jiayuan Dong, and Xun Huan. *Variational Sequential Optimal Experimental Design Using Reinforcement Learning*. June 2023. arXiv: 2306.10430 [cs, stat]. (Visited on 04/30/2024).

[100] Adam Foster, Desi R. Ivanova, Ilyas Malik, and Tom Rainforth. *Deep Adaptive Design: Amortizing Sequential Bayesian Experimental Design*. June 2021. arXiv: 2103.02438 [cs, stat]. (Visited on 04/30/2024).

[101] Richard S. Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. Second edition. Adaptive Computation and Machine Learning. Cambridge, MA London: The MIT Press, 2018. ISBN: 978-0-262-03924-6.

[102] Fatemeh Fathinezhad, Peyman Adibi, Bijan Shoushtarian, and Jocelyn Chanussot. "Graph Neural Networks and Reinforcement Learning: A Survey". In: *Artificial Intelligence*. Ed. by Jucheng Yang, Yarui Chen, Tingting Zhao, Yuan Wang, and Xuran Pan. Vol. 18. IntechOpen, Nov. 2023. ISBN: 978-1-80356-950-5 978-1-80356-951-2. DOI: 10.5772/intechopen.111651. (Visited on 05/02/2024).

[103] Paul Almasan, José Suárez-Varela, Krzysztof Rusek, Pere Barlet-Ros, and Albert Cabellos-Aparicio. "Deep Reinforcement Learning Meets Graph Neural Networks: Exploring a Routing Optimization Use Case". In: *Computer Communications* 196 (Dec. 2022), pp. 184–194. ISSN: 01403664. DOI: 10.1016/j.comcom.2022.09.029. arXiv: 1910.07421 [cs]. (Visited on 05/02/2024).

[104] Pranav Agarwal, Aamer Abdul Rahman, Pierre-Luc St-Charles, Simon J. D. Prince, and Samira Ebrahimi Kahou. *Transformers in Reinforcement Learning: A Survey*. July 2023. DOI: 10.48550/arXiv.2307.05979. arXiv: 2307.05979 [cs]. (Visited on 05/02/2024).

[105] Levent Bayındır. "A Review of Swarm Robotics Tasks". In: *Neurocomputing* 172 (2016), pp. 292–321. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2015.05.116.

[106] Andreas Kolling, Phillip Walker, Nilanjan Chakraborty, Katia Sycara, and Michael Lewis. "Human Interaction With Robot Swarms: A Survey". In: *IEEE Transactions on Human-Machine Systems* 46.1 (Feb. 2016), pp. 9–26. ISSN: 2168-2291, 2168-2305. DOI: 10.1109/THMS.2015.2480801. (Visited on 05/02/2024).

[107] Soon-Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. “A Survey on Aerial Swarm Robotics”. In: *IEEE Transactions on Robotics* 34.4 (Aug. 2018), pp. 837–855. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2018.2857475. (Visited on 05/02/2024).

[108] Chengke Xiong, Danfeng Chen, Di Lu, Zheng Zeng, and Lian Lian. “Path Planning of Multiple Autonomous Marine Vehicles for Adaptive Sampling Using Voronoi-based Ant Colony Optimization”. In: *Robotics and Autonomous Systems* 115 (May 2019), pp. 90–103. ISSN: 09218890. DOI: 10.1016/j.robot.2019.02.002. (Visited on 05/31/2022).

[109] L. Marques, A. Martins, and A.T. De Almeida. “Environmental Monitoring with Mobile Robots”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Edmonton, Alta., Canada: IEEE, 2005, pp. 3624–3629. ISBN: 978-0-7803-8912-0. DOI: 10.1109/IROS.2005.1545133. (Visited on 05/02/2024).

[110] Xinge Huang, Farshad Arvin, Craig West, Simon Watson, and Barry Lennox. “Exploration in Extreme Environments with Swarm Robotic System”. In: *2019 IEEE International Conference on Mechatronics (ICM)*. Ilmenau, Germany: IEEE, Mar. 2019, pp. 193–198. ISBN: 978-1-5386-6959-4. DOI: 10.1109/ICMECH.2019.8722887. (Visited on 05/02/2024).

[111] Genevieve Flaspohler, Nicholas Roy, and Yogesh Girdhar. “Near-Optimal Irrevocable Sample Selection for Periodic Data Streams with Applications to Marine Robotics”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 5691–5698. ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8460709. (Visited on 06/22/2022).

[112] Siva Kailas, Wenhao Luo, and Katia Sycara. “Multi-Robot Adaptive Sampling for Supervised Spatiotemporal Forecasting”. In: *Progress in Artificial Intelligence*. Ed. by Nuno Moniz, Zita Vale, José Cascalho, Catarina Silva, and Raquel Sebastião. Cham: Springer Nature Switzerland, 2023, pp. 349–361. ISBN: 978-3-031-49008-8.

[113] Xu Zhong, Allison Kealy, and Matt Duckham. “Stream Kriging: Incremental and Recursive Ordinary Kriging over Spatiotemporal Data Streams”. In: *Computers & Geosciences* 90 (May 2016), pp. 134–143. ISSN: 00983004. DOI: 10.1016/j.cageo.2016.03.004. (Visited on 12/18/2019).

[114] Danil Kuzin, Olga Isupova, and Lyudmila Mihaylova. “Spatio-Temporal Structured Sparse Regression With Hierarchical Gaussian Process Priors”. In: *IEEE Transactions on Signal Processing* 66.17 (Sept. 2018), pp. 4598–4611. ISSN: 1941-0476. DOI: 10.1109/TSP.2018.2858207.

[115] Shrutilipi Bhattacharjee, Soumya Kanti Ghosh, and Jia Chen. *Semantic Kriging for Spatio-temporal Prediction*. Vol. 839. Studies in Computational Intelligence. Singapore: Springer Singapore, 2019. ISBN: 9789811386633 9789811386640. DOI: 10.1007/978-981-13-8664-0. (Visited on 12/19/2019).

[116] Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. Dec. 2022. DOI: 10.48550/arXiv.1312.6114. eprint: 1312.6114 (cs, stat). (Visited on 05/02/2024).

[117] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. (Visited on 05/02/2024).

[118] Ashish Vaswani et al. *Attention Is All You Need*. Aug. 2023. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762 [cs]. (Visited on 05/02/2024).

[119] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014. (Visited on 05/02/2024).

[120] Prafulla Dhariwal and Alexander Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794. (Visited on 05/02/2024).

[121] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. Nov. 2014. DOI: 10.48550/arXiv.1411.1784. eprint: 1411.1784 (cs, stat). (Visited on 05/02/2024).

[122] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. Jan. 2020. arXiv: 2001.08361 [cs, stat]. (Visited on 04/18/2024).

[123] Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. "To Repeat or Not to Repeat: Insights from Scaling LLM under Token-Crisis". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 59304–59322.

[124] Aakanksha Chowdhery et al. *PaLM: Scaling Language Modeling with Pathways*. Oct. 2022. arXiv: 2204.02311 [cs]. (Visited on 04/17/2024).

[125] Laith Alzubaidi et al. "A Survey on Deep Learning Tools Dealing with Data Scarcity: Definitions, Challenges, Solutions, Tips, and Applications". In: *Journal of Big Data* 10.1 (Apr. 2023), p. 46. ISSN: 2196-1115. DOI: 10.1186/s40537-023-00727-2. (Visited on 04/17/2024).

[126] OpenAI. *Partnership with Axel Springer to Deepen Beneficial Use of AI in Journalism*. Available online. 2023. URL: https://openai.com/blog/axel-springer-partnership (visited on 12/13/2023).

[127] OpenAI. *OpenAI Data Partnerships*. Available online. 2023. URL: https://openai.com/blog/data-partnerships (visited on 11/09/2023).

[128] Muhammad Usman Hadi et al. *A Survey on Large Language Models: Applications, Challenges, Limitations, and Practical Usage*. July 2023. DOI: 10.36227/techrxiv.23589741.v1. (Visited on 04/18/2024).

[129] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. *Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model*. Nov. 2022. arXiv: 2211.02001 [cs]. (Visited on 04/18/2024).

[130] Ahmad Faiz et al. "LLMCarbon: Modeling the End-to-End Carbon Footprint of Large Language Models". In: *The Twelfth International Conference on Learning Representations*. 2024.

[131] Alexandra Sasha Luccioni and Alex Hernandez-Garcia. *Counting Carbon: A Survey of Factors Influencing the Emissions of Machine Learning*. Feb. 2023. arXiv: 2302.08476 [cs]. (Visited on 04/18/2024).

[132] Flor Ortiz et al. “Onboard Processing in Satellite Communications Using AI Accelerators”. In: *Aerospace* 10.2 (Jan. 2023), p. 101. ISSN: 2226-4310. DOI: 10.3390/aerospace10020101. (Visited on 04/19/2024).

[133] Sara Hooker. *The Hardware Lottery*. Sept. 2020. arXiv: 2009.06489 [cs]. (Visited on 04/25/2024).

[134] Groq Inc. *12 Hours Later: Groq is Running LLaMA-3 Instruct-8 70B by Meta AI on Its LPU Inference Engine*. Available online. 2024. URL: https://wow.groq.com/12-hours-later-groq-is-running-llama-3-instruct-8-70b-by-meta-ai-on-its-lpu-inference-enginge/ (visited on 04/19/2024).

[135] Volker Brühl. “Generative Artificial Intelligence – Foundations, Use Cases and Economic Potential”. In: *Intereconomics* 59.1 (Feb. 2024), pp. 5–9. ISSN: 1613-964X. DOI: 10.2478/ie-2024-0003. (Visited on 04/19/2024).

[136] S. Ambrogio et al. “An Analog-AI Chip for Energy-Efficient Speech Recognition and Transcription”. In: *Nature* 620.7975 (Aug. 2023), pp. 768–775. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06337-5. (Visited on 04/19/2024).

[137] Hunter Lightman et al. *Let’s Verify Step by Step*. May 2023. arXiv: 2305.20050 [cs]. (Visited on 04/17/2024).

[138] Jan Hatzius, Joseph Briggs, Devesh Kodnani, and Giovanni Pierdomenico. *Global Economics Analyst: The Potentially Large Effects of Artificial Intelligence on Economic Growth*. Available online. 2023. URL: https://static.poder360.com.br/2023/03/Global-Economics-Analyst_-The-Potentially-Large-Effects-of-Artificial-Intelligence-on-Economic-Growth-Briggs_Kodnani.pdf (visited on 03/26/2023).

[139] Keng-Boon Ooi et al. “The Potential of Generative Artificial Intelligence Across Disciplines: Perspectives and Future Directions”. In: *Journal of Computer Information Systems* (Oct. 2023), pp. 1–32. ISSN: 0887-4417, 2380-2057. DOI: 10.1080/08874417.2023.2261010. (Visited on 04/19/2024).

[140] Salman Ul Hassan Dar, Muzaffer Özbey, Ahmet Burak Çatlı, and Tolga Çukur. “A Transfer-Learning Approach for Accelerated MRI Using Deep Neural Networks”. In: *Magnetic Resonance in Medicine* 84.2 (2020), pp. 663–685. DOI: https://doi.org/10.1002/mrm.28148. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.28148. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.28148.

[141] Padmavathi Kora et al. “Transfer Learning Techniques for Medical Image Analysis: A Review”. In: *Biocybernetics and Biomedical Engineering* 42.1 (2022), pp. 79–107. ISSN: 0208-5216. DOI: 10.1016/j.bbe.2021.11.004.

[142] Mrinal R. Bachute and Javed M. Subhedar. “Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms”. In: *Machine Learning with Applications* 6 (Dec. 2021), p. 100164. ISSN: 26668270. DOI: 10.1016/j.mlwa.2021.100164. (Visited on 04/18/2024).

[143] Pierre-Emmanuel Novac, Ghouthi Boukli Hacene, Alain Pegatoquet, Benoît Miramond, and Vincent Gripon. “Quantization and Deployment of Deep Neural Networks on Microcontrollers”. In: *Sensors* 21.9 (Apr. 2021), p. 2984. ISSN: 1424-8220. DOI: 10.3390/s21092984. (Visited on 04/19/2024).

[144] Juraj Dudak, Michal Kebisek, Gabriel Gaspar, and Peter Fabo. “Implementation of Machine Learning Algorithm in Embedded Devices”. In: *2020 19th International Conference on Mechatronics - Mechatronika (ME)*. Prague, Czech Republic: IEEE, Dec. 2020, pp. 1–6. ISBN: 978-1-72815-602-6. DOI: 10.1109/ME49197.2020.9286705. (Visited on 04/19/2024).

[145] Alessio Gagliardi, V. Staderini, and S. Saponara. “An Embedded System for Acoustic Data Processing and AI-Based Real-Time Classification for Road Surface Analysis”. In: *IEEE Access* 10 (2022), pp. 63073–63084. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3183116. (Visited on 04/19/2024).

[146] Marcelo Brandalero et al. “AITIA: Embedded AI Techniques for Embedded Industrial Applications”. In: *2020 International Conference on Omni-layer Intelligent Systems (COINS)*. Barcelona, Spain: IEEE, Aug. 2020, pp. 1–7. ISBN: 978-1-72816-371-0. DOI: 10.1109/COINS49042.2020.9191672. (Visited on 04/19/2024).

[147] Swapnil Sayan Saha, Sandeep Singh Sandha, and Mani Srivastava. “Machine Learning for Microcontroller-Class Hardware: A Review”. In: *IEEE Sensors Journal* 22.22 (Nov. 2022), pp. 21362–21390. ISSN: 1530-437X, 1558-1748, 2379-9153. DOI: 10.1109/JSEN.2022.3210773. (Visited on 04/19/2024).

[148] Max Schwarzer et al. “Pretraining Representations for Data-Efficient Reinforcement Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 12686–12699.

[149] Gabriel Dulac-Arnold et al. “Challenges of Real-World Reinforcement Learning: Definitions, Benchmarks and Analysis”. In: *Machine Learning* 110.9 (Sept. 2021), pp. 2419–2468. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/s10994-021-05961-4. (Visited on 04/19/2024).

[150] Ofir Nachum, Shixiang (Shane) Gu, Honglak Lee, and Sergey Levine. “Data-Efficient Hierarchical Reinforcement Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[151] William Fedus et al. “Revisiting Fundamentals of Experience Replay”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 3061–3071. URL: https://proceedings.mlr.press/v119/fedus20a.html.

[152] Zhuangdi Zhu, Kaixiang Lin, Anil K. Jain, and Jiayu Zhou. “Transfer Learning in Deep Reinforcement Learning: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.11 (Nov. 2023), pp. 13344–13362. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2023.3292075. (Visited on 04/19/2024).

[153] Sanmit Narvekar et al. “Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey”. In: *Journal of Machine Learning Research* 21.181 (2020), pp. 1–50.

[154] Yujing Hu et al. “Learning to Utilize Shaping Rewards: A New Approach of Reward Shaping”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 15931–15941. (Visited on 04/19/2024).

[155] Marcin Andrychowicz et al. *Hindsight Experience Replay*. Feb. 2018. arXiv: 1707.01495 [cs]. (Visited on 04/19/2024).

[156] Rafael Figueiredo Prudencio, Marcos R. O. A. Maximo, and Esther Luna Colombini. “A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2024), pp. 1–0. ISSN: 2162-237X, 2162-2388. DOI: 10.1109/TNNLS.2023.3250269. (Visited on 04/19/2024).

[157] Masatoshi Uehara, Chengchun Shi, and Nathan Kallus. *A Review of Off-Policy Evaluation in Reinforcement Learning*. Dec. 2022. arXiv: 2212.06355 [cs, math, stat]. (Visited on 04/19/2024).

[158] Constantin-Valentin Pal and Florin Leon. “Brief Survey of Model-Based Reinforcement Learning Techniques”. In: *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*. Sinaia, Romania: IEEE, Oct. 2020, pp. 92–97. ISBN: 978-1-72819-809-5. DOI: 10.1109/ICSTCC50638.2020.9259716. (Visited on 04/19/2024).

[159] Athanasios Polydoros and Lazaros Nalpantidis. “Survey of Model-Based Reinforcement Learning: Applications on Robotics”. In: *Journal of Intelligent & Robotic Systems* 86 (May 2017), p. 153. DOI: 10.1007/s10846-017-0468-y.

[160] Aske Plaat, Walter Kosters, and Mike Preuss. *Deep Model-Based Reinforcement Learning for High-Dimensional Problems, a Survey*. Dec. 2020. arXiv: 2008.05598 [cs].

[161] J. Nathan Kutz, Steven L. Brunton, Krithika Manohar, Hod Lipson, and Na Li. “AI Institute in Dynamic Systems: Developing Machine Learning and AI Tools for Scientific Discovery, Engineering Design, and Data-Driven Control”. In: *AI Magazine* 45.1 (2024), pp. 48–53. ISSN: 2371-9621. DOI: 10.1002/aaai.12159.

[162] Nicholas Zolman, Urban Fasel, J. Nathan Kutz, and Steven L. Brunton. *SINDy-RL: Interpretable and Efficient Model-Based Reinforcement Learning*. Mar. 2024. arXiv: 2403.09110 [cs, eess, math].

[163] George Em Karniadakis et al. “Physics-Informed Machine Learning”. In: *Nature Reviews Physics* 3.6 (May 2021), pp. 422–440. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00314-5.

[164] Peter J. Baddoo, Benjamin Herrmann, Beverley J. McKeon, J. Nathan Kutz, and Steven L. Brunton. “Physics-Informed Dynamic Mode Decomposition”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 479.2271 (Mar. 2023), p. 20220576. ISSN: 1364-5021, 1471-2946. DOI: 10.1098/rspa.2022.0576.

[165] Chuizheng Meng, Sungyong Seo, Defu Cao, Sam Griesemer, and Yan Liu. *When Physics Meets Machine Learning: A Survey of Physics-Informed Machine Learning*. Mar. 2022. arXiv: 2203.16797 [cs, stat].

[166] Yanwen Xu, Sara Kohtz, Jessica Boakye, Paolo Gardoni, and Pingfeng Wang. “Physics-Informed Machine Learning for Reliability and Systems Safety Applications: State of the Art and Challenges”. In: *Reliability Engineering & System Safety* 230 (Feb. 2023), p. 108900. ISSN: 09518320. DOI: 10.1016/j.ress.2022.108900.