Title
Data-Driven Learning and Optimization of Dynamical Systems

Author
Makrygiorgos, Georgios

Publication Date
2023

Peer reviewed|Thesis/dissertation

Data-Driven Learning and Optimization of Dynamical Systems

By

Georgios Makrygiorgos

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Chemical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ali Mesbah, Co-chair
Professor Douglas Clark, Co-chair
Professor Karthik Shekhar
Professor Adam Arkin

Spring 2023

Data-Driven Learning and Optimization of Dynamical Systems

Abstract

Data-Driven Learning and Optimization of Dynamical Systems

by

Georgios Makrygiorgos

Doctor of Philosophy in Chemical Engineering

University of California, Berkeley

Professor Ali Mesbah, Co-chair

Professor Douglas Clark, Co-chair

Dynamical systems analysis and optimization is pivotal for safe, efficient, and resilient processes that consistently deliver high-quality products. Conventionally, decision-making and systems behavior analysis have relied heavily on physics-based models. However, these physics-based models pose several challenges to systems analysis and decision-making. These include prohibitively high computational costs associated with the numerical computations of the governing equations, limitations of our understanding of the system's underlying physical mechanisms that lead to insufficient and erroneous predictions, as well as the inherent nonlinearity and stochasticity that real-world systems exhibit. This thesis seeks to address these challenges by developing and applying data-driven methodologies for the dynamic analysis and optimization of complex systems, with emphasis on chemical and biochemical systems. The research presented in this dissertation can be distilled into three main contributions:

The first contribution revolves around the utilization of data-driven methods for approximating the dynamic behavior of dynamical systems and the applications that such data-driven models enable. In particular, the thesis focuses on data-driven strategies for learning the dynamics of systems under varying inputs, which can then be employed for uncertainty quantification analysis, optimal experiment design, and real-time decision making. This topic, though extensively investigated in the literature, remains challenging due to the limited, sparse, and noisy nature of available data. Our approach is rooted in the concept of flow-maps, which are operators that predict a system's future state based on its current state. We approximate this transition law using a polynomial chaos expansions-based Gaussian Process (GP), a probablistic non-parametric model that allows us to predict the expected behavior of the system while providing uncertainty bounds. Notably, our proposed approach demonstrates exceptional predictive capabilities even in low-data regimes and offers substantial computational savings compared to high-fidelity models for uncertainty quantification of dynamical systems.

The second contribution of this thesis is geared towards applying the Bayesian Optimization (BO) framework for the autotuning of general controllers. This challenge is critical in con-

trol theory due to the lack of easily derivable closed-form mathematical expressions for the system's closed-loop performance metrics. Limited and noisy closed-loop data can further compound this problem. BO is an ideal candidate method to address this problem, as it leverages the data-efficiency of GPs to create a probabilistic surrogate model to capture the relationship between decision variables and system performance. Through a careful balance of exploration and exploitation, BO strives to identify the globally optimal solution using informative queries from the closed-loop system. In particular, in this work we broaden the BO framework to address two significant aspects related to the autotuning of biochemical systems. Firstly, we tackle the challenge of BO under time-invariant uncertainty by proposing a new method for adversarially robust BO. This method concurrently learns the mapping from decision variables and uncertainties to performance. Secondly, we tackle the ubiquitous problem of multiple conflicting objectives that arises in real-world scenarios. We propose a multi-objective BO scheme in tandem with a data-driven model that encodes any existing information about the system's characteristics and is partially adaptable. The utility and performance improvement induced by such extensions are demonstrated through a bioreactor benchmark case study.

The third and final contribution of this thesis is a novel gradient-enhanced BO framework for closed-loop policy search. This advancement is pivotal in optimization problems where both zeroth- and first-order information are available (i.e., both the reward function and its gradient can be observed) during the query process. Traditional BO approaches may overlook the richness of the gradient information, potentially limiting their optimization efficiency. In contrast, our proposed approach, including two alternative methods, accelerates BO convergence by integrating both performance optimization and formal optimality conditions in the proposal of new query points. An important application of this method is in reinforcement learning, as policy-based methods under stochastic policies can provide objective function and gradient observations.

In conclusion, this thesis makes significant strides in the domain of data-driven analysis and optimization of dynamical systems, addressing crucial challenges that stem from the scarcity and low quality of data, a common trend in bioprocesses and integrated biomanufacturing systems. Future research can extend the application of these data-driven methods to emerging fields such as deep space bioprocessing. In such novel domains, data-driven models can fundamentally underpin the optimization of end-to-end system design, planning, and control. Furthermore, despite its demonstrated success and the extensions introduced in this work, BO presents significant opportunities for future research. These include managing high-dimensional design spaces and mixed-integer variables, incorporating black-box safety constraints, and leveraging advanced techniques when gradients are available.

To my mother, father, and brother, whose unwavering love, support, and encouragement have been the pillars of strength throughout this journey.

Contents

List of Figures

List of Tables

Acknowledgments

Over the past few years, I have undergone a remarkable journey, both personally and professionally. Leaving my home in Greece to pursue a PhD in the United States has proven to be a deeply rewarding experience in many ways. I have had the fortune of encountering incredible individuals, receiving outstanding mentorship, and participating in projects I once only dreamed of. Despite the challenges of moving 7,000 miles from home, I am grateful for the wealth of experiences and the sense of belonging I have found here. I wish to begin by expressing my appreciation for my advisor, Professor Ali Mesbah, who has provided outstanding guidance, patience, and support throughout my journey. Coming from a different scientific background, Ali offered me the chance to join his lab and provided me with the resources necessary to grow as a scientist and achieve success. I am also grateful for my co-advisor, Professor Douglas Clark, who allowed me to be part of the Center for Utilization of Biological Engineering in Space (CUBES) and contributed valuable advice over the years. Additionally, I extend my thanks to Prof. Joel Paulson at Ohio State University for his excellent guidance, incredible research insights and collaboration on numerous projects. Next, I would like to acknowledge those who played a key role in the creation and success of CUBES. Prof. Adam Arkin offered mentorship and insightful feedback on my work and would always show interest on my research projects. I also thank him for being part of my thesis and qualifying exam committees. Dr. Aaron Berliner, as the majordomo of CUBES, was not only an exceptional collaborator but also a very close friend, wonderful roommate, fantastic cook, and partner in crime during our vacations in Greece, always there to cheer me up when I was missing home. I appreciate him and his partner Emily Crawford for being great friends and housemates. I also extend my thanks to NASA for funding the CUBES project and supporting my research. I am grateful to Professor Karthik Shekhar for serving on my doctoral and qualifying exam committees and offering valuable advice on my thesis, as well as to Professor Kranthi Mandadapu and Professor Jay Keasling for their participation in my qualifying exam committee. Their feedback on my work really enlightened the path forward.

I also wish to extend my appreciation to all the members of the Mesbah lab, who have been companions throughout this journey. Dogan Gidon, Marc Martin Casas, Tor Aksel Heirung and Jared O'Leary, as more senior lab members, were instrumental in showing me the ropes when I joined in 2018. Their dedication to excellence and strong work ethic were truly inspiring. Angelo Bonzanini's exceptional work ethic and outstanding ideas and contributions made our collaboration seamless. I am grateful for Victor Miller's assistance and his admirable scientific curiosity and dedication. I would like to thank Kimberly Chan for her incredible scientific insights and our excellent collaboration. I would also like to thank Ketong Shao, whose attention to detail and thorough thinking have sparked numerous interesting research discussions. Moreover, I would like to thank Diogo Rodrigues for the great collaboration on early stages of my doctorate work and whose rigorousness was always a true inspiration for me. I would like to thank the newer members of the lab, Mira Khare, Shoubhanik Nath, Joshua Hang Sai Ip and Praneeth Srivanth for all the very interesting conversations and I wish them best of luck with their PhD, and, finally thank Giovanni Maria Maggioni for being an excellent collaborator with genuine love for science.

Furthermore, I want to recognize and extend my appreciation for the friends I made during my time in Berkeley. Elpiniki Apostolaki, my first friend here, provided invaluable support while we were housemates; I would like to thank her for kindness and thoughtfulness and for our moments around Bay Area. I would also like to thank Thanasis Geromichalos and his family for ensuring my smooth transition to the US, their support and for their delightful dinners. Antonis Alvertis, a great friend throughout my PhD, made my time in Berkeley more enjoyable with his kindness and camaraderie and together, along with his wife Kate Harris, we explored the beauties of California. I thank them both. I would also like to thank Valeria Sancisi for being a wonderful host and housemate during the first couple of years in the area.

My heartfelt thanks go to the friends and collaborators who supported me before and during my PhD journey. I am eternally grateful for my childhood friend Basilis Koutsogiannis, for always been there for me and always enthusiastically listening to me talking about research. He has been an exceptional friend all my life and along with his partner Georgia Lampropoulou have been nothing but supportive and caring throughout my PhD. I would to also thank Athina Podimata, who supported me in every step before coming to Berkeley. Nikolaos Kalafatis as my long-standing friend for all his support all these years, especially in times of hardships. Sofia Karatza, a very close friend since high-school, for her cheerfulness and love all these years. I am also extremely thankful to my close friends from my undergraduate time in Patras. Dr. Spiro Jorga for his amazing work ethic, caring nature and for being a dear friend who I always trust. Alexandros Mouchtaris for being a wonderful friend and for his "unique" sense of humor that made us bond. Giorgos Mitakis for being always a great friend and for all the encouragement. Dora Argyropoulou for her support and for always being a friend I can count on. Dr. Christos Stamatis for always being there to give me advice. I would also like to thank Dr. Dimitrios Fraggedakis, who has been part of my research journey from the beginning. I thank you all for being dear friends to this day. I am also extremely grateful to Professor Yannis Dimakopoulos and Professor John Tsamopoulos from University of Patras, for their mentorship and faith in me, giving me the opportunity to embark on my first research steps with them and for helping me achieve my dream of coming to the US for pursuing a doctorate degree.

I also want to express my immense gratitude to my partner Nicole Nappi, her son Dominick, and her mother Sylvia for their unconditional love and support over the past year, and for making me feel like a part of their family. Their appreciation and care have been extremely important to me. Nicole has been next to me in the most crucial part of my doctorate journey and her kind words and faith in me have always been uplifting and a motivation to push forward.

Finally, I would like to convey my profound appreciation to my family. First and foremost, my mother Georgia (Gogo), my father Spyros and my brother Marios for all their love and support since my childhood. Nothing would have been possible without them making sure that I am able to do what I love and helping me along the way. I thank them for believing in me and for giving me the opportunity to pursue my dreams. I am dedicating my thesis to them. I would also like to extend my sincere gratitude to my extended family members, whose boundless support and love have been a continuous source of strength throughout my academic journey

Chapter 1

Introduction

This chapter introduces the importance of data-driven methods for the dynamical analysis and optimization of complex chemical and biochemical systems. First, we discuss the significance of using dynamic models as a mathematical tool to describe the behavior of systems that exhibit highly nonlinear behavior, can be stochastic in nature, and for which our understanding of the underlying physics might be significantly limited. These characteristics necessitate the use of data-driven learning methods to create models that can predict the dynamics of the system of interest under varying conditions and often based on sparse and noisy data. Subsequently, dynamic models are used within closed-loop decision-making settings where data-driven optimization can greatly assist in discovering the optimal set of decision variables in a data-efficient manner.

1.1 Dynamical Chemical and Biochemical Systems

Analysis of dynamical systems is paramount in chemical and biochemical engineering in order to describe the behavior of complex chemical and biochemical processes over time [268, 51, 12]. Developing tools to understand their dynamics is essential for designing efficient and reliable industrial processes, developing new drugs and therapies, and improving our understanding of biological systems. Predicting the system's behavior under varying conditions and identifying key variables that control the system, allow for optimization of process performance, reduced costs, and ensured product quality and safety. The recent review by Mowbray et al. [176] provides an overview of numerous machine learning applications in the field of biochemical engineering, along with their potential integration into industrial settings; here we focus on methods that apply to dynamical systems.

Through interaction with physical systems, we can learn patterns and connections between inputs or excitations of systems and their responses or behaviors. Traditionally, scientific discovery relies on first-principles models that constitute the basis for describing the dynamics of systems. These are combined with empirical laws, which formalize interactions into equations, while data is used to quantify various properties and variables. Mathematical models based on conservation laws and constitutive equations are indispensable tools for analyzing the behavior of biochemical systems that exhibit complicated spatio-temporal dynamics. Dynamical systems are described by a system differential equations and are gen-

erally defined by the interplay of transport phenomena, thermodynamic laws, and kinetics. Nonetheless, computational approaches to solve such equations such as those based on finite differences [141] or finite elements [325] can be prohibitively expensive for certain type of analyses, as well as for real-time decision making. More specifically, the systematic and efficient quantification of the effects of model uncertainties on quantities of interest can be an arduous task using expensive-to-evaluate models. This is because uncertainty quantification (UQ) typically relies on sample-based methods, hence, simulations under varying parameters and/or initial/boundary conditions to obtain adequate samples for UQ can be prohibitive. Therefore, the concept of surrogate modeling emerged to alleviate this issue, e.g., as applied to expensive simulations of CO_2 sequestration [321] or fluid mechanics [180], where some quantity of interest (QoI) is approximated via a data-driven model, most notably Polynomial Chaos Expansions (PCE) [151]. On the other hand, adequate knowledge of the mechanisms that underlie the process of interest is not always available. Therefore, data-driven methods have emerged as alternatives for constructing models that serve specific purposes related to analysis and decision-making. The range of applications is vast. In Subsection 1.1, we provide concrete examples of the application of data-driven methods for model learning and optimization in the context of space biomanufacturing. In Subsection 1.2, we provide an overview of data-driven methods while discussing a few problems related to process control and optimal experimental design (OED).

Additionally, the limited understanding of the underlying mechanisms governing the behavior of chemical and biochemical systems, along with the type of performance metrics associated with various design problems, emphasizes the significance of data-driven optimization. For instance, designing metabolic pathways to maximize the production of valuable compounds is one such application [172]. Usually, discovering the optimal solution in this context involves probing the system under different design variables, which can be very costly and time-consuming if not done in a strategic manner (e.g., random search). Thus, in Subsection 1.3, we explore the topic of global optimization in "black-box" settings and provide an overview of the latest developments in the field.

Model Learning and Decision-Making for Space Biomanufacturing

A crucial application of data-driven modeling in biochemical systems is its potential to enable deep space exploration through biomanufacturing, supporting human-based missions. Established in 2017, the Center for Biological Engineering in Space (CUBES) explores how biotechnological advancements, which offer inherent mass, power, and volume advantages, can be harnessed for this purpose, as opposed to abiotic approaches. CUBES investigates the effective use of Martian in-situ resources (ISRU) for integrated biomanufacturing of essential products for Martian exploration, including biologically-produced pharmaceuticals, cell-based treatments/therapeutics, and materials for on-demand and modular additive manufacturing applications to maintain a crewed mission over an extended period [20]. Although numerous fundamental scientific breakthroughs have been achieved recently in the development of biotic life-support technologies for deep space missions, less focus has been placed on the role of rigorous system analysis and optimization for ensuring a safe and automated operation of such a biomanufactory from end to end, incorporating "astronaut-in-the-loop"

methodologies, where astronauts are involved as an essential part of the control and decision-making process [269].

While NASA [3] considers life-support systems (i.e., relying on plants and microbes) indispensable for such a task, operational aspects are often not systematically addressed. Consequently, Space Biological Engineering (SBE) has emerged as a novel interdisciplinary field, focusing on the development, implementation, and management of biologically-driven technologies to sustain life during long-term space missions. SBE merges synthetic biology as well as and bioprocess and systems engineering, taking into account the distinctive challenges and constraints of the space environment [19]. To this end, the traditional process engineering Design, Build, Test, Learn (DBTL) cycle [52] can be adapted for deep space biochemical engineering. Achieving the desired level of operational autonomy, necessitates the formal adaptation the foundations of process systems engineering (PSE) to space biomanufacturing. PSE is a broad multidisciplinary field, relying on mathematical modeling, optimization and various engineering principles for the design and control of chemical and physical processes [97], which has been greatly enhanced by data-driven models in recent years [138]. Here we briefly discuss the utility of data-driven modeling as part of the space biochemical engineering field. The benefits of fusing synthetic biology and biochemical engineering with systems engineering has already been realized; the review of Rollie et al. [237] discusses the concept of modularization and standardization in biological systems, stemming from the idea of unit operations in chemical engineering. The authors highlight the importance of multi-scale mathematical modeling and conclude with the remark that in order to incorporate synthetic biology into end-to-end designs, the systems need to be predictable, able to be tuned and provide robustness guarantees. Relating this to space biomanufacturing, this statement resonates with the fact that the operational landscape associated with deep space exploration can be highly uncertain. In [131], the authors discuss in particular the topic of biochemical system modeling from a utility point of view, i.e., how models can be utilized for model-based optimization and industrial integration [303], which is also paramount in the DBTL cycle in the context of SBE. Parallel to the recognition of the importance of modeling for SBE, recent advances in the filed of machine learning exemplify the use of such techniques for metabolic engineering [137]. On the same note, in [145] the use of artificial intelligence for the DBTL cycle analysis is discussed, focusing on its role towards aiding metabolic engineering design. Chapter 6 of this thesis provides more details on specific applications of data-driven models and several directions on how feedback and adaptive learning can be employed in both the prototyping phase and real-time operation of biological deep space exploration systems.

1.2 Data-Driven Model Learning and Uncertainty Quantification

In recent years, data-driven learning techniques have emerged as a powerful approach for modeling and understanding dynamic systems. These techniques leverage large or sometimes limited datasets to create approximate dynamic models, enabling efficient simulation,

prediction, and control of complex systems. The idea of using data-driven models for dynamics learning dates back several decades. A classical textbook in this field includes [249], which discusses the concept of system identification, i.e., the approximation of dynamics from observed input-output data, providing both the theory and practical considerations essential for such a task. Another fundamental textbook is [31], which extensively discusses autoregressive moving average (ARIMA) models and provides the iterative framework for parameter estimation and model validation.

Overview of Methods

Here we discuss several key methods that have been traditionally used for learning dynamical systems. Dynamic models are commonly represented by ordinary differential equations, describing the continuous time evolution of the states $s(t)$ of a system based on some set of equations $f(s,t)$, where t denotes the time, starting from some initial condition s_0. If $f(s,t)$ is known, the dynamic evolution can be computed either using analytical expressions (whenever the equations are simple enough), or, usually, based on time integration algorithms (e.g., Runge-Kutta [38]). Alternatively, the so-called flow-map function $\Phi_\delta(s(t),t)$ is an equivalent representation that relies on $f(s,t)$ and, if known, can be used to predict $s(t+\delta)$ from $s(t)$, where δ is some time step. Therefore, for the purpose of our discussion and to facilitate the understanding of the key concepts, we divide the methods for system learning into two broad classes: (1) methods that attempt to reconstruct the governing dynamic equations $f(s,t)$, and (2) methods that aim to learn the flow-map function $\Phi_\delta(s(t),t)$ directly, following the categorization in [50]. Note that the categorization of methods is not unique, as various perspectives and taxonomies can be applied to classify them.

In the first class, we can include methods for discovering analytical expressions for the governing equations of dynamical systems via symbolic regression [25] coupled with genetic programming and experimental data [247] to learn differential equations from a space of potential functional forms. A more recent work with foundations on symbolic regression and machine learning is that of [36] where the SINDy algorithm is introduced. SINDy is a data-driven method for discovering sparse and interpretable models of dynamical systems from observed data. The method relies on a library of candidate functions, solves an optimization problem to identify the most relevant terms, and reconstructs the governing equations in a form that is suitable for analysis and control, while is it is robust to measurement noise. Extensions of this work include [43] where SINDy is combined with a coordinate transformation to simultaneously learn both the governing equations and the coordinate system in which the dynamics are sparse. With the advent of deep learning, the task of learning the differential equations has been tackled via a class of continuous depth deep neural networks (DNN), known as neural ordinary differential equations (neural ODEs) [239, 222], applied to the discovery of stochastic systems [186] as well as parameter estimation of ODEs [33].

In the second class, we encounter methods that aim to directly approximate the flow-map $\Phi_\delta(\cdot)$ of a dynamical system. They focus on learning a mapping from the current state of the system to its state at a future time, bypassing the need to explicitly reconstruct the underlying governing equations. By leveraging the flow-map representation, these methods can efficiently predict the evolution of the system's states over time, even in cases where the governing equations are not easily identifiable or the system exhibits complex behavior. A

popular technique in this area relies on the, so-called Koopman operator [35, 189] which is a linear operator that acts on the space of functions defined on the state space of a dynamical system and can be used iteratively to obtain a trajectory of the system. In practice, data-driven approximations are used to find the Koopman operator such as dynamic mode decomposition [135, 306], which approximates the eigenvalues and eigenvectors of the Koopman operator. It is also worth noting that this form of dynamical systems approximation is very similar to the concept of state-space modeling which is widely used in model-based control applications [29, 229, 219, 134, 254].

Selection of Model Structures

In this thesis, we focus on the second class of model learning methods for dynamical systems, i.e., we attempt to predict the future states given some past trajectory data. A common model structure employed towards this task relies on parametric models, i.e., statistical models that assume a specific functional form for the underlying relationship between variables, with a finite number of parameters to be estimated from the data [22, 101]. Deep learning techniques have emerged as a powerful representation of highly complicated mappings, with application to dynamic systems [82]. Most notably, physics-informed neural networks [223] and dynamics reconstruction via neural networks under noisy data [240] have shown promise for data-driven modeling of nonlinear dynamical systems. Recently, Qin et al. [220, 221] proposed a deep learning-based approach for data-driven approximation of the integration operator of differential equations from observations of state variables.

On the other hand, non-parametric models [302] offer a flexible alternative to parametric models for data-driven modeling of dynamic systems. Unlike parametric models, non-parametric models do not impose a predefined model structure and, hence, do not rely on a fixed number of parameters. This allows them to adapt more freely to the underlying data, often resulting in improved performance in capturing complex and unknown relationships. Popular models in this category include kernel-based methods such as support vector machines [277] and decision trees or random forests [53]. Gaussian Processes (GPs) is another versatile class of non-parametric, kernel-based models, extensively employed in machine learning and statistics for tasks such as regression and classification [228, 304], relying on Bayes' theorem. GPs are inherently connected to the normal distribution; while the normal distribution is a probability distribution over a finite-dimensional random vector, GPs extend this concept by defining a distribution over infinite-dimensional function spaces. In essence, a GP assigns a multivariate normal distribution to the function values at a finite set of input points. The key feature that links GPs to the normal distribution is the assumption that a finite collection of function values exhibits a joint normal distribution. This relationship allows GPs to leverage the properties of normal distributions, such as closed-form expressions for marginal and conditional distributions, making them a powerful tool for probabilistic modeling and inference in machine learning and statistics.

Here, we discuss the fundamental concepts behind GPs and more mathematical details will be given in the subsequent chapters. A GP is completely characterized by two components: a prior mean function and a kernel function (also known as the covariance function). The prior mean function, denoted as $m(x)$, represents the expected value of the GP at any given input point x. It encodes our prior belief about the function we aim to model before

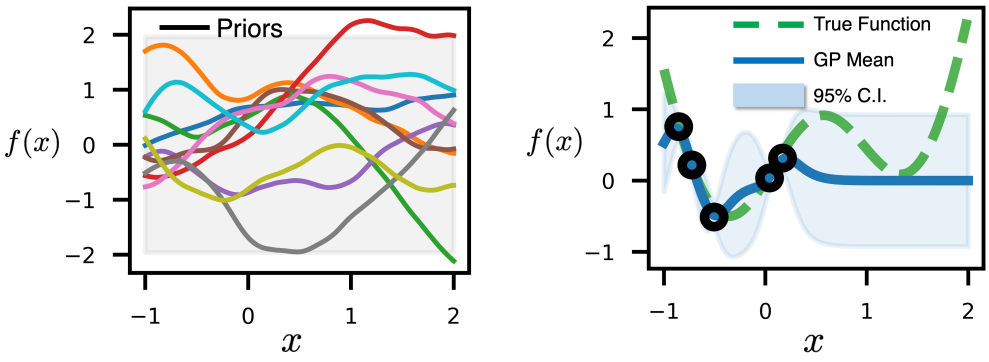

Figure 1.1: Example of a GP model for an unknown function $f(x)$. The true function is given by the green dashed line. The available (noisy) function observations are represented by the black circles. By using the observations, a GP model is fitted. The posterior predicted mean at each x is given by the solid blue line, while the light-blue shaded area represents the 95% confidence interval based on the posterior predicted variance, asumming that for each x, the prediction follows a Gaussian distribution.

observing any data. In many cases, the prior mean function is set to zero, however, if we have some knowledge about the function we can directly incorporate it. The most important element in a GP model is the kernel function $k(x, x')$, a positive definite function that quantifies the similarity (or covariance) between function values at two input points x and x'. It plays a crucial role in defining the shape and smoothness of the underlying functional behaviors that the GP can represent. The choice of the kernel function determines the expressiveness of the GP and has a significant impact on the model's ability to capture the underlying structure of the data. Common kernel functions include the squared exponential (also known as radial basis function) and Matérn [116].

An example of how GPs work is illustrated in Figure 1.1, where the goal is to learn a function $f(x)$, based on a few noisy observations. In the left plot, we see several prior functions drawn from a Gaussian Process with a zero mean function and a chosen kernel function. The shown functions represent our initial belief about the possible shapes and behaviors of the true function $f(x)$ before incorporating any observations. The diversity of prior functions indicates the flexibility and expressiveness of the GPs, which allows it to capture a wide range of functional forms. The right plot shows the updated GP after incorporating the noisy observations. The solid line represents the posterior mean function, which is our best estimate of the true function $f(x)$ given the available data. The shaded region around the mean function corresponds to the 95% confidence interval, reflecting the uncertainty in our predictions. As can be seen, the GP predictions are more accurate and exhibit smaller variance in regions where observations are available. This is because the GP leverages the information from the observed data points to adjust its beliefs about the underlying function. In regions with no observations (or away from them), the GP predictions revert to the prior mean function (in this case, zero) and exhibit larger uncertainty, as indicated by the wider confidence intervals. This demonstrates the ability of GPs to adaptively provide more

accurate predictions and better uncertainty quantification in regions where data is available while still offering reasonable predictions in unexplored regions.

UQ and Optimal Experimental Design of Dynamical Systems

UQ is a critical component in the analysis, control, and design of dynamical systems, as it facilitates the systematic assessment of the influence of uncertain parameters and disturbances on system performance. Traditional deterministic models may fail to accurately predict the behavior of the system in the presence of uncertainties, resulting in suboptimal optimization strategies and experimental designs. Optimal experimental design (OED) based on physics models is a significant field in which data-driven models are employed provide robustness to the analysis. OED [30, 71] is a systematic approach to designing experiments with the objective of obtaining the most information from the least number of experiments by selecting the experimental conditions that maximize the information gain. In the context of dynamical systems, OED objectives typically involve model selection, such as in the case of biochemical networks [294], and optimally estimating model parameters [164] Standard OED approach rely on deterministic models which potentially lead to ineffective designs due to their inability to account for the inherent uncertainties present in complex systems. To this end, robust OED attempts to partially alleviate this issue by computing the excitation inputs based on perturbation ranges of parameters around some nominal values, e.g., formulated as max-min optimization problems [218, 130], or using a probabilistic treatment that accounts for the statistical properties of uncertainties [170] based on PCEs. Nevertheless, robust OED is not suitable when the prior distribution of parameters is very broad or deviates from normality. These limitations can be addressed by the general and powerful framework of Bayesian OED that explicitly incorporates uncertainty into the utility metrics associated with information gain from experiments, under noisy, incomplete, or indirect data enabling more robust designs [42, 236]. Computational challenges arising from Bayesian OED have been addressed using surrogate modeling, such as PCEs [110, 199], and safety considerations during OED have been explored using GP modeling [207].

1.3 Data-Driven Optimization

Overview of Methods

The task of numerical optimization is of paramount importance in a plethora of engineering applications. Here, we primarily focus on the task of optimizing a "black-box" function $f(x)$, i.e., cannot be given in a closed mathematical form and the effect of the decision variables on the system performance is only attainable via noisy and expensive queries. Optimization methods are typically divided into derivative-based or derivative-free, based on whether the gradient of the function is used by the algorithm to discover the optimal solution. A common situation encountered in the optimization of real-world, industrial systems is that where models are not available and direct interaction with the true system is required. In this thesis we focus in the latter case, i.e., derivative-free methods. The paper of Rios et al. [234] provides a comprehensive review of derivative-free optimization,

including a historical overview of algorithmic developments, a categorization of state-of-the art techniques as well as a set of comparative benchmarks. Perhaps the most simple to understand and implement methods rely on pure search strategies over a finite number of points in the domain, based on some "passive" experimental design, e.g., following a random selection of points in the domain or relying on a grid. Hence, brute force or random search methods, albeit extremely data intensive, have been of interest, especially when combined with heuristics or empirical rules that help restrict the search space and direct the search process [142, 318]. Early work on derivative-free optimization led to development of simplex-based methods [57]; for instance, the Nelder-Mead method [182] works by maintaining a simplex (a polytope with $d+1$ vertices in the d-dimensional space) and updating it through operations such as reflection, expansion, contraction, and shrinkage. These operations are designed to explore the search space and converge to a local minimum or maximum without the need for gradient information. Nelder-Mead belongs to the family of local methods, while in the same category we encounter surrogate-based methods as well, e.g., trust region [216] and implicit filtering [90]. In this thesis, we mainly focus on the family of global search methods. Similarly to local methods, surrogate models (which are also known as response surfaces) are also used in that case to guide the search over the true function. Sobester et al. [259] provides a comprehensive review of surrogate modeling techniques and their applications to optimization. Various methods in this category utilize surrogate models along with utility criteria to enable a guided search; efficient global optimization (EGO) uses Kriging surrogate models for deterministic global optimization [115] while radial basis functions have also been used in the context of stochastic global search [231]. Stochastic global algorithms have been recognized to provide efficient solutions to the challenges posed by black-box functions. Genetic algorithms employ a population of candidate solutions, which evolve through operations such as selection, crossover, and mutation. By mimicking the process of evolution, GAs are capable of exploring the search space more effectively than random search or simplex-based methods [109]. Another population-based algorithm is particle swarm optimization (PSO), inspired by the collective behavior of birds flocking and fish schooling [121]. Each particle represents a potential solution and moves through the search space by adjusting its position based on its own best position and the best position of the swarm, balancing exploration and exploitation across the domain. Finally, another notable algorithm in this category is simulated annealing [126], which is particularly useful when the function contains multiple local optima.

Bayesian Optimization

In the realm of surrogate model-based and derivative-free stochastic global optimization techniques, Bayesian Optimization (BO) has recently gained a great amount of attention due to its success in discovering (nearly) optimal solutions in tasks where the performance function associated with the system is "black-box" in nature [80, 251]. In essence, BO attempts to convert the original optimization problem into a sequence of easier-to-solve sub-problems that progressively aim to update our knowledge about the relationship between the decision variables and the performance function, via observing the performance at selected points in the domain based on some utility metric. This efficient search via intelligent selection of query points provides the most valuable information about the objective function while

minimizing the number of costly evaluations needed to identify the optimal solution. BO employs three fundamental elements to achieve this. The first element is a statistical model that learns the relationship between the decision variables and the performance of the system. BO employs probabilistic machine learning to develop a statistical representation of the unknown objective function $f(x)$, which offers not only a point estimate prediction at some x that has not been observed yet, but also quantifies the uncertainty or confidence in the prediction. The most popular choice of such a probabilistic model relies on GPs, however, alternatives such as Bayesian neural networks [265] and random forests [112] have been explored. The second element is the acquisition function. This strategically designed function is utilized to determine the most advantageous point(s) to evaluate within the function's domain. The acquisition function is responsible for striking a balance between exploration, i.e., suggesting points in the domain where the predicted performance is highly uncertain, and exploitation, i.e., directing the search in a narrow region where it is more likely to obtain the optimal solution. There are multiple acquisition functions and designing them is an active area of research in literature. Popular acquisitions for BO are the probability of improvement and expected improvement [80], multiarmed bandit acquistions such as lower/upper confidence bound [258], entropy search [104] or knowledge gradient [310]. The third element in BO is the optimal point selection. The third element is a criterion for selecting the optimal point. In practice, we select as the optimal point x^* the one for which the observed performance is the highest among the already performed queries, however, a criterion that adjusts for observation noise can be employed.

An Illustration

To illustrate the main idea of BO, assume that we are interested in the minimization of $f(x)$, where $x \in \mathcal{X} \subset \mathbb{R}^d$ are our decision variables in the d-dimensional space, and the function at some x is only observed via noisy measurements $y(x) = f(x) + w$, where $w \in \mathcal{W}$ is some noise realization. The closed-form of $f(x)$ is not known and the only way to learn the objective is through noisy observations. Figure 1.2 illustrates two different iterations of BO. The algorithm begins with just two noisy observations of the true function, which are then utilized to fit a GP model. In this initial stage, the posterior mean resembles a simple linear function (the simplest function that connects two points), and we can observe the confidence or predictive uncertainty intervals throughout the domain. This is determined by the acquisition function (given as the solid orange line), which expresses the utility of querying any particular point in the domain. Here we rely on the lower confidence bound (LCB) function, suited for a minimization problem

$$\alpha(x) = \mu(x) - \beta\sigma(x), \tag{1.1}$$

where $\mu(\cdot)$ is the posterior mean, $\sigma(\cdot)$ is the square root of the posterior variance and β is a hyperparameter that reflects the exploration-exploitation tradeoff. In the minimization setting, the points with the most utility are the ones that minimize the LCB. We can see that suggested point to query at this stage is the lower bound of the domain; this is because, due to the uncertainty, exploring the space provides significant utility, which is reflected in the LCB function through the predictive variance term, weighted by the exploration hyperparameter, i.e., the $\beta\sigma(x)$ dominates. As more points are queried, the posterior GP is

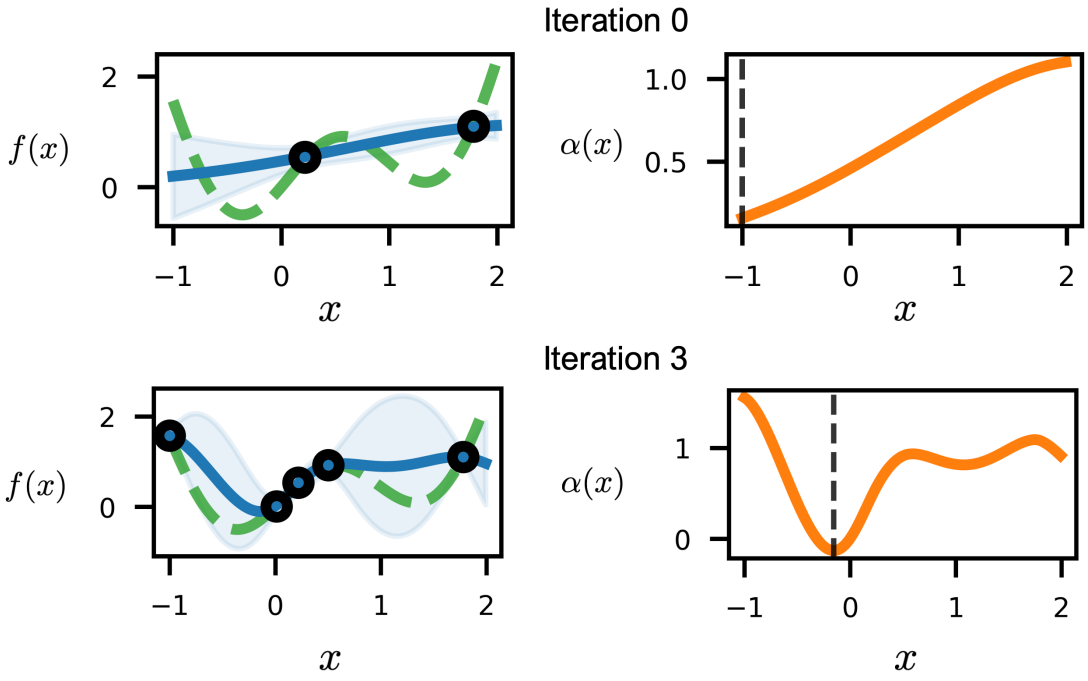

Figure 1.2: An illustration of Bayesian Optimization for a simple one-dimensional function. On the left, the Gaussian Process posterior mean (solid blue line) and variance (light blue confidence intervals) are plotted along with the true function (green dotted line). The right plot shows the acquisition function at the current iteration and the suggested point to be queried (black vertical dotted line). Top row shows the initial BO step when only two observations (denoted with black circles) are available. After a few points have been observed, based on the suggested queries, the shape of the surrogate GP function changes as seen in the bottom row and the acquisition function proposes points closer to the true optimum.

updated, and the acquisition function can locate a narrow region to be exploited, since $\sigma(x)$ becomes progressively smaller and the expected value of the function $\mu(x)$ dominates in the AF. At each iteration, we have the option to claim as the solution the point x that leads to the best observation so far. The algorithm terminates based on the available experimental/computational budget or when some convergence criterion is met.

Advances in BO

As the field of BO continues to grow, new advanced techniques are being developed to tackle increasingly complex optimization problems, going beyond the traditional BO framework; new ideas such as multifidelity modeling, multiple objectives and gradient-enhanced optimization are introduced. By incorporating advanced techniques, BO can better handle optimization problems with high-dimensional search spaces, expensive function evaluations, and noisy or incomplete data. Here we briefly discuss several of these advancements.

Multi-Fidelity BO

As stated before, BO is an attractive technique for data-efficient optimization, yet, the number of suggested points might still be prohibitively expensive to query in order to converge to the globally optimal solution. Although standard BO builds a probabilistic surrogate model for the objective based the expensive observations of the true system, the core notion of multi-fidelity BO (MFBO) [118] is to construct more objective surrogates using observations from lower-fidelity, but also less expensive sources [261], following the general framework of multi-fidelity optimization (MFO) [77]. For instance, in the context of biochemical processes, physical experiments, high-fidelity physics-based models and lumped-parameter models can be thought of as levels of fidelity. The main efforts of algorithmic development in MFBO revolve around discovering efficient ways to correlate the various fidelities (e.g., via multi-output GPs [146]), which are mutually dependent; subsequently, the acquisition function attempts to query the various fidelities at multiple locations in the decision variable space. Typically, the lower-fidelity models are prioritized in order to explore the space and suggest regions of possible optimality of the higher-fidelity models or experiments. This complex exploration-exploitation trade-off aims to optimally allocate the querying budget across fidelities.

Multi-Objective BO

BO has been widely utilized in cases where a single objective drives the querying procedure, however, in several applications we are interested in multiple objectives that are conflicting in nature. This is the main notion of multi-objective optimization (MOO) [62, 99]. Data-driven optimization methods, such as those based on genetic algorithms [244], have been exploited before to solve MOO problems [129, 163]. BO offers a viable, data-efficient alternative to MOO problems; hence multi-objective Bayesian Optimization (MOBO) has gained attention recently. In MOO there is typically no single optimizer, thus, the goal is to discover a set of *Pareto* optimal points that define the so-called Pareto front; each Pareto optimal solution ensures that simultaneous improvement in all objectives is not attainable. Similarly to single-objective BO, MOBO typically constructs a probabilistic surrogate for each objective individually. The standard approach to MOBO has been to build upon the standard acquisition functions based on the notion of the hypervolume (HV), which, in words, is a measure of space covered by the Pareto front and some reference point. To this end, an expected hypervolume improvement (EHVI) acquisition has been proposed [69, 58] (as an extension to EI), while using an MO variant of UCB has also been used [215]. An alternative approach relies on *information-theoretic* criteria where the Pareto-front entropy is maximized [278]. Overall, the querying process seeks to suggest points that belong to the Pareto optimal set. In **Chapter 3** MOBO will be discussed in the context of model learning for model-based control applications.

Gradient-Enhanced BO

The importance of incorporating gradient observations, $\nabla_x f(x)$, into BO has been recently recognized, e.g., [253]. In practice, gradient observations can be established via automatic differentiation [15, 213], finite differences or thanks to theorems such as the policy gradient

theorem [276] and its variants that approximate the gradients of reward/cost signals in the context of reinforcement learning. Based on the fact that the gradient of a GP is also a GP [227], a gradient-conditioned joint kernel matrix function can be defined such that the resulting GP model yields multiple outputs, i.e., predicts the function and its gradients at given test points while taking into account all the correlations between those functions and corresponding available observations [309]. This is particularly useful in higher dimensional spaces or when observations are available in limited locations in the domain, since for each training point we have $d + 1$ pieces of information, leading to a better approximation of the function compared to uniquely using just function observations. Thus, gradient information can enable a faster discovery of the optimum solution. In **Chapter 5** this topic will be discussed extensively in the context of policy-based reinforcement learning (RL).

Applications of BO

The problems that BO is suitable to address are ubiquitous in real-world applications. One such domain where BO has made a significant impact is materials science, enabling the rapid discovery of materials with desired characteristics. BO has been utilized in several applications such as (i) automation of experiment design along with high-throughput experiments in [93], showing major improvement against a grid-search approach, (ii) optimal design of graphene thermoelectrics [315], (iii) optimal closed-loop growth of nanotubes [45], (iv) effective screening and discovery of metal-organic frameworks [279], and (iv) towards creating an agent for establishing "self-driving" laboratories for thin film applications [152]. Several benchmarks are analyzed in [144]. BO has also been utilized in the discovery of pharmaceutical products [243, 143], as the experiments for drug design induce large costs and very large time-frames, while BO can be utilized to actively guide the process even in noisy settings as shown in [16]. Moreover, BO spans to other domains such as finance [94], biomedical engineering [185] and aerospace design [136].

Another crucial application of interest for BO, which is also a major part of this thesis, is the utilization of BO as a data-efficient approach for discovering optimal control [150]. The idea of tuning pre-existing policies on the real system to facilitate the policy learning procedure has been utilized in the context of biochemical engineering [208], still, a large number of interactions with the system might be needed for discovering an optimal policy, which is impractical. To alleviate this problem, BO for policy search and RL has recently received a lot of attention in the literature. For example, in [160] prior information is incorporated into BO to warm-start the search process of discovering the optimal policy on the real system. Other works that highlight the utility of BO for policy search include [81] and [178]. Moreover, RL with multiple performance objectives has been studied (e.g., [184, 119]), while MOBO has been employed in this case as a method for optimal policy discovery such as in [291] where the closed-loop system performance and robustness are viewed as conflicting objectives.

1.4 Thesis Outline and Organization

The contribution of this PhD research work and dissertation is two-fold. The first contribution (Aim 1) is the development of data-driven approaches for learning dynamic systems with uncertain inputs and utilizing them for UQ analysis and OED. The second contribution (Aim 2) is the development of methods for the closed-loop optimization of dynamic systems in a data-efficient manner using BO. All the works that contributed to this PhD research are listed in Table 1.1, while the works that are presented in this dissertation are in bold.

Table 1.1: Summary of contributions of this Ph.D. research.

Work	Aim	Contributions
[154]	1	Review of UQ methods for system analysis of 2D crystallization
[157]	1	Data-driven discovery of dynamic systems and UQ of biochemical systems
[235]	1	Tractable polynomial optimization for global solution of Bayesian OED
[236]	1	Bayesian OED under arbitrary prior and noise distributions
[158]	1	UQ of a Martian life support system
[27]	2	Fast model predictive control using neural networks and GPs
[262]	2	Constrained BO for auto-tuning of model-predictive controllers under noisy measurements
[197]	2	Adversarially robust BO for controller tuning under uncertainty
[159]	2	Multi-Objective BO for performance oriented model learning for control
[44]	2	Multi-Objective BO for policy adaptation
[155]	2	Gradient-enhanced BO using acquisition ensembles and multi-objective optimization
[156]	2	Gradient-enhanced BO via enforcing optimality constraints

The structure of the thesis is as follows. **Chapter 2** presents an approach for learning the dynamics of biological and biochemical systems via data-driven flow-maps. This approach accounts for the effects of uncertain parameters or control inputs, and variable stepping times. We employ a GP-based model enhanced by PCEs to achieve high accuracy in predicting dynamic trajectories, enabling a variety of analyses, including UQ and Bayesian parameter inference.

Chapter 3 shifts focus to the closed-loop optimization of a dynamic system. We introduce a MOBO approach for learning a dynamic model used for control purposes. Addition-

ally, we propose a novel composite neural network-based structure that leverages any prior model available for the system to enhance its performance for closed-loop control.

Chapter 4 addresses the autotuning of general control structures when relying on high-fidelity simulators with time-invariant uncertainties, such as parametric uncertainty. We present an Adversarially Robust BO (ARBO) approach where a GP model concurrently learns the mapping from decision variables and uncertainties to performance measures. We employ an alternating optimistic/pessimistic sub-problem optimization scheme to solve the robust closed-loop performance optimization problem.

Chapter 5 presents an approach for BO that directly incorporates observations of performance function gradients into the sequential decision-making process. We demonstrate two alternative methods that accelerate BO convergence by considering both zeroth-order (i.e., performance observations) and first-order (i.e., formal optimality conditions) criteria when proposing new query points. One potential application for this approach is RL, as policy-based methods under stochastic policies can provide gradient observations thanks to the policy gradient theorem and its variants.

Finally, **Chapter 6** concludes the thesis and explores potential avenues for future work, particularly regarding the application of data-driven methods in methods for biomanufacturing in deep space missions.

Chapter 2

Data-Driven Flow-Map Models for Dynamics Learning and Fast Uncertainty Quantification

This chapter discusses the topic of data-driven estimation of the flow-map operator to learn the system dynamics directly from observed data in cases where the governing equations are not known. Additionally, expensive models can be approximated using flow-map compositions for uncertainty quantification applications. Data-driven flow-map models learn integration operators of governing differential equations in a black-box manner, making them useful for deriving fast-to-evaluate surrogates for expensive models or reconstructing long-term system dynamics via experimental observations. We present a data-efficient approach to data-driven flow-map modeling using polynomial chaos-Kriging and demonstrate it for various systems, including a co-culture bioreactor and microbial electrosynthesis reactor. Such models are important for designing and optimizing bioprocesses and integrated biomanufacturing systems. This chapter is adapted from [157].

2.1 Introduction

Computational models have become indispensable tools for understanding the complex behavior of biological and biochemical systems towards design and optimization of bioprocesses and integrated biomanufacturing systems [12]. Recently, there has been a growing interest in data-driven methods for modeling the uncertain and nonlinear dynamics of biochemical systems, as these models constitute the cornerstone of various model-based analyses and decision-making tasks such as experiment design, hypothesis testing and parameter inference [79, 92, 113]. Data-driven modeling is especially useful when it is formidable to derive first-principles descriptions for systems whose complex behavior can span over multiple length- and time-scales. Data-driven models have shown promise for inferring the dynamics of cellular systems and metabolic networks (e.g., [248, 56]). Hybrid models (aka gray-box models) that combine physics-based models with data-driven descriptions of unknown or hard-to-model phenomena have also proven useful for describing the complex behavior of biochemical systems [61, 297, 250, 319]. In this work, we focus on *data-driven discovery*

of dynamical systems, whereby the goal is to learn directly the governing equations from system observations. A class of data-driven discovery methods for unknown systems relies on basic assumptions about the structure of the underlying equations [25]. To this end, a popular technique is based on sparse identification from dictionaries of possible governing terms, such as the SINDy algorithm presented in the introduction, which has been shown to be particularly useful when limited system observations are available. On the other hand, non-parametric modeling approaches relax the necessity of using a library of candidate terms [102]. Following the approach of directly learning the transition rules, or flow-maps, for the system of interest, the usefulness of this approach for discovery of dynamics of biological systems has been demonstrated on several benchmark problems in [272], mainly since it removes the necessity of assumptions about the dynamic model structure.

We are also interested on how these methods can be used for model-based uncertainty quantification (UQ) in biochemical and biological systems when relying on expensive-to-evaluate computational models. Predictions of the behavior of biochemical systems are generally subject to various sources of uncertainty due to unknown model structure, parameters, and/or initial and boundary conditions. This has spurred development of a plethora of set-based [271] and probabilistic [179, 256] methods for forward and inverse UQ problems (e.g., [128, 241, 293, 170, 198]). However, the most commonly used UQ methods rely on Monte Carlo sampling [39], which can be intractable for expensive computational models of biochemical systems, especially when models consist of a large number of differential equations and/or have a large number of uncertain inputs.

As stated in Chapter 1, surrogate modeling is being increasingly used to facilitate complex UQ analyses that would otherwise be computationally prohibitive. The key notion in surrogate modeling is to construct a data-driven mapping between inputs to a system and the quantities of interest in a non-intrusive manner, in which the "data generating process," e.g., a high-fidelity model, is treated as a black-box to generate as few training samples as possible [273]. Such a data-driven representation can be used as a computationally efficient surrogate for expensive computational models in order to predict the output quantities as a function of inputs. A variety of surrogate modeling techniques such a generalized and sparse polynomial chaos [312, 23], Kriging [54] and deep learning [288] have been successfully applied to various biological and biochemical systems (e.g., [198, 270, 233, 246, 206]). Nonetheless, a critical challenge in the majority of these techniques arises from capturing the time-evolution of the states in an efficient manner. The most common approach, known as *time-frozen* surrogate modeling [209, 154], for predicting the time-evolution of states, relies on constructing separate surrogate models for all time points at which the states must be predicted. As such, the "time-frozen" approach can be an inflexible and inefficient way of surrogate modeling for dynamical systems, especially in dynamic UQ and decision-making problems that hinge on making predictions over an adaptive sequence of time instants.

In this chapter, we leverage the notion of flow-map (de)composition, as also investigated in [220, 221], for data-efficient discovery of system dynamics from experimental observations or high-fidelity simulation data. Conceptually, a flow-map is an analytical operator that maps the current state and input of a system to a future state based on exact integration of model equations over some specified time step. Numerical integration schemes for ordinary differential equations in fact seek to numerically approximate flow-maps to compute the time-evolution of state variables as a function of input variables. Here, we propose to

approximate flow-maps in a data-driven manner via non-intrusive surrogate modeling, such that the resulting *data-driven flow-map* is a surrogate for differential operators of the differential equations governing a dynamical system. Hence, data-driven flow-map models are able to discover system dynamics irrespective of the unknown structure of model equations. In addition, data-driven flow-map models can address the above-described challenge of "time-frozen" approaches to surrogate modeling via circumventing the need for construction of separate surrogate models at different time instants. This can be especially useful for fast UQ and optimization-based analyses of dynamical systems that hinge on repeated runs of expensive computational models over a sequence of time instants.

We demonstrate the usefulness of data-driven flow-maps for discovery of system dynamics from data, as well as for fast UQ applications based on expensive computational models. Here, sparse polynomial chaos Kriging (PCK) [204] is used for data-driven approximation of flow-maps owing to its data efficiency, ability to approximate complex mappings and ability to quantify the uncertainty of model predictions. The versatility of data-driven flow-maps is first demonstrated via the discovery of the transient behavior of benchmark problems and a co-culture bioreactor using noisy data. Subsequently, we show how data-driven flow-maps can speedup forward and inverse UQ analyses of a dynamic microbial electrosynthesis reactor, achieving up to a 100-fold gain in computational speed.

2.2 Methods

In this section, we present the idea of flow-map (de)composition for dynamical nonlinear systems. We first introduce the notion of flow-map functions, which we seek to approximate in a data-driven manner based on time-evolution of system states. This is followed by a discussion on the data generation strategy and the PCK method used in this work to approximate flow-map functions for the variables of interest.

Flow-map Compositions

Consider a dynamical, time-invariant, nonlinear system described by

$$\frac{d\boldsymbol{s}}{dt} = \boldsymbol{f}(\boldsymbol{s}, \boldsymbol{x}), \quad \boldsymbol{s}(t=0) = \boldsymbol{s}_0, \tag{2.1}$$

where $\boldsymbol{s} \in \mathbb{R}^{n_s}$ is the vector of state variables with initial conditions $\boldsymbol{s}_0$, $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the vector of input variables, and $\boldsymbol{f}(\boldsymbol{s}, \boldsymbol{x}) : \mathbb{R}^{n_s} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_s}$ is the vector of (possibly unknown) system equations; $\mathbb{R}$ denotes the set of real numbers. Eq. (2.1) describes the time-evolution of the states $\boldsymbol{s}$ of a nonlinear system as a function of the inputs $\boldsymbol{x}$. Notice that in this work the inputs $\boldsymbol{x}$ can represent either model parameters, or manipulated input variables to a biochemical system, as will be discussed later.

A flow-map function is a mapping that predicts the transition of a dynamical system from the current to future state [220]. We define a flow-map function Φ_δ as

$$\boldsymbol{s}(t+\delta; x) = \Phi_\delta(\boldsymbol{s}_t, x), \tag{2.2}$$

$$\boldsymbol{s}(t+\delta; \boldsymbol{x}) = \boldsymbol{s}(t; \boldsymbol{x}) + \int_t^{t+\delta} \boldsymbol{f}(\boldsymbol{s}(t'; \boldsymbol{x}), \boldsymbol{x}) dt', \tag{2.3}$$

where δ is a time-lag (i.e., integration time step). Eq. (2.3) describes the one-step transition between the states of a system, in some interval $(t, t+\delta)$. The integral term that appears in Eq. (2.3) can be considered as a flow-map residual since it represents the discrepancy between the current and future states. Although Eq. (2.1) provides a continuous-time description of a dynamical system, the notion of transitioning among states, as implied by Eq. (2.3), hinges on discretizing the time domain over which the system evolves. Accordingly, the idea of flow-map compositions can be applied to compose a sequence of one-step transitions to define state trajectories over time [220]. Once a sequence of flow-maps $\{\Phi_{\delta_1}, \Phi_{\delta_2}, \ldots, \Phi_{\delta_K}\}$ is established, the flow-maps can be used to predict the states $\boldsymbol{s}$ at any discrete time instant using the K-fold composition

$$\Phi_\Delta = \Phi_{\delta_K} \circ \cdots \circ \Phi_{\delta_1}, \tag{2.4}$$

where $\circ$ denotes the function composition operator and Δ is the sum of the time-lags over the K discrete time steps (i.e., $\Delta = \sum_{j=1}^{K} \delta_j$). Eq. (2.4) indicates that, starting from some initial states, the K-fold flow-map function Φ_Δ governs the state transitions over the time-lags $\delta_1, \ldots, \delta_K$ wherein at each discrete time step the states are a function of the previous states as given by Φ_{δ_j}. Note that, in general, the time lags δ_j in Eq. (2.4) need not be the same.

In practice, the set of differential equations in Eq. (2.1) describing the system dynamics may not be known, or, when known, their numerical solution may be expensive. In this paper, we aim to learn an approximate surrogate for the flow-map function Φ_δ in Eq. (2.2) from high-fidelity simulation or experimental data. Data-driven flow-map models can be established from simulation data to provide an efficient surrogate for expensive computational models of the form in Eq. (2.1) that, for example, rely on numerical integration of a large number of highly nonlinear and stiff differential equations, as is commonly the case for complex biochemical systems. Notice that in this case, data-driven flow-map models essentially approximate a numerical integrator of the differential equations in Eq. (2.1). Alternatively, in the absence of any knowledge about the governing equations (i.e., functions $\boldsymbol{f}$ in Eq. (2.1)), flow-map models can be directly learned from experimental observations in order to discover the unknown system dynamics.

Data Generation

The data generation and model training strategy adopted in this work is summarized in Figure 2.1.

Consider that we have access to one or more, N_T in total, trajectories of state variables $\boldsymbol{s}_k$ over a discrete-time horizon $k = 0, 1, \cdots, T-1$, where k is the discrete-time index and T is the length of the time horizon of the training trajectories. As a remark, note that $k = 0$ corresponds to t_0, i.e., the initial time, $k = 1$ to $t_1 = t_0 + \delta_1$, $k = 2$ to $t_2 = t_0 + \delta_1 + \delta_2$ and so forth. The trajectory can be generated either from simulations or experiments. For some time interval indexed by k and corresponding time t_k where the states are known, we observe a transition in states $\boldsymbol{s}_k \rightarrow \boldsymbol{s}_{k+1}$. Hence, we have the current states (at time t_k) and obtain the future states (at time $t_k + \delta_k$), while the interval length represents the time lag δ_k. Moreover, within each interval, other inputs $\boldsymbol{x}_k$, such as manipulated inputs to the system, may be varying and thus should be accounted for in the data collection procedure. In

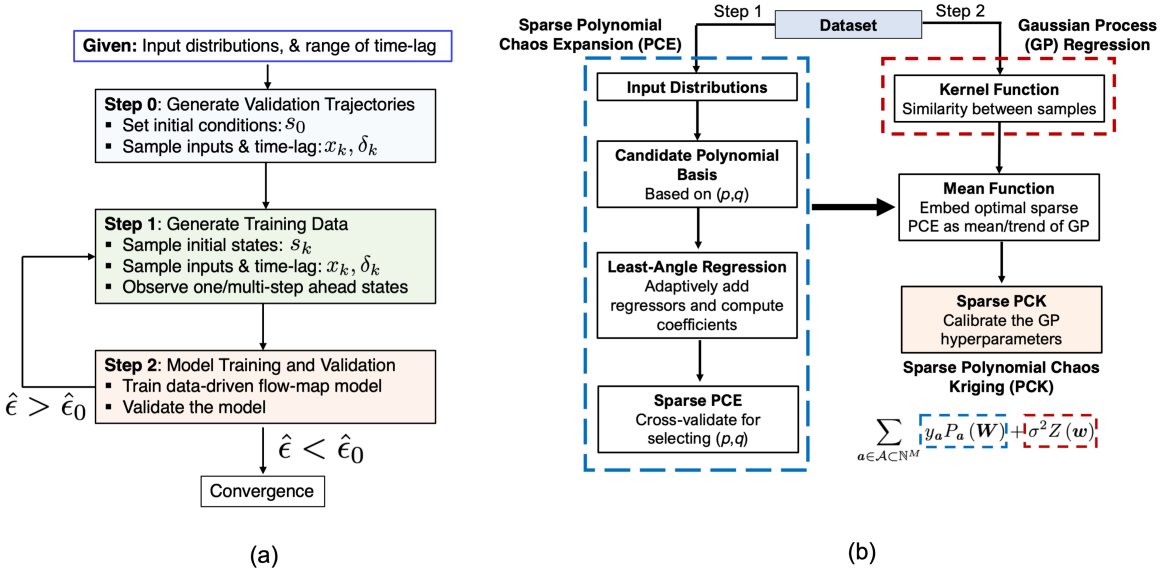

Figure 2.1: (a) Algorithm for data generation and training of data-driven flow-map models. Validation trajectories are first generated. Then, one/multi-step ahead simulations or experiments are performed to observe successor states given the initial states, inputs, and time-lag. Subsequently, the data-driven flow-map model is trained. Finally, the prediction accuracy of the trained model is assessed against the long-time validation trajectories. If the prediction accuracy $\hat{\epsilon}$ is larger than some pre-specified threshold $\hat{\epsilon}_0$, the model training and validation process will be repeated. The training procedure for PCK is depicted in (b). Several parameters must be selected during the model training, including the polynomial order, hyperbolic truncation parameter, covariance function and the regression method used for estimating the expansion coefficients.

summary, each time interval in a trajectory contains the information for a one-step transition and, therefore, yields a single sample for the dataset.

Accordingly, at each time instant k, a single training sample is acquired, consisting of the current states s_k, the input values of interest x_k, and the lag time δ_k, i.e., those are the input variables in the dataset, while the corresponding label is the set of states that the system arrives at, i.e., s_{k+1}. Thus, our dataset (or experimental design) looks as follows: the input data have the form $\mathcal{X} = \{(\boldsymbol{s}_0, \boldsymbol{x}_0, \delta_0), (\boldsymbol{s}_1, \boldsymbol{x}_1, \delta_1), \ldots, (\boldsymbol{s}_{T-1}, \boldsymbol{x}_{T-1}, \delta_{T-1})\}$ while the corresponding outputs are $\mathcal{O} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_T\}$. Note that there is usually some degree of freedom in choosing the lag time δ in simulations, whereas the choice of δ is often limited by how fast measurements can be acquired in experiments. For trajectory generation, it is crucial to vary the initial conditions $\boldsymbol{s}_0$ and inputs $\boldsymbol{x_k}$ within some allowable range, as well as the time lag δ whenever applicable. The training data must cover a wide range of state, input and time lag values, as relevant to the application of the trained models. We note that an effective strategy for generating simulation data is via one-step transitions, i.e., trajectories of length equal to 1. That is, instead of generating an entire trajectory given some initial conditions $\boldsymbol{s}_0$, we can randomly sample the state-space, along with the input parameters and time lag, in order to compute the corresponding future states. The sampling step (step 1)

is summarized in Figure 2.1(a). We remark that, although random sampling is used here to generate the training data, probabilistic models such as PCK used in this work provide confidence estimates on their predictions that can be used towards active learning-based sampling (e.g., see [290]).

Data-driven Flow-maps using Polynomial Chaos Kriging

In this work, we use sparse PCK [204, 158] to discover a data-driven flow-map model $\widetilde{\Phi}(\boldsymbol{w}_k)$ for the dynamical system in Eq. (2.1), i.e., step 2 in Figure 2.1(a). Note that since the time-lag is part of the inputs, we drop the δ subscript from the flow-map function symbol. The PCK training is summarized in Figure 2.1(b). Let us denote the vector of current states, input variables, and lag time by $\boldsymbol{w}_k = [\boldsymbol{s}_k^\top \ \boldsymbol{x}_k^\top \ \delta_k]^\top \in \mathbb{R}^M$, where $M = n_s + n_{\underset{\sim}{x}} + 1$. Thus, we denote the data-driven approximation of the flow-map in Eq. (2.2) by $\widetilde{\Phi}(\boldsymbol{s}_k, \boldsymbol{x}_k, \delta_k) : \mathbb{R}^{n_s} \times \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_s}$. The main benefits of using PCK for constructing data-driven flow-map models include: (i) being more data efficient, especially as compared to data intensive feedforward neural networks [272], when used for discovery of biological system dynamics from system observations; (ii) offering significant improvements in the computational efficiency of data generation for surrogate modeling for dynamical systems as compared to time-frozen polynomial chaos approaches [153, 154] ; and (iii) characterizing the uncertainty of model predictions. To this end, PCK combines the global approximation capability of polynomial chaos expansions, extensively used for surrogate modeling of (bio)chemical systems (e.g., [64, 199, 187]), with the local interpolation scheme of Kriging (i.e., Gaussian processes (GP) [227]). The polynomial structure of PCK makes its training data efficient, whereas Kriging offers the ability to quantify the uncertainties of model predictions.

In the context of PCK, $\boldsymbol{w}_k$ is a realization multivariate random variable W with a (known) joint probability distribution f_W, i.e., $W \sim f_W$. The PCK approximation of the flow-map is defined as

$$\boldsymbol{\mathcal{Y}} = \widetilde{\Phi}(\boldsymbol{w}) = \sum_{\boldsymbol{a} \in \mathcal{A} \subset \mathbb{N}^M} y_{\boldsymbol{a}} P_{\boldsymbol{a}}(\boldsymbol{W}) + \sigma^2 Z(\boldsymbol{w}), \tag{2.5}$$

where $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{n_s}$ denotes the predicted variables (QoIs) at step $k+1$, which are typically a subset of the states $\boldsymbol{s}$. The first term in Eq. (2.5) describes the trend (or mean) of the a GP using a polynomial chaos expansion (PCE), while the second term $Z(\boldsymbol{w})$ describes the variance of the predicted variable. $P_{\boldsymbol{a}}(\boldsymbol{W})$ represent the multivariate polynomial basis functions that are orthogonal with respect to the probability distribution f_W over the support $\mathcal{D}_W$ of the distribution, i.e., the range over which random numbers are defined and can be drawn from. The orthogonality condition implies that

$$\mathbb{E}\{P_i(\boldsymbol{W}) P_j(\boldsymbol{W})\} = \int_{\mathcal{D}_{\boldsymbol{W}}} P_i(\boldsymbol{w}) P_j(\boldsymbol{w}) f_{\boldsymbol{W}}(\boldsymbol{w}) d\boldsymbol{w} = \delta_{ij}, \quad \forall i, j \in \mathbb{N}^M, \tag{2.6}$$

with $\mathbb{E}$ being the expectation operator and δ_{ij} the Kronecker delta;

Eq. (2.6) gives the tensor product of M univariate polynomials that are orthonormal with respect to their corresponding marginal probability distribution. Optimal L_2-convergence of the expansion of orthogonal polynomials has been established based on the Wiener-Askey

scheme for various probability distributions [313, 40], although arbitrary orthogonal basis functions with sub-optimal convergence can also be constructed directly from moments of the random variable $\boldsymbol{W}$ [196]. The truncated polynomial chaos expansion takes the form

$$\sum_{\boldsymbol{a}\in\mathcal{A}} y_{\boldsymbol{a}} P_{\boldsymbol{a}}\left(\boldsymbol{W}\right), \tag{2.7}$$

where $y_{\boldsymbol{a}}$ are the coefficients of the basis functions. The order of the expansion is dictated by the multi-index $\boldsymbol{a} \in \mathcal{A}$, with $\mathcal{A} \subset \mathbb{N}^M$ being the set of the multi-indices kept in the truncated expansion. The truncation scheme aims to limit the infinite expansion of the trend to a series of maximum order p. To address the challenges that arise due to increasing the order of the polynomial basis for better approximation and/or the large dimension of $\boldsymbol{w}$ a q-norm scheme [23] limits the possible multi-index values according to:

$$\mathcal{A}^{M,p,q} = \{\boldsymbol{a} \in \mathcal{A}^{M,p} : ||\boldsymbol{a}||_q \leq p\}, ||\boldsymbol{a}||_q = \left(\sum_{i=1}^{M} a_i^q\right)^{\frac{1}{q}}. \tag{2.8}$$

The latter notation represents a vector consisting of M elements that are all natural numbers. Therefore, the multi-index here is essentially an extended index that represents the order of each monomial that participates in each polynomial term in the PCE. Note that originally the multi-index represents any combination of polynomials of any arbitrary order if $\mathcal{A} = \mathbb{N}^M$. Nevertheless, to keep the expansion of the trend term in Eq. (2.5) finite and tractable, it must be truncated up to a finite order p based on the q-norm. Therefore, the allowed values of the multi-index are determined by the tuple (p, q). In addition, for the GP-related term, $Z\left(\boldsymbol{w}\right)$ is a standard normal random distribution with variance σ^2.

As described, the multivariate random variable $\boldsymbol{W}$ consists of the states $\boldsymbol{s}$, input variables $\boldsymbol{x}$, and time lag δ. When $\boldsymbol{x}$ corresponds to uncertainties of a computational model (e.g., uncertainties in model parameters and/or initial conditions), their probability distribution is typically available *a priori* from parameter inference. As such, their respective polynomial basis functions can be chosen according to the Wiener-Askey scheme (e.g., Hermite basis for Gaussian distributions, Legendre for uniform distributions). On the other hand, when $\boldsymbol{x}$ corresponds to manipulated variables of a system, as is the case in the discovery of system dynamics, the input variables can typically be modeled as uniform distributions within a known range. The time lag δ can also be modeled as a uniform distribution within some range of interest for the application at hand. However, the distribution of states s_k is dependent on the realized state trajectories when the training data are generated and, thus, cannot be established *a priori*. Here, we assume states follow a multivariate Gaussian distribution with a mean and covariance computed from the training samples.

The coefficients $y_{\boldsymbol{a}}$ of the polynomial chaos expansion can be determined in a non-intrusive manner via solving a least-squares problem [22]. Here, we induce further sparsity by modifying the coefficient estimation problem to a L_1-regularized regression problem [101]. The regularized coefficient estimation problem can be efficiently solved using the least-angle-regression (LAR) algorithm [68], which estimates the coefficients of the most relevant terms of the expansion, setting the rest of the coefficients to zero. Moreover, $Z(\boldsymbol{w})$ in Eq. (2.5) is defined in terms of a kernel function $R(|\boldsymbol{w} - \boldsymbol{w}'|, \theta)$, i.e., a function that provides some measure of similarity between different realizations of the random variable $\boldsymbol{W}$. Here, we use the

Matérn kernel function [227]. Overall, the parameters of the PCK that must be determined using the training data include the coefficients $y_{\boldsymbol{a}}$ of the trend, the variance term σ^2, and the hyperparameters θ of the kernel function. This is efficiently done via maximum-likelihood estimation [204].

In this work, the following procedure is used for deriving the PCK flow-maps using the data generation scheme of Section 2.2. We use the sequential PCK approach proposed in [204], where a PCE is first trained based on the available data and is then embedded as the trend of PCK. This procedure is shown in Figure 2.1(b). For training the PCE, we allow the PCE's maximum order to vary from 1 to 5; higher order polynomials are avoided to retain a smaller expansion (i.e., less degrees of freedom) and mitigate overfitting. The truncation factor q is varied from 0.7 to 0.85 since the resulting maximum order of the polynomials will ensure that we do not have highly nonlinear interaction terms while allowing for elimination of few of interaction terms. The optimal value of q is chosen based on cross-validation. The hyperparameters of PCK are selected using a data-driven optimization algorithm, namely the covariance matrix adaptation–evolution strategy [100]. Finally, it should be noted that using PCK as the surrogate model places some limitation on the number of input variables $\boldsymbol{w}$ that can be handled. Typically, GP-based models are utilized for lower dimensional spaces due to the "curse-of-dimensionality" [287] [1]. On the other hand, sparse PCEs can effectively deal with high input dimensions, thanks not only to the truncation schemes that are employed, but also the sparse regression schemes, e.g., LAR, that include only the most informative terms in the expansion, thus minimizing the number of unknown coefficients. To quantify the quality of the PCK predictions during the training phase we use the leave-one-out cross-validation (LOOCV) error that is estimated from the training data (during LAR in Figure 2.1). When one-step ahead test samples are available, validation errors can readily be evaluated and are used for cross-validation.

So far, we have described the flow-map modeling procedure by utilizing one-step transition data, hence the PCK model is able to predict one step ahead states. Nevertheless, we are interested in long-term integration of the dynamical system. To this end, Figure 2.2 shows how a data-driven flow-map model can be used sequentially to predict the time-evolution of the states of a dynamical system.

As stated, at each time instant k the PCK flow-map model essentially "integrates" the states forward in time by δ_k until the final time is reached. Hence, we can also assess the ability of the data-driven flow-map models in approximating the integration operator and, hence, their predictive accuracy over a multi-step integration horizon. This is the d Given $i = 1, \ldots, N_V$ validation state trajectories, each of which of length T_i, we define the

[1] Not only the concept of Euclidian distance (which is the main feature of kernels) becomes less meaningful in higher dimensions, but also higher-dimensional input spaces require more data to be efficiently discovered, hence rendering the inference part of GP-based models harder. This challenge can be partially adressed via sparse GPs [257, 282]

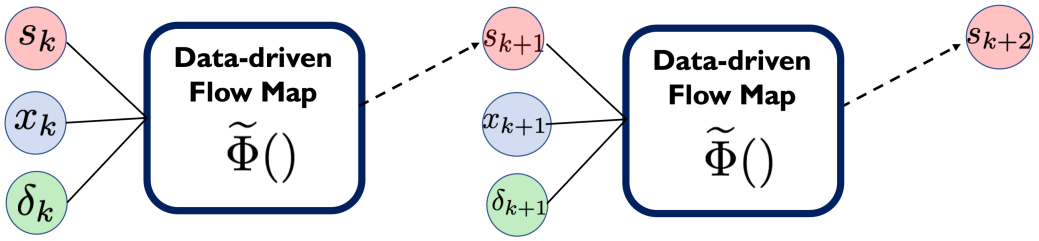

Figure 2.2: Data-driven flow-map models for predicting the state variables of a dynamical system over time. The flow-map model $\widetilde{\Phi}$ takes the current states $\boldsymbol{s}_k$, inputs $\boldsymbol{x}_k$, and lag time δ_k at a discrete-time instant k as inputs to predict the states $\boldsymbol{s}_{k+1}$ at the subsequent time instant $k+1$. By sequentially repeating this procedure, the time-evolution of the states in relation to the inputs can be established.

normalized, time-averaged prediction error of the state variables (QoI) , ϵ_i, as

$$\epsilon_i = \sum_{k=0}^{T_i} \frac{1}{T_i} \frac{||\boldsymbol{\mathcal{Y}}_{k,i} - \boldsymbol{\mathcal{Y}}_{k,i}^{\text{true}}||_2}{||\boldsymbol{\mathcal{Y}}_{k,i}^{\text{true}}||_2} \tag{2.9a}$$

$$\hat{\epsilon} = \frac{1}{N_V} \sum_{i=1}^{N_v} \epsilon_i, \tag{2.9b}$$

where $||\cdot||_2$ is the 2-norm of a vector; $\boldsymbol{\mathcal{Y}}_{k,i}^{\text{true}}$ and $\boldsymbol{\mathcal{Y}}_{k,i}$ are, respectively, the vector of state variables (QoIs) in the validation dataset and those predicted by the data-driven flow-map models at time instant k for each validation run i. In the remainder, we refer to ϵ_i as the mean trajectory error (MTE), whereas $\hat{\epsilon}$ is the average MTE over all validation trajectories.

2.3 Data-Driven Discovery of Dynamical Systems

In this section, we apply the PCK-based flow-map modeling approach to learn the dynamics of several benchmark systems using limited data. The first case study, based on the Morris-Lecar system, compares the performance of the PCK model with neural network modeling results of [272]. The second case study, based on the Lorenz system, focuses on reconstructing the dynamics of a chaotic system in which variations in parameters significantly change the solution landscape. Lastly, we show how the flow-map modeling approach can be used for discovering the dynamics of a co-culture bioreactor under noisy observations and how the variance term of PCK provides a measure of uncertainty of model predictions.

Morris-Lecar System

The first benchmark problem is the Morris-Lecar system [174], which describes neuronal excitability. This system was used in [272] to examine neural network-based flow-map models

for the discovery of nonlinear dynamics. In particular, a residual neural network was used to represent the data-driven flow-map model, in which only the flow-map residual is learned by skipping the input connection to the neural network and adding it to the output of the latter. Here, we aim to recreate the results of the aforementioned work, demonstrating the data efficiency of the proposed PCK approach to data-driven reconstruction of dynamics. The dynamics of the Morris-Lecar system are described by

$$\begin{aligned} C_M \frac{dV}{dt} &= -g_L(V - V_L) - g_{Ca}(V - V_{Ca})M_\infty \\ &\quad - g_k(V - V_K)N + I_{app} \end{aligned} \tag{2.10a}$$

$$\frac{dN}{dt} = \lambda_N(N_\infty - N), \tag{2.10b}$$

where V (mV) is the voltage difference between the sides of the membrane and N represents the probability for the potassium channel being open.

$$M_\infty = 0.5\left(1 + \tanh\frac{V - V_1}{V_2}\right) \tag{2.11}$$

$$N_\infty = 0.5\left(1 + \tanh\frac{V - V_3}{V_4}\right) \tag{2.12}$$

$$\lambda_N = \phi\cosh\frac{V - V_3}{2V_4}. \tag{2.13}$$

The Type I model parameters M_∞, N_∞ and λ_N depend on the voltage and are defined as

$$\begin{aligned} &[C_M, g_L, V_L, g_{Ca}, V_{Ca}, g_k, V_k, V_1, V_2, V_3, V_4, \phi] = \\ &[20, 2, -60, 4, 120, 8, -84, -1.2, 18, 12, 17.4, 0.066], \end{aligned} \tag{2.14}$$

taken from [272]. Here, it is assumed that the model parameters are fixed, as we aim to reconstruct the system dynamics as a function of injected current $x_k = I_{app}$ that can vary within the range $[0,\ 300]$ A. Specifically, we aim to predict the long-term system dynamics, starting from given initial conditions, under a fixed I_{app}. To compare our results with those in [272], δ_k was chosen to be 0.2 ms; we did not consider the time-lag as part of the PCK model. In other words, the input data consisted of samples in the form $(V_k, N_k, x_k) \to (V_{k+1}, N_{k+1})$, where x_k is constant for every $k \in [0, T]$ for a given trajectory. This system exhibits a saddle node bifurcation, which leads to an oscillatory behavior depending on the value of input I_{app}. Thus, the data-driven flow-map model must capture the oscillatory behavior for different values of I_{app}.

To train the PCK-based flow-map model, we generated one-step ahead samples of the states V_k and N_k by randomly drawing the initial states from $[-75,\ 75] \times [0,\ 1]$. Here, we first examine the convergence error of the flow-map model to characterize how many samples of states would be necessary for data-driven reconstruction of the system dynamics. We quantify the convergence error in terms of the average MTE in Eq. (2.9) based on three validation trajectories generated for $I_{app} = \{0,\ 60,\ 150\}$. Figure 2.3 shows the average MTE

estimated over 1,000 time steps in relation to the number of training samples, where the vertical line around each error represents one standard deviation based on 5 repetitions of the analysis. It is evident that the error converges after about 160 samples, suggesting that a limited number of training samples is needed.

Figure 2.4 shows the reconstructed dynamics by the PCK-based flow-map model trained using 240 samples in comparison with the true dynamics.

As can be seen, there is no visible discrepancy between the true time-evolution of the system and the reconstructed dynamics. The system exhibits a bifurcation behavior, as evident from the phase plots shown in Figure 2.4(c), (f), (i). Yet, the PCK-based flow-map model is able to capture this complex behavior and accurately predict the system dynamics over a long-time horizon. We note that a 500-fold saving in the number of training samples is observed as compared to [272] in which a residual neural network representation was used for the flow-map model. This is while the PCK model also yields slightly more accurate predictions.

Lorenz System

We now consider a chaotic dynamical system based on the well-known Lorenz benchmark problem [264]. The Lorenz system has been widely used in the data-driven modeling lit-

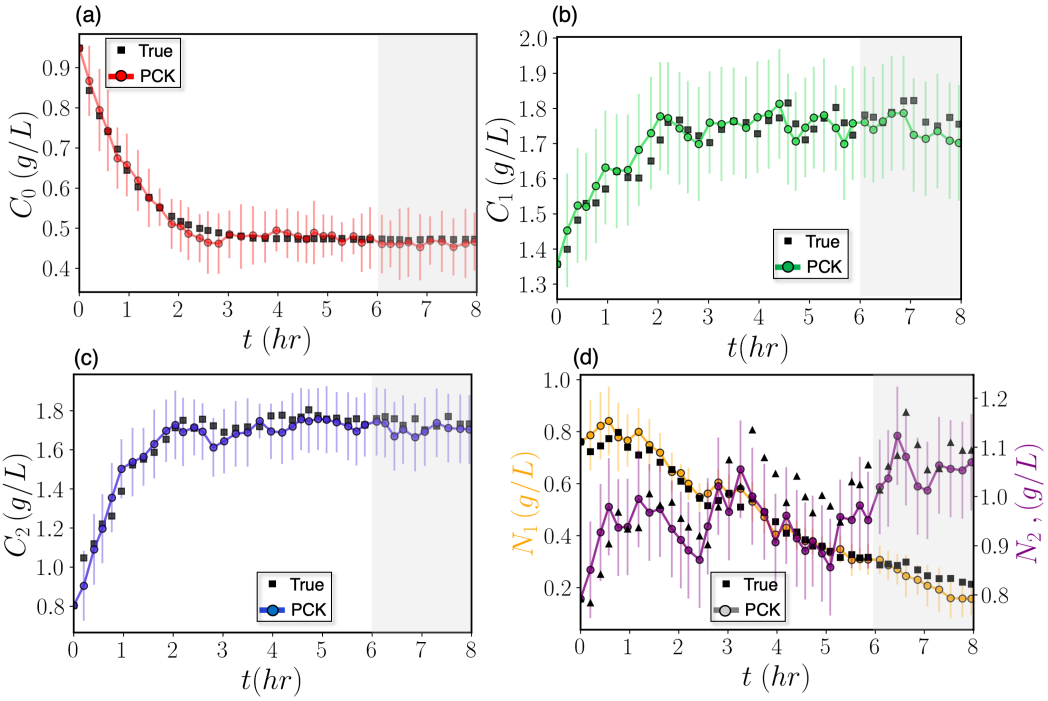

Figure 2.3: The average mean trajectory error, $\hat{\epsilon}$, of the PCK-based flow-map model for the Morris-Lecar system in relation to the number of training samples, N_s. The error is estimated based on three validation trajectories generated for the input I_{app} values $\{0,\ 60,\ 150\}$. The vertical bars represent the standard deviation of the error estimated based on 5 repeats of the training.

erature (e.g., [66, 224]). The Lorenz system is described by the following set of nonlinear ordinary differential equations

$$\frac{da}{dt} = \sigma(b - a) \tag{2.15a}$$

$$\frac{db}{dt} = a(\rho - c) - b \tag{2.15b}$$

$$\frac{dc}{dt} = ab - \beta c, \tag{2.15c}$$

where $\boldsymbol{s} = [a,\ b,\ c]^\top$ are the system states and $\boldsymbol{x} = [\sigma,\ \rho,\ \beta]^\top$ are the uncertain model parameters. Chaotic behaviors can be encountered in various chemical and biological systems, including in the growth of biological populations with non-overlapping generations [166] and

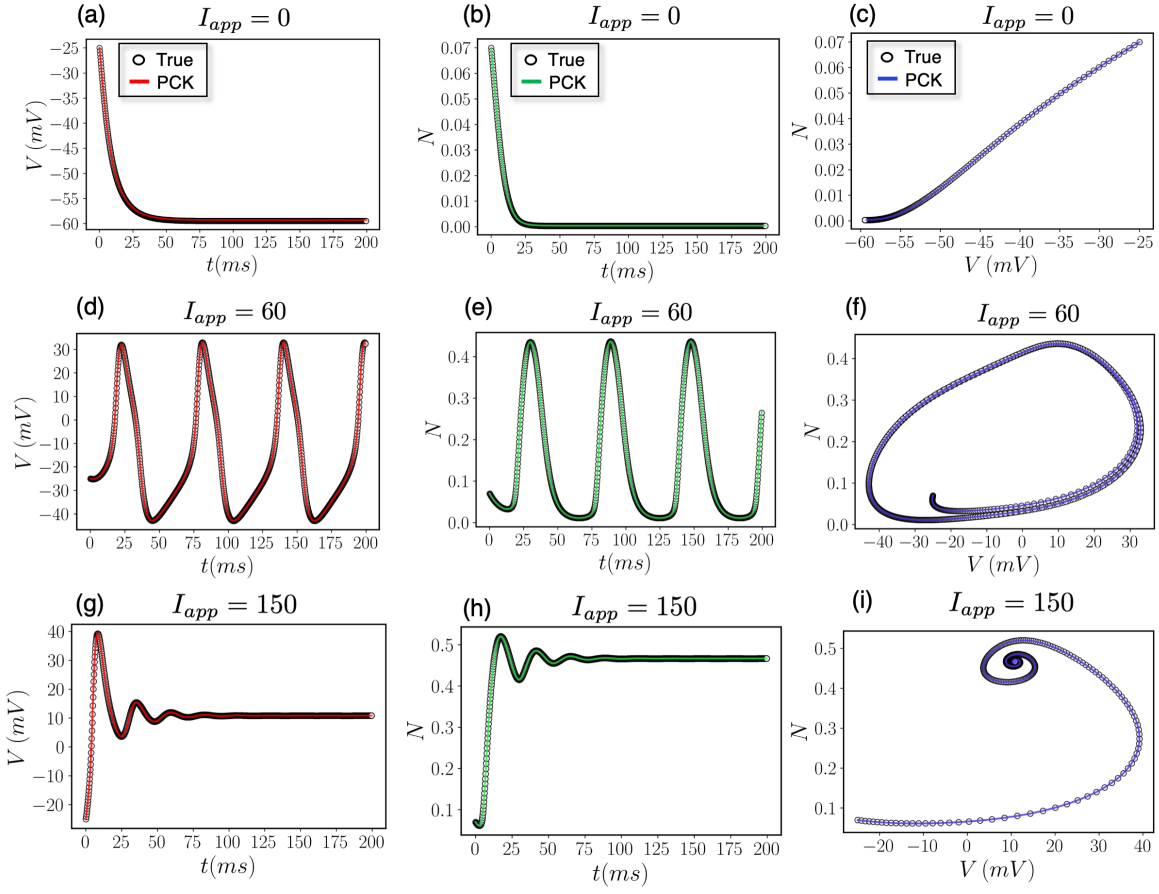

Figure 2.4: Reconstructed dynamics of the Morris-Lecar system by the PCK-based flow-map model in comparison with the true system dynamics for the input I_{app} values $\{0,\ 60,\ 150\}$. The PCK-based flow-map model is trained using 240 samples. The left column shows the time-evolution of voltage difference, V; the middle column shows the time-evolution of the channel opening probability, N; and the right column shows the corresponding phase plots.

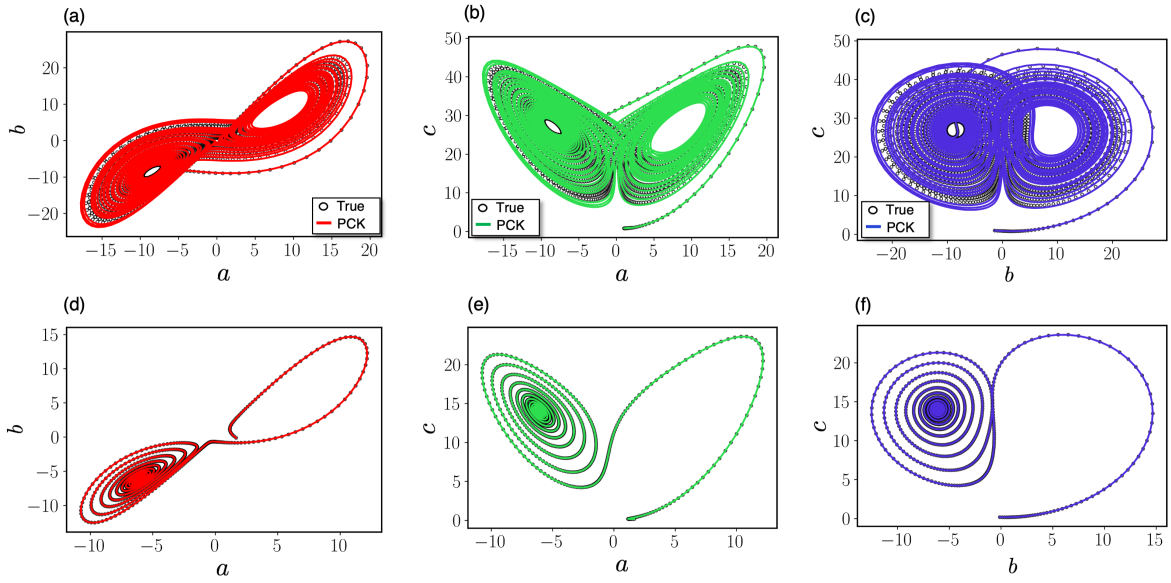

Figure 2.5: Phase plots of the reconstructed dynamics of the Lorenz system by the PCK-based flow-map model in comparison with the true system dynamics for different values of model parameters. Subplots (a)-(c) correspond to the model parameters $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$. Subplots (d)-(f) correspond to the model parameters $\sigma = 10$, $\beta = 8/3$, and $\rho = 15$.

the peroxidase–oxidase oscillator [188]. Here, we consider a constant time-lag $\delta = 0.01$ that captures the intrinsic time-scale of the system [35].

The Lorenz system exhibits a chaotic behavior based on the initial conditions $\boldsymbol{s}_0$, while its long-term behavior is highly affected by the uncertain parameters $\boldsymbol{x}$. The nominal initial conditions and parameters of the system are, respectively,
$\boldsymbol{s}_0 = [1.9427,\ -1.4045,\ 0.9684]^\top$ and $\boldsymbol{x}_0 = [10,\ 28,\ 8/3]^\top$, for which the system oscillates around two attractors. Here, the training data consisted of 500 random samples of the state-space $\boldsymbol{s}$ within the range $[-10,\ 10] \times [-10,\ 10] \times [-10,\ 10]$ and the parameters $\boldsymbol{x}$ within the range $[8,\ 12] \times [10,\ 30] \times [1,\ 5.5]$. We used two validation trajectories to compare the true system dynamics with those reconstructed by the PCK-based flow-map model: one trajectory based on the nominal initial conditions and parameters and the other based on $\boldsymbol{x} = [10,\ 15,\ 8/3]^\top$ and $\boldsymbol{s}_0 = [1.6655,\ -0.1178,\ 0.1748]^\top$.

Figure 2.5 shows phase plots of the reconstructed oscillatory dynamics of the Lorenz system, in comparison with the true system dynamics, over a simulation horizon of 5,000 time steps. We observe that the qualitative behavior of the Lorenz system is different when the parameter ρ is varied, while the PCK-based flow-map model is able to reconstruct the dynamics in both cases. The MTE is 0.522 for the nominal validation trajectory and 0.0013 for the second validation trajectory. Although the error for the nominal validation trajectory seems relatively high, the main characteristics of the true dynamics are adequately captured, as evident from Figure 2.5(a)-(c). That is, the limit cycles, the amplitude of oscillation and period are adequately captured. These predictions are consistent with those reported in

[224]. However, we note that reconstruction of the Lorenz dynamics using neural networks typically requires on the order of a few thousands of training samples [240, 35], whereas the PCK model here was trained using 500 samples.

Transient Co-culture System

In this case study, we demonstrate the ability of PCK-based flow-map models to learn the transient behavior of a co-culture system with variable inputs. In particular, we focus on the startup dynamics of a continuous bioreactor driven by the competition of several auxotrophs [190]. To emulate data collection from a real system, we use a nonlinear dynamic model of the bioreactor [286] to generate observations of the system states, which are then corrupted with independent and identically distributed state-dependent measurement noise $e_i \sim \mathcal{N}(0, 2.5 \times 10^{-2} s_k^i)$, with i being an index for the measured states and k the time index. The five state variables $\boldsymbol{s}_k$ of the bioreactor include: the population of the two species $N_1(Cells/L)$ and $N_2(Cells/L)$, the auxotrophic nutrients concentrations $C_1(g/L)$ and $C_2(g/L)$, and the common shared carbon source concentration $C_0(g/L)$. The bioreactor has three process inputs $\boldsymbol{x}_k$ that can be varied in time. The process inputs are the dilution rate D (hr^{-1}) that varies within the range $[0.75,\ 1.5]$ (hr^{-1}), as well as the feed substrate concentration of auxotrophs $C_{1,in}$ (g/l) and $C_{2,in}$, both varying in the range $[1.5,\ 2]$ (g/l). To generate data for training the PCK-based flow map models, short simulation "experiments" with a fixed length of $T = 30$ steps with $\delta_k \in [0.15,\ 0.25]$ hr^{-1} were performed. At each time step k during the multi-step experiments, inputs $\boldsymbol{x}_k$ were varied over the time interval δ_k and noisy observations of the states were collected.

For the validation plots of Figure 2.6, we begin by some random initial condition at $k = 0$, by applying an input x_0 over some interval δ_0. The model predicts the mean of the states at $k = 1$, as well as their variance.

The integration proceeds by taking a next step based on the mean value of the states at $k = 1$, predicting the states at $k = 2$. Using only the mean value to compute trajectories is probably the simplest way when Gaussian Process state space models are utilized, however, there are more sophisticated ways for the trajectory generation [107], which are beyond the scope of the paper. Note that properly incorporating the uncertainty in multi-step ahead predictions is a complicated issue addressed in the literature [214, 91]. Here, it suffices to use a deterministic function, e.g., the mean value of the data-driven flow-map model, to integrate in time since this way we avoid the major issue of using noisy inputs into our PCK model. The validation trajectories have a length of $N_k = 40$ steps ahead, extending slightly beyond the training range. Moreover, thanks to the nature of the PCK model, we can also simply characterize the confidence of the model to the prediction of the dynamics. To get some uncertainty estimates on the predicted trajectories, at each step k, we plot the $3\sigma(w_k)$ error bars around the mean. Overall, we observe that the true, noiseless trajectories are embedded within the confidence intervals of the PCK predictions.

2.4 Uncertainty Quantification of Expensive Computational Models

In this section, we demonstrate the utility of data-driven flow-maps for the UQ of a Microbial Electrosynthesis (MES) bioreactor using a high-fidelity computational model that is subject to uncertainty in model parameters and initial conditions. In particular, we show how flow-maps can be used as surrogate models for efficient sample-based approximation of distribution of state variables, global sensitivity analysis, and Bayesian parameter inference, when the original model is prohibitively expensive for a sample-based analysis.

We consider the batch MES bioreactor shown in Figure 2.7 for CO_2 fixation [1], with potential applications in space biomanufacturing [20]. The bioreactor consists of a well-mixed liquid bulk phase that contains dissolved CO_2, i.e., substrate. A microbial community forming a biofilm grows on the cathode of the bioreactor. The dissolved substrate diffuses into the biofilm through a linear boundary layer and is then consumed by bacteria towards

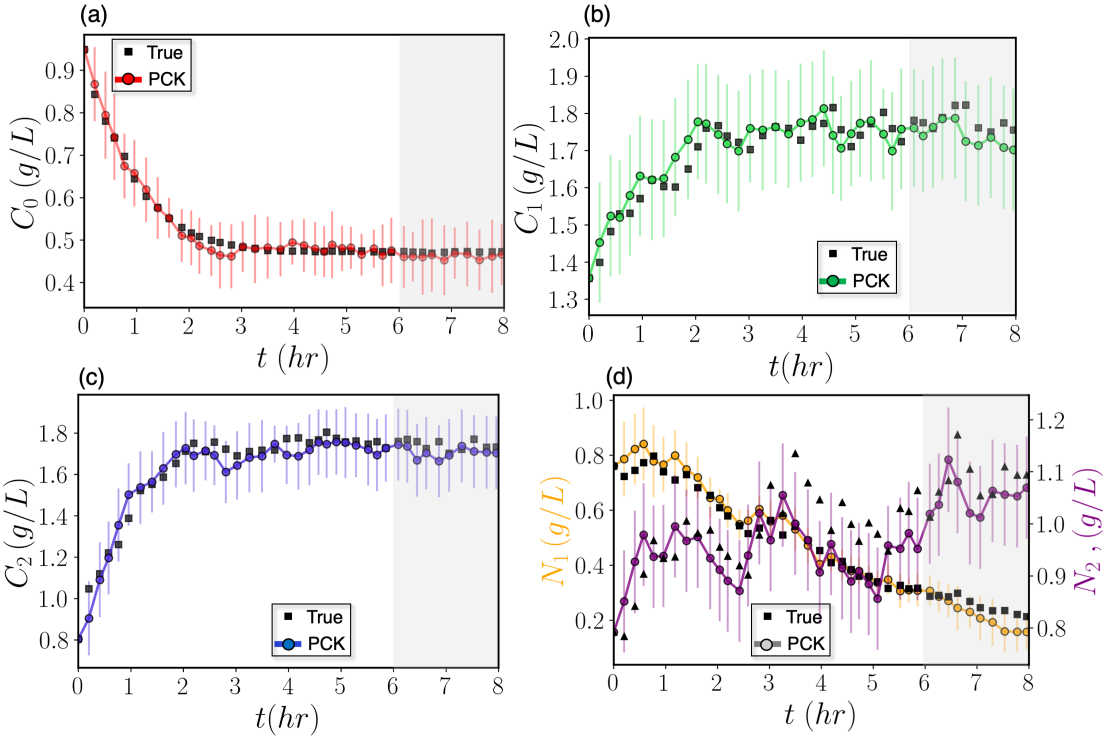

Figure 2.6: Predictions of the state variables of the transient co-culture system via the PCK-based flow-map models in comparison with the observed state trajectories. The colored lines/points correspond to the predicted trajectories by the mean of the PCK models, starting from some initial states at $t = 0$ hr. Black symbols represent the observed trajectories at specific snapshots during a validation run. Vertical error bars represent the uncertainty in the predictions of the PCK models, estimated as plus/minus two standard deviations from the mean value. The shaded areas correspond to a time interval that was not accounted for when training the PCK models.

the growth of the biofilm. This leads to spatial distribution of the substrate concentration within the biofilm. Voltage is applied to the cathode while the biofilm acts as a conductive matrix through which electron transport takes place. Both the substrate CO_2 in the biofilm and the local overpotential due to the current flux contribute to the biofilm growth kinetics described by the dual Monod-Nerst model [283].

A computational model of the dynamics of the MES bioreactor is adopted from [120, 161], with some modifications. Within the biofilm, the cell growth leads to the production of acetate as a metabolic product. A primary modeling approach in the aforementioned papers assumes the total biomass has a constant concentration and exists in two forms, active and inactive, each of which occupies some volume fraction. We assume that biomass exists only in active form, thus the equations describing the volume-fraction change within the film effectively become a single equation for the rate of change of film thickness, L_f, which is a differential state in our system. Moreover, the film growth is affected by a constant detachment rate. It is also assumed that the reaction occurs only within the biofilm, so the only source of acetate in the bulk phase comes from exchange with the biofilm through the boundary layer. We further assume the transport-reaction phenomena in the biofilm are much faster than the transport that occurs across the boundary layer and in the bulk phase; accordingly, the conservation laws inside the biofilm are considered to be in pseudo steady-state [120]. Hence, the computational model consists of a set of nonlinear second-order ordinary differential equations that describe the spatial distribution of substrate, acetate and overpotential within the biofilm, coupled with a set of first-order ordinary differential equations that describe the concentration of CO_2 in the bulk phase S_b, the acetate concentration in the bulk phase P_b, and the biofilm thickness L_f. As such, the three state variables of the system are

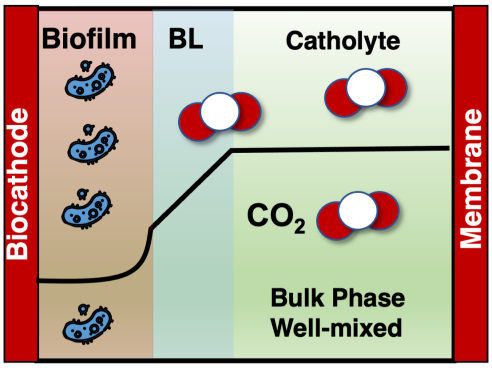

Figure 2.7: Schematic of the microbial electrosynthesis bioreactor. The bioreactor consists of 3 regions: the bulk phase, the biofilm, and a boundary layer (BL) in between. The black line represents a typical concentration profile of some species as predicted by the computational model used in this work. The concentration is assumed to be constant in the bulk phase, changing linearly across the boundary layer, and exhibiting a more complicated shape in the biofilm.

described by

$$\frac{dL_f}{dt} = (Y\hat{q} - r_d)L_f \tag{2.16a}$$

$$\frac{dS_b}{dt} = \frac{A_f}{V_r} j_S \tag{2.16b}$$

$$\frac{dP_b}{dt} = \frac{A_f}{V_r} j_P, \tag{2.16c}$$

where $Y(\frac{mgX}{mmolS})$ is the biomass yield coefficient, $\hat{q}(\frac{mmolS}{mgXdays})$ represents an average substrate consumption specific rate within the biofilm, $r_d\,(\frac{1}{days})$ is a detachment rate, $A_f\,(cm^2)$ is the cross-sectional area of the biofilm, and $V_r\,(cm^3)$ is the bioreactor volume. The mass balances for the substrate and product are a function of the flux of each species across the linear boundary layer described by

$$j_m = \frac{D_b}{L_b}(m_f(z = L_f) - m_b), \quad m = S, P, \tag{2.17}$$

where m denotes the species (i.e., substrate and product), $D_b\,(\frac{cm^2}{days})$ is the diffusivity coefficient in the boundary layer and $L_b(cm)$ is the thickness of the boundary layer. The subscript f denotes the species concentration in the film at position $z = L_f$. The equations that describe the diffusion phenomena within the film are given in the SI. In order to determine the concentrations at L_f, a boundary value problem (diffusion within the film) must be solved at each time step, as the concentrations in the biofilm are a function of the bulk concentrations. The computational model is fairly expensive for UQ analyses that rely on Monte Carlo sampling; each model run takes on average 4.5 minutes. The model is subject to time-invariant uncertainty in its parameters and initial conditions. Specifically, the model uncertainty comprises of the conductivity of the biofilm k_{bio}, the maximum growth rate μ_{max} of the Nerst-Monod model, the yield Y, the Monod affinity constant K_s, as well as the acetate production-related parameters α and β. These six uncertain parameters are assumed to follow a uniform probability distribution. Their nominal values are $[k_{bio},\ \mu_{max},\ Y,\ K_s,\ \alpha,\ \beta]^\top = [1 \times 10^{-3},\ 4.5,\ 0.25,\ 3.0,\ 0.1,\ 2 \times 10^{-5}]^\top$, while they vary uniformly $\pm 10\%$ from the nominal values.

In this case study, we construct data-driven flow-map models of the PCK form in Eq. (2.5) for the output variables $\boldsymbol{\mathcal{Y}} = [L_f\ S_b\ P_b]^\top$, such that the six sources of uncertainty constitute the vector of input variables $\boldsymbol{x}$ in Eq. (2.5). The three flow-map models, one for each QoI, were trained using simulation data generated via the computational model for lag times in the range of $\delta = [0.05,\ 0.1]$ days, which allow us to adequately capture the bioreactor dynamics. Notice that clearly the lag time δ must always be larger than the integration time step of the computational model.

The training dataset consists of full state trajectories, as well as one-step ahead samples of the states. We initially generate $N_T = 30$ trajectories, with fixed uncertain parameters in time, over a process time span from 0 to 3.5 days, which corresponds to approximately $T = 50$ samples per trajectory. Then, using the states $\boldsymbol{s}_k$ corresponding to each sample $\boldsymbol{w}_k$, we randomize the uncertain parameters and perform one-step ahead simulations. In this way, approximately 1,400 training samples were generated, while 800 samples are used

for training the PCK models. The rationale behind not randomizing the states is that the validation trajectories (step 0 of Figure 2.1) indicate that there is a high correlation among state values. For instance, as L_f grows in time (under insignificant detachment), S_b decreases due to consumption. Thus, for a given set of uncertain parameters and initial states, a few full state trajectories will help generate more informative training samples.

Figure 2.8 shows the predicted trajectories using the data-driven flow-map PCK model for a given realization of uncertainty and initial conditions, while the true trajectory is juxtaposed. The trajectories correspond to a time-march of 50 steps ahead. We observe a perfect agreement between the predicted and validation trajectories, with the average MTE for the three states being approximately $\hat{\epsilon} = 2.5 \times 10^{-4}$.

An important remark should be made here regarding the benefits of the presented flow-map approach to surrogate modeling of dynamical systems in comparison with the so-called time-frozen approaches discussed in Section 2.1. First, the flow-map models provide the flexibility to approximate the distribution of states at any time instant of interest without the need for constructing a separate surrogate model for each time instant, as in time-frozen surrogate modeling. For example, if we were to use a time-frozen approach, 50 separate PCK models would need to be constructed for each QoI to predict the time-evolution of the QoI distribution over the 50 time instants considered here. Thus, not only a flow-map modeling approach significantly reduces the number of surrogate models that must be constructed to only one model for each QoI, it also provides flexibility via alleviating the need to build the models at pre-specified time points. Furthermore, the flow-map modeling approach enables more efficient data generation. To clarify this point, let us assume that N_p realizations of uncertainty are sufficient for generating a rich training dataset that yields surrogate models with low approximation error. In the case of the time-frozen approach, we would require to generate N_p full state trajectories since the states must be observed at all time instants for all uncertainty realizations. This approach to data generation can become prohibitively expensive, in particular when data generation relies on expensive simulations. However, training the flow-map models, in principle, requires simulation of a limited number of full state trajectories (in this study, 25 trajectories), whereas N_p training samples can be straightforward generated via one-step ahead integration of the computational model. In

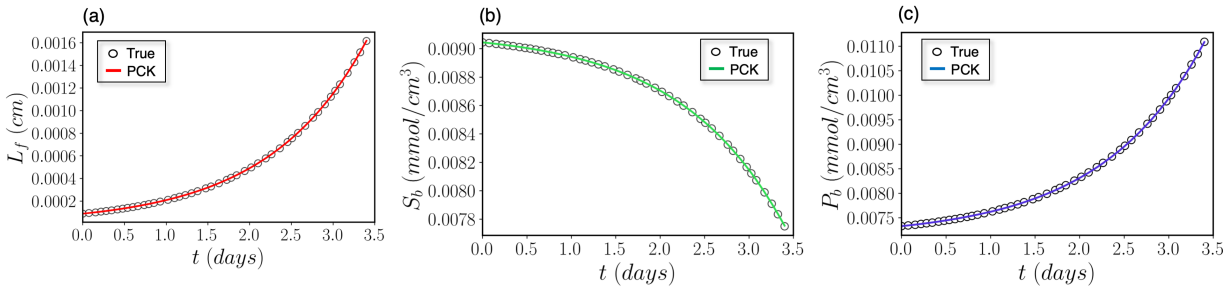

Figure 2.8: Predicted state trajectories of the the microbial electrosynthesis bioreactor: (a) biofilm thickness, L_f, (b) CO_2 concentration in the bulk phase, S_b, and (c) acetate concentration in the bulk phase, P_b. Hollow points represent the validation trajectories, while the solid lines represent the trajectories predicted by the PCK-based flow-map models.

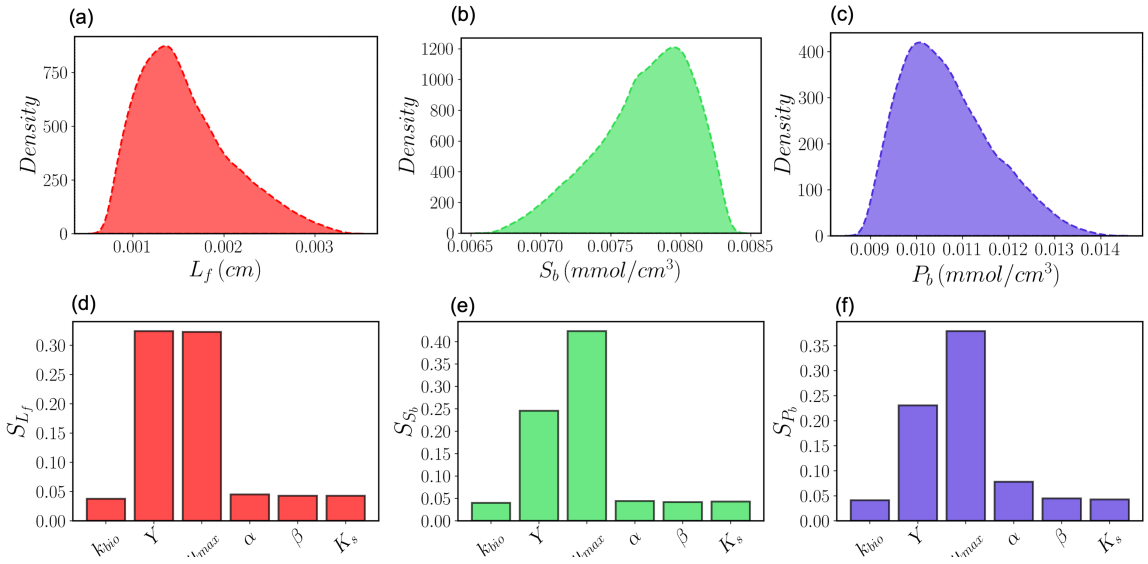

Figure 2.9: Fast uncertainty propagation and global sensitivity analysis of the the microbial electrosynthesis bioreactor using data-driven flow-map models of quantities of interest. Subplots (a)-(c) show the kernel density estimates of the distribution of the biofilm thickness (L_f), concentration of CO_2 in the bulk phase (S_b), and acetate concentration in the bulk phase (P_b) predicted by the PCK models at time $t = 3.5$ days. The distributions of L_f, S_b and P_b are approximated via Monte Carlo sampling using 20,000 realizations of uncertain model parameters, where a 100-fold computational speedup in sample-based approximation of the distributions is attained. Subplots (d)-(f) show the Borgonovo indices, denoted by $\mathcal{S}$, that quantify the global sensitivity of L_f, S_b and P_b at $t = 3.5$ days with respect to the six uncertain model parameters. The Borgonovo indices are approximated based on 20,000 uncertainty realizations.

the following, the use of PCK-based flow-map models is demonstrated for expensive UQ analyses.

Forward Uncertainty Propagation and Global Sensitivity Analysis

Here, we use the data-driven flow-map models for efficient uncertainty propagation via sample-based approximation of the distribution of the three output variables. Figure 2.9(a)-(c) shows the distribution of the states at $t = 3.5$ days.

To approximate their distribution, the flow-map models were evaluated using 20,000 realizations of the model uncertainty $\boldsymbol{x}$. Each run of the data-driven flow-map model takes on average less than 3 seconds,[2] as opposed to the average run time of 4 minutes of the computational model. This implies that the flow-map models significantly accelerate the

[2]Notice that the evaluation time of a PCK model depends on a multitude of factors, such as the degree of the polynomial basis functions, kernel type, and, mainly, amount of data used to train the model. Additionally, a kernel-based model such as PCK is more expensive to evaluate than a polynomial chaos expansion.

uncertainty propagation, enabling an approximately 100-fold increase in the computational speed. This is especially beneficial when the distributions are skewed (or bi-modal), as in Figure 2.9(a)-(c). In this case, a large number of samples, $\mathcal{O}(10^4 - 10^5)$, would be typically required for accurate sampled-based approximation of distribution, or statistical moments of the quantities of interest. Although not shown here, we can efficiently approximate the distribution of states at any time instance using trajectories generated by the surrogate model.

Moreover, we use the data-driven flow-map models to perform a global sensitivity analysis in order to asses the importance of the six uncertain model parameters, $\boldsymbol{x}$, on the state variables $\boldsymbol{\mathcal{Y}}$. This is done via evaluation of the Borgonovo indices [28], denoted by $\mathcal{S}$, which are based on the full distribution of some quantity of interest, as opposed to their statistical moments. The results of global sensitivity analysis of for the states at $t = 3.5$ days are shown in Figure 2.9(d)-(f), where each bar corresponds to a different uncertain model parameter. The Borgonovo indices are approximated using the same 20,000 samples used in forward UQ. We observe that the probabilistic uncertainty of yield Y and maximum growth rate μ_{max} have the most dominant effects on the variability of the three states, while the product concentration P_b is also significantly affected by the uncertainty in the parameter α, which is the metabolism-related productivity constant.

Bayesian Inference of Unknown Model Parameters

We now use the data-driven flow-map models to solve a Bayesian inference problem in order to infer the uncertain model parameters $\boldsymbol{x}$. Bayesian inference relies on Bayes theorem to estimate the posterior probability distribution of the unknown model parameters from available data. Here, noisy observations of L_f, S_b and P_b at time instants $\{0.5,\ 1.0,\ 1.5,\ 2.0,\ 2.5,\ 3.0,\ 3.5\}$ days constitute the dataset $\mathcal{D}$ used for parameter inference; measurement noise is modeled as a Gaussian distribution with zero mean and state-dependent variance. Once a vector of system measurements $\boldsymbol{d}$ at a time instant is observed, the change in our knowledge about the unknown parameters is described by Bayes' rule [122]

$$f_{\boldsymbol{x}|\mathcal{D}}\left(\boldsymbol{x}|\boldsymbol{d}\right) = \frac{f_{\mathcal{D}|\boldsymbol{x}}\left(\boldsymbol{d}|\boldsymbol{x}\right) f_{\boldsymbol{x}}\left(\boldsymbol{x}\right)}{f_{\mathcal{D}}\left(\boldsymbol{d}\right)}, \tag{2.18}$$

where $f_{\boldsymbol{x}|\mathcal{D}}$ denotes the posterior distribution of the uncertain parameters after observing the data; $f_{\mathcal{D}|\boldsymbol{x}}$ is the likelihood function that describes the probability of observing data given the parameter estimates; $f_{\boldsymbol{x}}$ is the prior distribution of parameters; and $f_{\mathcal{D}}$ is the so-called evidence or marginal likelihood that ensures the posterior distribution integrates to 1.

As Eq. (2.18) implies, Bayesian inference provides an explicit representation of the uncertainty in the parameter estimates via characterizing the full posterior distribution of unknown parameters $\boldsymbol{x}$. The prior distribution of parameters and the likelihood function must be specified to solve Eq. (2.18). Here, we used the same uniform distributions as those used to construct the PCK surrogate models to represent the prior distributions, although these can be different. The likelihood function is specified by the observation noise model, which is assumed to be zero-mean Gaussian with state-dependent variance in this work. We use a particle filtering method, namely sequential Monte Carlo (SMC) [147], to approximately solve the Bayesian inference problem by iteratively updating the posterior $f_{\boldsymbol{x}|\mathcal{D}}$ at every

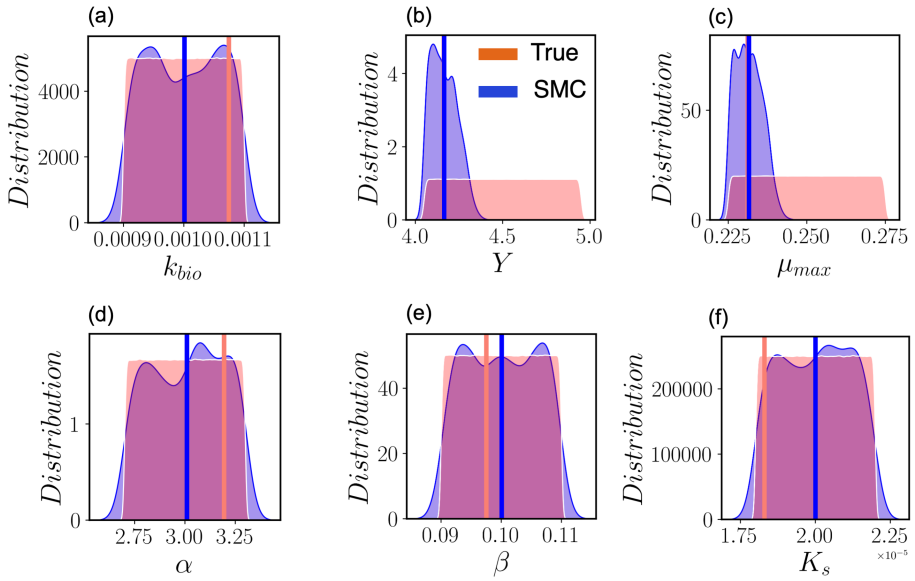

Figure 2.10: Bayesian inference of unknown parameters of the computational model of the microbial electrosynthesis bioreactor. The parameters are estimated via sequential Monte Carlo using 20,000 particles. Red and blue distributions represent the prior and posterior distributions of the unknown model parameters at time 3.5 days, respectively. The red vertical lines correspond to the true parameters, while the blue vertical lines are the estimated posterior mean value of parameters.

time instant that system observations become available; see [154] for further details. Notice that parameter estimation via Bayesian inference methods such as SMC relies on accurate construction of the probability distributions in Eq. (2.18). As described in Section 2.4, the data-driven flow-map models enable efficient sample-based approximation of the distributions using a very large number of samples, which otherwise could be impractical using an expensive computational model.

Figure 2.10 shows the posterior distribution of the parameters $\boldsymbol{x}$ at $t = 3.5$ days estimated via SMC using the dataset $\mathcal{D}$, as specified above. The posterior distributions are approximated using 20,000 particles. Note that the posterior distribution ranges seem to be larger than the prior in some cases, which is an artifact of the kernel density estimation (i.e., the selection of the bandwidth parameter) [59]. Figure 2.10 suggests that only the posterior distributions of parameters Y and μ_{max} have changed significantly with respect to their priors. It is also evident that the mean of the posterior distributions (blue vertical lines) for parameters Y and μ_{max} provides a fairly accurate estimate for the true, but unknown, parameter values (brown vertical lines). In particular, the true value and the posterior mean are indistinguishable, while the posteriors are much more narrow compared to priors as stated before. Nonetheless, the posterior distributions for the other parameters remain similar to their priors with little to no change, suggesting these parameters cannot be estimated using the available dataset $\mathcal{D}$. This can be attributed to the lack of information content of system observations $\mathcal{D}$ for inferring the unknown parameters; a deficiency that

can be addressed via optimal experiment design [235, 199]. We again note the flexibility of the flow-map models that would allow us to seamlessly add new observation points, should that become necessary for better parameter inference, without the need to construct new surrogate models for the states observed at new time points.

2.5 Conclusions

This paper presented a flow-map modeling approach based on polynomial chaos Kriging for the discovery of system dynamics from data. Data-driven flow-map models directly approximate the integration operator of differential equations that describe the state transitions of a dynamical system as a function of system state and input variables. We illustrated the usefulness of the proposed approach for learning mathematical descriptions of nonlinear dynamical systems and deriving dynamic surrogate models for fast uncertainty quantification applications. Our analyzes reveal that polynomial chaos Kriging-based flow-maps offer significant benefits in terms of data efficiency, as well as computational efficiency of data generation, for the discovery of nonlinear system dynamics and surrogate modeling.

Chapter 3

Performance-Oriented Model Learning via Bayesian Optimization

This chapter demonstrates the utility of data-driven modeling for decision-making and, in particular, in the context of model learning for closed-loop control-based applications. The closed-loop performance of model-based controllers, such as model predictive control, largely depends on the quality of their underlying model of system dynamics. Inspired by the notion of identification for control, this chapter presents a strategy for performance-oriented model adaptation for control. Starting from a data-driven control-relevant model based on prior knowledge about the system, the goal is to set up and solve a performance-oriented model adaptation problem towards optimizing the closed-loop system performance. To handle multiple conflicting objectives, multi-objective Bayesian optimization (MOBO) is used. The MOBO approach is demonstrated on a benchmark bioreactor case study. This chapter is adapted from [159].

3.1 Introduction

Model-based and optimization-based control approaches such as model predictive control (MPC) are widely used for control of constrained multivariable systems in a wide range of applications [229]. An important practical consideration in model-based control design is the model quality, which can greatly affect the closed-loop control performance, especially when the inherent robustness provided via measurement feedback is insufficient to mitigate model uncertainties. There has been a plethora of work on robust and stochastic model-based control and MPC to systematically account for model uncertainties and disturbances in the control problem formulation (e.g., [167, 169]). However, these strategies generally provide robustness to uncertainties at the expense of conservative control performance.

The traditional practice in model-based control design has relied on developing models independent of their control-oriented performance, i.e., how the predictive quality of models would influence the closed-loop control performance. An alternative view in handling system uncertainties in model-based control is to focus on the performance-oriented quality of models, rather than their general predictive quality. This point-of-view inspired much of the developments in the areas of *closed-loop identification* [292, 78] and *identification for*

control (I4C) [88]. The notion of I4C has been extensively explored for designing fixed-order controllers [87] and, more recently, for data-driven model learning and adaptation for MPC [14, 210, 262]. The fundamental idea of I4C is that the model that provides the best closed-loop performance may not be the one yielding the smallest prediction errors. This notion suggests that, for control applications, data-driven models must be identified or adapted by optimizing for their control-oriented predictive quality, which can be quantified in terms of pre-selected closed-loop performance measures of interest.

This paper focuses on performance-oriented model learning for model-based control applications. In recent years, there has been significant work on augmenting physics-based or data-driven models with a learning-based model such as deep neural networks or Gaussian process regression to capture unmodeled system dynamics (e.g., [89, 316, 13, 47, 181, 320, 211]). This modeling paradigm has been shown to be particularly useful in the context of learning-based control to mitigate plant-model mismatch towards safe and high-performance operation of complex technical systems (e.g., see [105, 168, 106, 26] and the references therein). To this end, this paper presents a systematic approach for performance-oriented model adaptation for control problems with multiple, possibly conflicting objectives. We consider a composite model structure consisting of a prior system model (physics-based or data-driven) and a data-driven model whose parameters can be efficiently adapted in a performance-oriented manner. Since in general closed-form expressions do not exist for closed-loop control performance measures, derivative-free optimization is used for performance-oriented model adaptation. In particular, we use Bayesian optimization (BO) [251], which has emerged as a powerful derivative-free method for optimizing black-box functions.

Owing to its ability to directly handle expensive-to-evaluate and noisy functions, BO has been employed in various applications, including hyperparameter selection for machine learning algorithms [258, 27] and automated strategies for controller tuning [18, 14, 183, 76, 124, 150, 200, 262, 245, 197]. However, the proposed BO approaches to model learning and controller auto-tuning rely on a single-objective formulation that is inadequate when the control design problem entails multiple and often conflicting control objectives [10, 232]. Thus, this paper proposes a multi-objective BO (MOBO) approach to model learning by leveraging the MOBO algorithm in [58]. The key notion of MOBO is to sample a probabilistic surrogate model of the multiple performance objectives, learned from closed-loop data, in an informed manner using a computationally efficient expected hypervolume improvement acquisition function [58, 11], which aims to discover Pareto optimal solutions of the multi-objective optimization problem.

We illustrate the MOBO approach to performance-oriented model learning on a benchmark bioreactor case study that has been widely used in MPC and input design studies (e.g., [103, 203]). Our simulation results demonstrate that the proposed model adaptation procedure can effectively learn the plant-model mismatch within fewer process runs, as compared to closed-loop identification, yielding a control-oriented model that provides satisfactory closed-loop performance with respect to multiple closed-loop performance measures of conflicting nature. We note that the proposed MOBO approach is readily generalizable to controller auto-tuning under process uncertainty. The paper is organized as follows. Section 3.2 presents the general formulation of the performance-oriented model learning problem. Section 3.3 describes the MOBO algorithm. Section 3.4 presents the composite model structure considered in this work, along with a discussion on adaption of the composite model

within a black-box optimization framework. Section 3.5 demonstrates the proposed approach on the benchmark bioreactor case study and compares its performance to closed-loop identification.

3.2 Problem Statement

We aim to control the discrete-time, uncertain nonlinear system

$$x^+ = f(x, u, \mathrm{w}), \tag{3.1}$$

where $x \in \mathbb{R}^{n_x}$ is the current state, x^+ is the successor state, $u \in \mathbb{R}^{n_u}$ is the control input, and $\mathrm{w} \in \mathbb{R}^{n_\mathrm{w}}$ represents unknown system disturbances. For the case of perfectly measured state of (3.1), we consider a model-based control law $\kappa(x;\theta)$, such that $\kappa : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_u}$ maps the state to the control input while parametrized by the adaptable model parameters $\theta \in \Theta$ that lie in a bounded set $\Theta \subseteq \mathbb{R}^{n_\theta}$.[1] No assumptions are imposed on the complexity or structure of the time-invariant control law $\kappa(\cdot)$, which can be a non-convex or implicitly-defined function of x and θ. By substituting a given controller $u = \kappa(x;\theta)$ in (3.1), we arrive at the closed-loop system

$$x_{k+1} = f(x_k, \kappa(x_k;\theta), \mathrm{w}_k), \tag{3.2}$$

where x_k and w_k are, respectively, the state and disturbance at discrete time step $k \in \mathbb{N}$. We note that every single trajectory of the closed-loop system (3.2) over a finite number of time steps T is fully governed by the choice of the model parameters θ and system uncertainties $\mathbf{w} = \{x_0, \mathrm{w}_0, \mathrm{w}_1, \cdots, \mathrm{w}_{T-1}\}$, where x_0 denotes the initial system state. We assume the uncertainties $\mathbf{w} \in \mathcal{W}$ follow a probability distribution $\mathrm{P}_\mathcal{W}$ defined over a bounded support $\mathcal{W}$.

In this work, we look to determine the "optimal" model parameters $\theta^\star$ by solving the multi-objective optimization problem

$$\theta^\star = \mathrm{argmin}_{\theta\in\Theta}\{\mathbb{E}_\mathbf{w}[\phi_1(\theta,\mathbf{w})], \mathbb{E}_\mathbf{w}[\phi_2(\theta,\mathbf{w}), \ldots, \mathbb{E}_\mathbf{w}[\phi_M(\theta,\mathbf{w})]\}, \tag{3.3}$$

where $\{\phi_m(\cdot)\}_{m=1}^M : \Theta \times \mathcal{W} \to \mathbb{R}$ are M measurable functions that define the closed-loop performance objectives. The performance objectives in (3.3) are defined in terms of expectations $\mathbb{E}_\mathbf{w}[\phi_m(\theta,\mathbf{w})] = \int_\mathcal{W} \phi_m(\theta,\mathbf{w})\mathrm{P}_\mathcal{W}$, for any scalar function of the closed-loop trajectories (3.2).[2] The performance objectives are generally defined in terms of reference tracking error, nonlinear economic costs, and violation of quality or safety-critical constraints. Yet, closed-form expressions do not exist for these closed-loop performance measures, which makes the solution of the multi-objective optimization problem (3.3) particularly challenging.

The goal of this paper is to learn the parameters θ of a data-driven model that captures the plant-model mismatch in a performance-oriented manner. We assume that a prior model

[1] In general, the parameters θ can include various controller design and tuning parameters (e.g., see [192]). In this paper, however, we use θ to only denote model parameters that can be adapted.

[2] Alternatively, the optimization problem (3.3) can be formulated as a robust problem in which the performance measures $\{\phi_m(\cdot)\}_{m=1}^M$ are defined in terms of worst-case performance objective or constraint violations [192, 201].

(either physics-based or data-driven) of the system (3.1) is available. As such, a data-driven component, whose parameters θ can be adapted, augments this prior model. This modeling approach is particularly advantageous for solving the performance-oriented model learning problem (3.3) for the following reasons. First, it encodes the *a priori* available knowledge of the process, thus is more likely to yield improvements in the closed-loop performance compared to initiating the model learning procedure randomly with no prior knowledge of the system. This can significantly enhance the performance-oriented learning of θ, especially when experimentation budget for model learning is limited. Second, adapting θ in a region informed by prior knowledge can translate to changes in parameters such that model predictions still retain their physical meaning, significantly aiding in constraint satisfaction. Third, particularly when a physics-based model is available *a priori*, the physics embedded in the predictive model can aid achieving meaningful predictions, which is not guaranteed with a data-driven model that operates outside the identification range. In the next section, we review the main concepts of the MOBO algorithm used for solving the multi-objective optimization problem problem (3.3).

3.3 Multi-Objective Bayesian Optimization

In general, closed-form expressions do not exist for the M closed-loop performance objectives $\{\phi_m(\theta)\}_{m=1}^{M}$ in the multi-objective optimization problem (3.3). Here, we resort to multi-objective Bayesian optimization for solving the performance-oriented model learning problem. Let us define a vector-valued objective function as $\boldsymbol{h}(\theta) : \mathbb{R}^{n_\theta} \to \mathbb{R}^M$ where each entry of $\boldsymbol{h}(\theta)$ corresponds to a closed-loop performance objective in (3.3), i.e., $h_m(\theta) = \mathbb{E}_{\mathbf{w}}\left[\phi_m(\theta, \mathbf{w})\right]$, given the real-valued decision variables $\theta \in \Theta$. We use sample average approximation (SAA) [127] to approximate the expectation of each performance objective as

$$h_m(\theta) = \mathbb{E}_{\mathbf{w}}\left[\phi_m(\theta, \mathbf{w})\right] \approx \frac{1}{N_{\mathrm{w}}} \sum_{j=1}^{N_w} \phi_m(\theta, \mathbf{w}_j), \tag{3.4}$$

where N_{w} is the total number of realizations of disturbance $\mathbf{w}$ for a given θ. As such, the sample-average approximations of (3.4) yields noisy observations of the closed-loop performance objectives

$$\psi_m(\theta) = h_m(\theta) + \epsilon_m, \tag{3.5}$$

where ϵ_m represents the observation noise for the mth performance objective. In the remainder, we use $\boldsymbol{\psi}(\theta)$ to denote the measured performance objectives for a given θ.

The fundamental idea of BO is that the objective function(s) may have an unknown mathematical form, but can be queried at any point in the design space [251]. This is particularly important since it avoids the necessity of having explicit expressions or approximations of derivatives for solving the optimization problem (3.3). BO takes a sequentially querying approach to update our belief over possible realizations of the performance objectives and, subsequently, the location of the global optimum of the decision variables; or the true Pareto front in MOBO, as described below. There are two key elements in BO. The first one is a probabilistic surrogate model trained to approximate the objective function(s). Gaussian processes regression [227] is the most common surrogate modeling approach in BO, although

other options, e.g., tree Parzen estimators [322], have also been used. The second element is the so-called *acquisition function*, which is optimized to determine the next query point in the design space that is most promising to be global optimum. There are various acquisition functions; the most widely used ones are the upper or lower confidence bound as well as the expected improvement [80]. The core idea of the acquisition function is to use the uncertainty information provided of the probabilistic surrogate of the objective function(s) to systematically tradeoff exploration and exploitation of the design space, so that the "optimal" combination of decision variables can be established within a given number of runs.

In this work, Gaussian process models are trained for the M performance objectives using the closed-loop data $\{(\theta^i, \boldsymbol{\psi}_m^i)\}_{i=1}^R,\ \forall m = 1, \ldots, M$, where the index R corresponds to the R^{th} closed-loop control run. Since there is typically no single optimizer in multi-objective optimization, the goal of MOBO is to discover a set of Pareto optimal points that define the Pareto front. We denote the finite set of Pareto optimal points by $\mathcal{P}$. A solution $\boldsymbol{h}(\theta)$ is said to *Pareto dominate* $\boldsymbol{h}(\theta')$ if $h_i(\theta) \geq h_i(\theta'),\ \forall\, m = 1, \ldots, M$, while $\exists\, m' \in \{1, \ldots, M\} : h_{m'}(\theta) > h_{m'}(\theta')$.[3] Here, we use the MOBO algorithm proposed in [58] to discover the Pareto dominate solutions of the multi-objective optimization problem (3.3). The MOBO algorithm uses a computationally efficient expected hypervolume improvement (EHVI) acquisition function [58, 11], which is essentially the extension of the commonly used expected improvement acquisition function in single-objective BO [70]. Since the multiple objectives form hyperplanes in the M-dimensional space in which they reside, EHVI relies on the hypervolume indicator [98] of a finite Pareto set $\mathcal{P}$, given some reference point θ_r. The hypervolume indicator is defined as $\mathcal{HV}(\mathcal{P}, \theta_r) = \lambda_M(\bigcup_{i=1}^{|\mathcal{P}|} \mathcal{R}[\theta_r, \boldsymbol{\theta_i}])$, where $\mathcal{R}[\theta_r, \boldsymbol{\theta_i}]$ is the hyper-rectangle bounded by the vertices $[\theta_r, \boldsymbol{\theta_i}]$ and λ_M is the M-dimensional Lebesgue measure of the space dominated by $\mathcal{P}$ and bounded from below by θ_r. Thus, given a set of new observed points $\mathcal{Z}$, we are interested in the improvement of the hypervolume, denoted by $\mathcal{HVI}$, stemming from these observations. Accordingly, the EHVI acquision function, denoted by $\alpha_{EHVI}(\theta_c)$, is defined in terms of $\mathcal{HVI}$ as

$$\mathcal{HVI}(\mathcal{Z}) = \mathcal{HV}(\mathcal{P} \cup \mathcal{Z}, \theta_r) - \mathcal{HV}(\mathcal{P}, \theta_r) \tag{3.6a}$$

$$\alpha_{EHVI}(\theta_c) = \mathbb{E}\left[\mathcal{HVI}(\boldsymbol{h}(\theta_c))\right]. \tag{3.6b}$$

Intuitively, the EHVI acquisition function seeks to determine the decision variable θ_c that, when observed, maximize the hypervolume covered by the corresponding Pareto front, given a reference point. The $\mathcal{HVI}$ can be calculated by partitioning the non-dominated space into disjoint hyper-rectangles [58]. Note that the acquistion function can seek more than one point per iteration, i.e., q points can be queried. It is worth mentioning that since the performance objectives $\boldsymbol{h}(\theta)$ are probabilistically approximated by Gaussian process models, the improvement of the hypervolume ($\mathcal{HVI}$) in the EHVI acquisition function should be marginalized over the probability distribution of possible objectives under the normal distribution. Therefore, the EHVI acquisition function for q points is given by

$$\alpha_{qEHVI} = \mathbb{E}[\mathcal{HVI}(\{\mathbf{h}(\theta_i)\}_{i=1}^{i=q})] = \int_{\mathbb{R}^M} \mathcal{HVI}(\{\mathbf{h}(\theta_i)\}_{i=1}^{i=q}) d\boldsymbol{h}, \tag{3.7}$$

[3] This is the case for a maximization problem while the inequality should be reversed for minimization problems.

where the subscript qEHVI denotes that the acquisition is based on the EHVI after observing q points. For the case of $q > 1$, the multivariate integral in (3.7) does not admit an analytical solution. Thus, it is approximated via Monte Carlo sample-based integration.

3.4 Composite Model Structure for Model Adaptation

In this section, we present a composite model structure that is particularly suited for performance-oriented model learning. The fundamental assumption is that a prior model of the system (3.1) exists, which can be either physics-based or data-driven. We represent the prior model by the discrete-time description

$$\widetilde{s}_{k+1} = \widetilde{f}(\widetilde{s}_k, u_k), \tag{3.8}$$

where $\widetilde{s} \in \mathbb{R}^{n_s}$ denotes the predicted state by the prior model and the model equations $\widetilde{f}$ can take any linear or nonlinear functional form. In effect, the presented model learning procedure relies on a prior model that predicts the future state as a function of the current input and state.

The prior model (3.8) is augmented with a data-driven model whose parameters θ can be learned in a performance-oriented manner using the MOBO approach described in Section 3.3. This data-driven model essentially represents the mismatch between the prior model (3.8) and the system (3.1), which can arise due to uncertainty in parameters or structure of model equations in (3.8). A commonly used approach in learning-based control to represent state- and input-dependent plant-model mismatch is via Gaussian process regression (e.g., [105, 27]), although other data-driven methods can also be utilized. Here, we consider a data-driven mapping $\mathcal{F}_\theta : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ parameterized by θ, which is an implicit function of state and control input, to represent the uncertainty of the prior model (3.8). Accordingly, the model used to design the control law $\kappa(x;\theta)$ (see Section 3.2) takes a general composite functional form as

$$s_c = (\mathcal{F}_\theta \circ \widetilde{s}_{k+1}) \tag{3.9a}$$

$$s_{k+1} = g(s_c), \tag{3.9b}$$

where $\circ$ denotes the composition operator, s_c is the result of the composition, and g is a general function of this composition that yields the successor state s_{k+1} of the composite model used for control design. Notice the difference between the successor state $\widetilde{s}_{k+1}$ predicted by the prior model (3.8) and the successor state s_{k+1} predicted by the composite model (3.9).

A popular choice for the data-driven mapping $\mathcal{F}_\theta$ is artificial neural networks due to their universal approximation capability [95], while offering sufficient flexibility for model adaptation purposes. Note that Bayesian neural networks with confidence bounds on their predictions could also be useful [284]. Here, we use a feedforward neural network to represent the mapping $\mathcal{F}_\theta$, which is called a "performance adapted network" (PAN) in the remainder of this paper.

The PAN constitutes a nonlinear, input-output mapping, where information is propagated from the input layer to the output layer via L hidden layers that contain units known as nodes [95]. Given an input z, the output of the PAN is defined as

$$\mathcal{F}(z;\theta,\mathcal{C}) = \boldsymbol{W}_{L+1} \circ (\sigma_L \circ \boldsymbol{W}_L) \circ \cdots \circ (\sigma_1 \circ \boldsymbol{W}_1)(z), \tag{3.10}$$

where $\boldsymbol{W}_i$ is the weight and bias matrix between the ith and $(i+1)$th layers of the network and σ_i is the activation function. In (3.10), $\theta = \{\boldsymbol{W}_i\}_{i=1}^{L+1}$ are the PAN parameters that must be adapted via MOBO, whereas $\mathcal{C}$ are the preselected hyperparameters [72].

The construction of PAN must ensure that the predicted state s by the composite model (3.9) is physically meaningful. Furthermore, we generally desire a PAN architecture that has (i) a minimum number of parameters that must be adapted in a performance-oriented manner; thus, keeping the MOBO problem tractable; and (ii) a form that is explainable such that we can establish meaningful bounds for the parameters θ. Bounding the space of decision variables θ in the MOBO problem is crucial for both the convergence of the optimization algorithm, as well as for ensuring that the predicted state is physically meaningful. Here, we consider an implicit multiplicative form for the composite model (3.9), which takes the form

$$s_{k+1} = \left(\mathcal{F}_\theta \circ \widetilde{f}(\widetilde{s}, u)\right) \widetilde{f}(\widetilde{s}, u), \tag{3.11}$$

where the term $\left(\mathcal{F}_\theta \circ \widetilde{f}(\widetilde{s}, u)\right)$ is in effect a muiltiplicative/scaling term to the predictions of the prior model. This scaling property can be realized by using a sigmoid activation function in the last layer of the PAN

$$\sigma(\zeta_j) = \omega_0 \frac{1}{1+e^{-\zeta}} \in [0, \omega_0] \quad \zeta \in \mathbb{R}, \tag{3.12}$$

where ζ is the input to the activation function, i.e., the output of the last hidden layer, and ω_0 is a factor that rescales the sigmoid as $[0,1] \to [0,\omega_0]$. We note that the choice of activation functions and structure for the hidden layers of PAN is rather arbitrary.

Remark 1: An additive model with any choice of activation function can also be used for capturing the plant-model mismatch. In this case, the composite model (3.9) would take the form $s_{k+1} = \widetilde{f}(\widetilde{s}, u) + \left(\mathcal{F} \circ \widetilde{f}(\widetilde{s}, u)\right)$ such that $\mathcal{F}$ can be an explicit function of the current state and input. This is a commonly used model structure in learning-based MPC [105, 27].

The iterative performance-oriented model adaptation approach via MOBO is depicted in Figure 3.1. This figure shows how a prior model of the system is augmented with the PAN, i.e., (3.9), whose parameters θ are adapted after every process run using closed-loop data. The iterative model learning procedure suggests new model parameters for the next run. A potential challenge in the proposed approach can arise from a possibly large number of parameters θ when a general PAN architecture with multiple layers/nodes and arbitrary activation functions is used. Even a shallow, scaled PAN may suffer from the same issue if the model state, $\widetilde{s}$, has an excessively high dimension. One option to overcome this challenge is to take advantage of high-dimensional BO (HDBO) algorithms that rely on low-dimensional embeddings of the design space [301, 140]. However, HDBO is most useful when the intrinsic dimensionality of the black-box optimization problem is small, which is often not known *a priori*. Alternatively, here we take a heuristic approach, based on global sensitivity analysis (GSA) [260, 242, 154], to determine a subset of PAN parameters θ to which the closed-loop performance objectives are most sensitive. This is further discussed in Section 3.5.

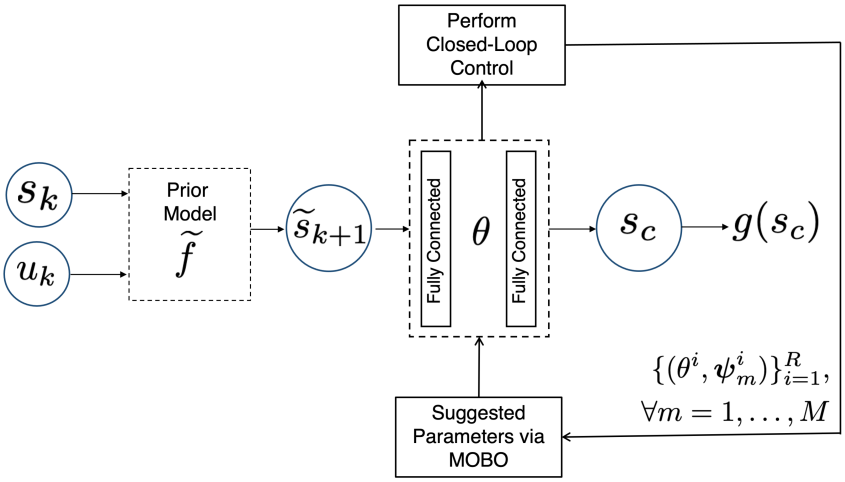

Figure 3.1: The iterative performance-oriented model learning approach via MOBO. A prior model of the system is augmented with a so-called performance adapted network (PAN) with parameters θ, resulting in a composite model structure $g(s_c)$ used for model-based control. MOBO uses noisy observations of M closed-loop performance objectives, $\{(\theta^i, \boldsymbol{\psi}_m^i)\}_{i=1}^R$, $\forall m = 1, \ldots, M$, collected till process run R to suggest new model parameters θ for the next process run.

3.5 A Bioreactor Case Study

We demonstrate the proposed performance-oriented model learning approach on a benchmark bioreactor case study with economic control objectives. Ensuring the optimal performance of bioreactors for short operational cycles is crucial in the context of biologically-driven mission design for deep space exploration.

Systems engineering of this type, where controllers and their models can be automatically tuned, is of paramount importance for the automation of processes with limited human-interaction, an objective of the Center for the Utilization of Biological Engineering in Space (CUBES) [20].

The system used for this case study involves the operation of a continuous fermenter [4], which has been used in control [103] and experimental design for active fault detection [203] studies. The reactor has a constant volume and is assumed to be isothermal and well-mixed. The process state variables consist of the concentrations of biomass X (g/L), substrate S (g/L), and product P (g/L). The manipulated variables are the dilution rate D (hr^{-1}) as well as the feed concentration of the substrate S_f (g/L). The objective of the controller is to strike a balance between maximizing the amount of product at the reactor outlet, while maintaining a high biomass concentration. The transient mass balances that describe the

process (i.e., the plant model) are given by

$$\dot{X} = -DX + \mu X \tag{3.13}$$

$$\dot{S} = D(S_f - S) - \frac{1}{Y_{X/S}}\mu X \tag{3.14}$$

$$\dot{P} = -DP + (\alpha\mu + \beta)X, \tag{3.15}$$

where $Y_{X/S}$ is the cell-mass yield, μ is the specific growth rate, while α and β are related to the product yield. The specific growth rate is assumed to follow a modified Monod kinetics law which takes into account both substrate and product inhibition

$$\mu = \frac{\left(1 - \frac{P}{P_m}\right) S\mu_{\max}}{K_m + S + \frac{S^2}{K_i}}, \tag{3.16}$$

where μ_{max} is the maximum growth rate, K_m is the Monod affinity constant, P_m is a maximum product concentration, denoting inhibition from P, and K_i is the substrate-inhibition constant. The values of the parameters used in for the plant model are given in Table 3.1. In addition, we induce some degree of plant-model mismatch by changing the parameters of the plant model, obtaining a physics-based simulation model the parameters of which are also reported in Table 3.1.

Table 3.1: Bioreactor model parameters

Parameters	**Plant Model**	**Physics-based Model**
$Y_{X/S}$ (g/g)	0.25	0.1
α (g/g)	2.5	1.5
β (hr^{-1})	0.85	1
$\mu_{\max}$ (hr^{-1})	1	2
K_m	1.25	1
K_i (g/L)	20	10
P_m (g/L)	5.75	10

Pefrormance Measures and MPC Problem Formulation

The bioreactor is assumed to operate for a total time of $t = 5\,(\text{hr})$ with a control sampling time of $\delta t = 0.1$ hr. Thus, the duration of the entire process is consists of $k = 0, \ldots, T$ time steps where $T = 50$. The two performance metrics of interest are: (i) the total amount of product that exits the reactor over the entire process horizon, denoted by and $\phi_1(\theta, \mathbf{w})$; and (ii) the final concentration of the biomass $\phi_2(\theta, \mathbf{w})$, which are given by

$$\phi_1(\theta, \mathbf{w}) = V \int_0^{t_f} P(\tau, \theta, \mathbf{w}) D(\tau, \theta, \mathbf{w}) d\tau \approx \sum_{k=0}^{T-1} V P_k(\theta, \mathbf{w}) D_k(\theta, \mathbf{w}) \delta t \tag{3.17a}$$

$$\phi_2(\theta, \mathbf{w}) = X_T(\mathbf{w}), \tag{3.17b}$$

where $V = 10$ L is the reactor volume. At each sampling step $k \in \mathbb{N}_0^T$, the controller solves the following finite-horizon optimal control problem (OCP)

$$V(x_0) = \min_{x_{\cdot|k}, u_{\cdot|k}} \sum_{i=0}^{N-1} l(x_{i|k}, u_{i|k}) + \alpha_N l_N(x_{N|k}) \tag{3.18a}$$

$$s.t. \quad x_{i|k} = \mathcal{N}(x_{i|k}, u_{i|k}; \theta), \quad \forall i \in \mathbb{N}_0^{N-1} \tag{3.18b}$$

$$(x_{i|k}, u_{i|k}) \in \mathbb{X} \times \mathbb{U}, \quad \forall i \in \mathbb{N}_0^{N} \tag{3.18c}$$

$$x_{0|k} = x_0, \tag{3.18d}$$

where $N = 10$ is the prediction horizon, $l(\cdot)$ denotes the stage cost, $l_N(\cdot)$ the terminal cost, while a_N is a weight for the terminal cost. The stage cost has the form of ϕ_1, however defined over the prediction horizon instead the process horizon, and the terminal cost is same as ϕ_2, referring to the end of the prediction horizon (and not the end of the entire process). Thus, having those two types of costs in the objective function aims to represent a scalar version of the conflicting objectives problem. Nevertheless, the main idea is that the MPC problem does not need to be treated as a MO dynamic optimization problem at its design phase. Instead, the dynamics learning problem should be able to handle this contradicting nature of costs. As for the disturbances that affect the BO learning procedure, we assume uncertainty in the initial conditions that are drawn from a uniform distribution, $\mathbf{w}_j \sim \mathcal{U}(x_0)$, with j denoting the index of the realization. The SAA approximation of the expected value is the average of 2 runs per design variable queried. A nominal uncertainty realization that we used in subsequent analysis is $\mathbf{w}_0 = [0.4, 0.2, 0.1]$. We solve the MPC problem by interfacing IPOPT with the CasADi library [299, 7].

Modeling and Sensitivity Analysis

We assume the case where the HF model that is available is not efficient for on-line MPC. In this case, an initial neural network (prior model) is trained using input-output data from the physics-based model. To approximate the system dynamics, we resort to a residual Neural Network (rNN) [220]. The neural network consists of $L = 3$ hidden layers and $L_n = 20$ nodes each. This leads to a total of 1023 trainable parameters. To obtain training data, we use 100 simulations which yield 5000 data samples. A train-test splitting of the data is perfromed and 80% of them are reserved for training. The neural network is constructed using `keras`. The model is trained for 1000 epochs with a batch-size of 10, using a learning rate of 5×10^{-3} and the `Adagrad` optimizer (whose default learning rate is 10^{-3} in `keras`) [48]. The achieved mean absolute error loss (MAE), based on the validation set, is approximately $\epsilon_p = 2.8 \times 10^{-3}$, thus the HF model and its surrogate have negligible discrepancies. Once an initial rNN is established, we augment the model with a PAN, based on the idea of transfer learning [111, 208]. The rNN is "frozen", i.e., its parameters remain fixed throughout the learning process, while a PAN is added to its last layer. In particular, we use a shallow, single layer PAN with sigmoid activation as described in Sec.3.4. Therefore, the input to the sigmoid activation is a linear transformation of the prior model outputs

$$\zeta_j = \omega_{j,i} \cdot \widetilde{s}_{k+1}^i + b_j, \tag{3.19}$$

where $\omega_{j,i}$ is the weight connecting the i^{th} node of prior model outputs to the j^{th} node of PAN layer and b_j is the corresponding bias. In this case, a heuristic initialization of the PAN parameters can be performed. Note that the input to the sigmoid function is a linear transformation of the rNN outputs, i.e., physics-based predictions. Then, for $\omega_{i,j} = 0$ and $b_i > 5$, the sigmoid is $\sigma(\zeta) \approx 1$, due to the exponential term getting very small values. Moreover, an order of magnitude analysis of the typical values of the states (based on the HF model) reveals that for large and small enough $\omega_{j,i}$ and b_j, the input saturates, i.e., the sigmoid is very close to 1 or 0, respectively. Therefore, we set lower and upper bounds to the PAN parameters around the aforementioned values and perform a GSA on the performance metrics based on the HF model, using the nominal value for uncertainty, $\mathbf{w}_0$. All weights are bounded in [-0.5,2], the biases in [0,5], while the ω_0 factor in [0.65,1.35]. Figure 3.2

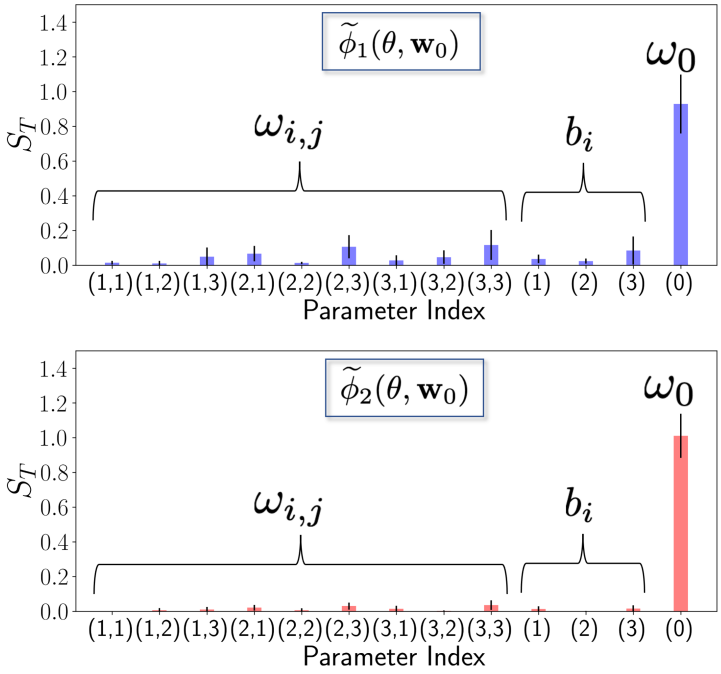

Figure 3.2: Global sensitivity analysis (GSA) results that demonstrate the influence of the shallow PAN parameters on the nominal performance metrics under a high-fidelity plant model. The horizontal axis denotes the index of the parameter while the vertical axis shows the total Sobol' sensitivity index

shows the results of the GSA for both performance metrics. In both cases, we observe that the most influential parameter is the scaling factor ω_0, followed by $\omega_{2,3}$ and $\omega_{3,3}$ as well as b_3. Therefore, we select these 4 parameters for the dynamics learning. For the rest of the parameters we fix their values based on the sample that maximizes $\widetilde{\phi}_1(\theta; \mathbf{w}_0)$, which is the primary objective.

Single-Objective Model Learning

For the first case study, we consider the problem of single-objective BO (SOBO), attempting to maximize $h_1(\theta)$, i.e., the sample-average productivity of the reactor after a fixed operational time. To this end, we set $\alpha_N = 0$ for the MPC problem as we are not interested in the concentration of biomass at the end of the process. For the initialization and bounding of the tuning parameters, we rely on the GSA results as described previously, except for ω_0 which is set to 1, as we want the performance learning procedure to initiate from a point close to the nominal model's solution. Alternatively, one can use the baseline set of parameters (zero weights, high biases, scaling factor equal to 1) as an initial guess (which would result into a PAN that is equal to 1).

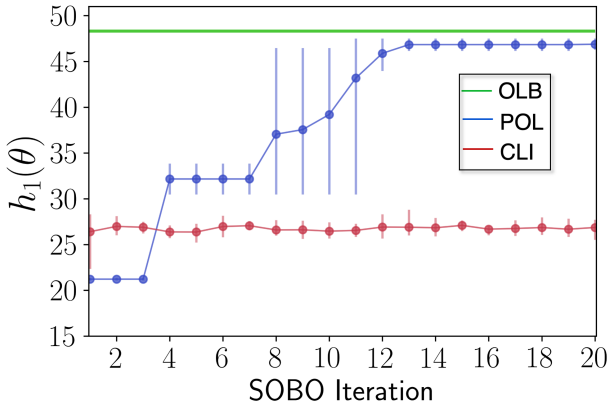

Figure 3.3: Convergence plot of SOBO. The blue points represent the average performance (vertical axis) across the BO iterations (horizontal axis) in the case of Performance-Oriented Learning (POL). Red points represent the average performance when the neural network is retrained using new input-output data from the previous run of the process, i.e., closed-loop identification (CLI). The green line represents the open-loop, theoretically best (OLB), solution under a nominal realization of uncertainty. The vertical lines at each point represent the range of performances observed during each replicate of the BO algorithm.

The results comparing the performance-oriented learning strategy (POL) and the closed loop identification (CLI) approach are shown in Figure 3.3. We replicate the SOBO algorithm 5 times to investigate the effect of uncertainty realizations, by fixing the BO algorithm random seed, but allowing for different realizations of process uncertainty each time.

The results indicate that a performance-oriented strategy is able to significantly improve the closed-loop performance when a physics-based/data-driven informed initial guess is provided. The optimal performance that is discovered in each replicate of the SOBO algorithm, denoted by the blue line and the corresponding vertical lines (performance range), is consistently better than the performance observed when operational data are used to retrain the initial rNN, denoted by the red line. For the latter, we use the latest batch input-output data from the plant and update the last hidden layer parameters. Note that the first 5 iterations of the SOBO algorithm correspond to a random search, since they are the points used to train the GP surrogate of the objective. The green line represents an upper limit to the

achievable performance (under the nominal disturbance realization), since it is calculated via the theoretically best solution of the open-loop problem (OLB) over the entire process horizon, without plant-model mismatch. It is observed that SOBO attains a performance very close to the theoretical maximum after a number of iterations.

Furthermore, it is also interesting to examine the state and optimal input profiles that are discovered via closed loop retraining and by applying SOBO after each set of closed-loop runs, compared to the open loop prediction. For clarity we show only the profiles of the states and inputs that appear in the MPC cost and SOBO learning objective. The plots correspond to the nominal realization of uncertainty.

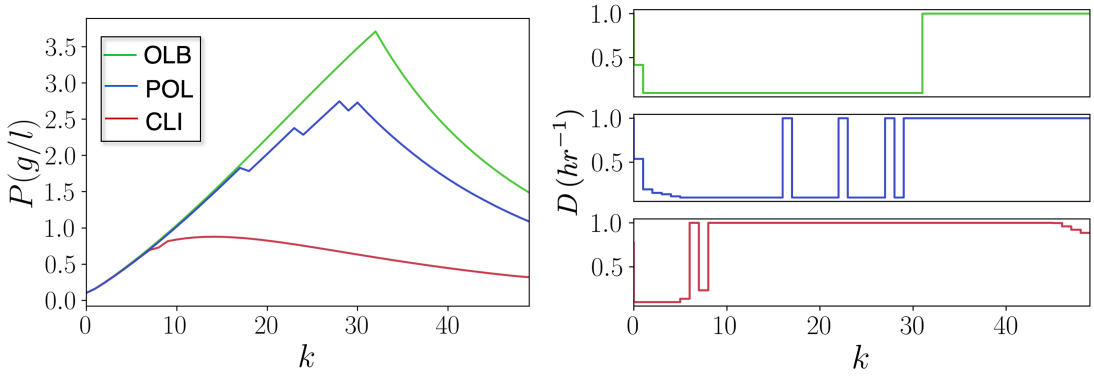

Figure 3.4: Closed-loop profiles under the nominal realization of uncertainties. The horizontal axis denotes the current stage/time step and the vertical axis shows (i) the concentration of product in the left plot, (ii) the dilution rate in the right plot. Green lines correspond to open-loop theoretically best solution (OLB), blue lines to the Performance-Oriented Learning (POL) in the SOBO case and red lines correspond the closed-loop identification case (CLI).

The MPC solution profiles are depicted in Figure 3.4 along with the open-loop solution. As for the open-loop solution, given that it has full and perfect preview of the dynamics, we observe that it predicts a simple switching-regime for the dilution rate, which is initially kept at low levels in order to promote growth, but then increases to the upper bound so that we are able to extract as much as possible product. For the performance-oriented learned dynamics case, the dilution rate is qualitatively similar at the initial stages with the only difference being some larger dilution rates being used sporadically in the middle of the process. Subsequently, the state profiles of the BO and the open loop cases are very close. On the other hand, the CLI profiles, denoted by the red lines, are qualitatively different. Although the predictions of the model are more accurate, in a predictive plant-model mismatch sense, the myopic nature of MPC forces the system to operate with high dilution rates early on, without being able to anticipate that this leads to low product concentration in later stages.

Multi-Objective Model Learning

In this case, we modify the previous problem, as our goal is not just to discover the dynamical model that maximizes the amount of the available metabolic product, but we are also interested in solutions that retain an acceptable amount of biomass at the end of the process. For instance, this problem might arise in a scenario where our primary objective is the production of P, however, there is a need for biomass for downstream processing. These objectives are conflicting. In the SOBO case, we observed that the optimal strategy is to initially enforce low-dilution rates and operate at high dilution rates mid-process in order to get more quantity of P. Intuitively, however, one can reason that following the same strategy in the MOBO case would not be optimal for maximizing the biomass concentration, since increasing the dilution rate significantly can lead to near-washout conditions. For this particular problem, we utilize the terminal cost that was introduced earlier with $\alpha_N = 2$, leading to a scalar version of the MO problem within the MPC. The selection of this factor depends on the significance of maximizing this secondary performance metric, i.e., the value of biomass at the end of the process.

For the solution of the MOBO problem, we use the same bounds as in the SOBO case for the search space and use 10 initial random samples (RS) in order to construct the initial surrogates for the objectives. Moreover, we query and observe one point per MOBO iteration, hence $q = 1$. First, we need to examine the algorithm convergence which is expressed by the hypervolume indicator in the MO counterpart of the dynamics learning problem.

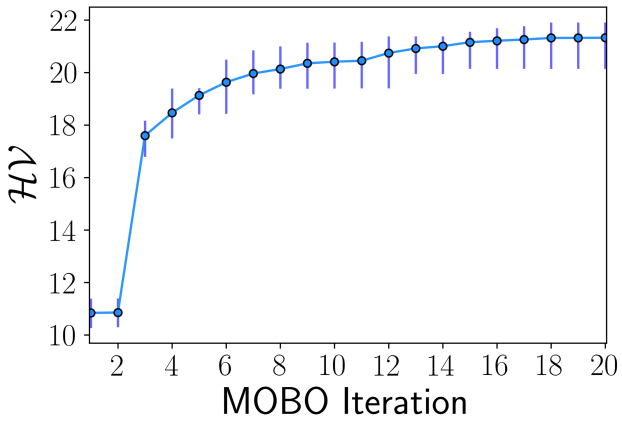

Figure 3.5: Hypervolume convergence plot for the MOBO problem. The horizontal axis denotes the number of iterations of the acquisition-based search of the MOBO algorithm. The vertical axis denotes the average hypervolume at each iteration. The vertical lines represent the range of the $\mathcal{HV}$ observed across replicates.

Figure 3.5 shows the growth of the hypervolume as the MOBO iterations proceed. The circles denote the average $\mathcal{HV}$ across replicates, while the vertical lines show the corresponding range. We observe that the algorithm reaches acceptable performance levels after approximately 10 queries. We should note, however, that the notion of convergence in the MOBO case should be interpreted differently than the SOBO case. In particular, although in SOBO more iterations would lead to a negligible reward after some iteration, in MOBO a

number of extra informed runs (via the acquisition function), lead to the discovery of more Pareto optimal points, which do not necessarily improve the $\mathcal{HV}$ much, but lie in the Pareto front, thus constitute important candidate dynamical models for operation purposes.

Having discussed the $\mathcal{HV}$ convergence plot, we now examine the optimality of the points discovered by the acquisition function. The fundamental result of MOBO is essentially the approximate Pareto front that has been established after the querying the search space using the acquisition function α_{qEHVI}.

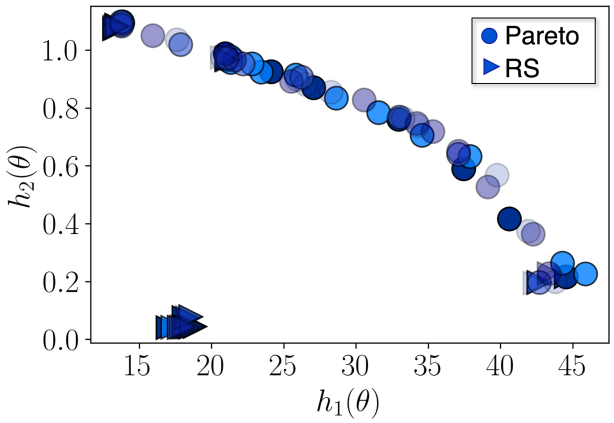

Figure 3.6: MOBO solutions for the two-objective problem. The triangles represent the initial points that are discovered using random samples (RS) used for training the GPs. The circles represent the points proposed by the acquistion function. Various replicates are represented with different faintness.

Figure 3.6 shows the obtained Pareto front for 5 replicates of the algorithm (with a fixed initial seed for the MOBO solver but different realization of the uncertainties in the plant). The replicates are denoted with varying shade and faintness. It is interesting to first observe the results of the RS performed in order to train the initial GP surrogates for the objectives, represented by the triangles. Most of those random observations are limited in a narrow region corresponding to both low productivity and concentration of biomass, with a few of them corresponding to the extreme cases (high product amount/low concentration or vice-versa). Nevertheless, the points discovered using MOBO seem to form a clear Pareto front, which is consistent among runs.

Remark 2: The consistency of the algorithm is affected by the levels of process noise and measurement noise. The effect of both of these noise sources can be mitigated by increasing the number of experimental realizations of the plant, which is not always feasible due to budget constraints. In simulation case studies, robust BO schemes can be used in order to find the design that maximizes the performance in a worst-case sense. In addition, it is interesting to inspect the closed-loop solution for a Pareto point that achieves high terminal concentration of biomass but low P production.

The results on Figure 3.7 show that, although the primary objective of the MPC is to achieve high productivity, MOBO has a learned a dynamical model that leads to a fundamentally different solution landscape. In particular, the switching-regime that we observed in the

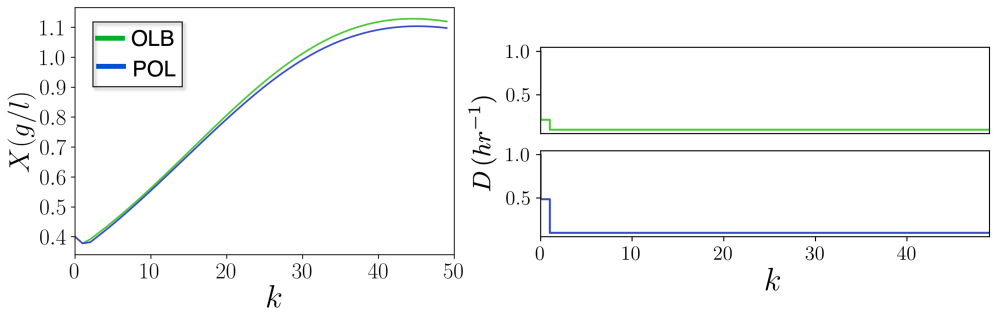

Figure 3.7: Closed-loop profiles under the nominal realization of uncertainties. The horizontal axis denotes the current stage/time step and the vertical axis shows (i) the concentration of biomass in the left plot, (ii) the dilution rate in the right plot. Green lines correspond to open-loop theoretically best solution (OLB) and blue lines to the Performance-Oriented Learning (POL) approach in the MOBO case, for a design that maximizes the final concentration of biomass.

SOBO case does not occur here, suppressing significantly the production of the metabolic product, which has a value of $\phi_1(\theta^*, \mathbf{w}_0) = 13.6(\text{g})$, while $\phi_2(\theta^*, \mathbf{w}_0) = 1.09(\text{g/l})$. For the numerical solution of the problem we used the BoTorch library [4]

3.6 Conclusions

This paper presented a performance-oriented model adaptation approach for model-based control of uncertain systems. The proposed approach relies on the notion of identification for control in which the underlying model of a controller is adapted over several process runs to reduce plant-model mismatch specifically towards maximizing preselected closed-loop performance measures. The model learning problem is formulated as a black-box, multi-objective optimization problem that is solved via Bayesian optimization. The effectiveness of the proposed approach was demonstrated for control-relevant model learning for MPC of a benchmark bioreactor problem for the cases of single- and multi-objective control. In the single-objective case, we observed that, given a fixed budget of process runs, the proposed approach gives control-oriented models that lead to a significant improvement in closed-loop performance compared to model identification using closed-loop data. This can be attributed to the use of a prior model in the proposed composite model structure that helps BO target a relevant exploration space; thus, converging to a better-performing model in terms of closed-loop performance in fewer process runs. In the multi-objective case, we observed that the informed exploration of the model parameter space is able to accommodate conflicting closed-loop performance objectives and lead to controllers that give rise to vastly different closed-loop dynamics. Our future work will focus on investigating constrained BO formulations for model learning and auto-tuning of controllers for safety-critical applications.

[4]The code used for the MOBO algorithm has been adapted by https://github.com/panos108/Multi-objective-Bayesian-Opt.git.

Chapter 4

Adversarially Robust Bayesian Optimization for Auto-Tuning of General Controllers Under Uncertainty

This chapter demonstrates the use Bayesian Optimization (BO) for controller autotuning in the presence of uncertain parameters in the performance function. An open challenge when applying BO to auto-tuning is how to effectively deal with uncertainties in the closed-loop system that cannot be attributed to a lumped, small-scale noise term. We address this challenge by developing an adversarially robust BO (ARBO) method that is particularly suited to auto-tuning problems with significant time-invariant uncertainties in an expensive system model used for closed-loop simulations. ARBO relies on a Gaussian process model that jointly describes the effect of the tuning parameters and uncertainties on the closed-loop performance. From this joint Gaussian process model, ARBO uses an alternating confidence-bound procedure to simultaneously select the next candidate tuning and uncertainty realizations, such that only one expensive closed-loop simulation is needed at each iteration. The advantages of ARBO are demonstrated on two case studies, including an illustrative problem and auto-tuning of a nonlinear model predictive controller using a benchmark bioreactor problem. This chapter is adapted from [197].

4.1 Introduction

Recent years have witnessed significant progress in the design and application of optimization- and learning-based controllers that can deal with multivariable dynamics, constraints, and uncertainties that appear in the system and/or the environment. However, the design of such advanced controllers hinges on the selection of several *tuning parameters* that may strongly affect closed-loop performance and constraint satisfaction. Additionally, these tuning parameters can come in a variety of different forms including continuous (e.g., weight parameters), discrete (e.g., logical switching conditions such as adaptive tuning), and categorical (e.g., type of numerical discretization scheme) representations, which implies their impact on performance can be highly nonlinear and non-convex. Therefore, in practice, these tuning parameters are usually selected via trial-and-error experimentation or heuristic-based

strategies that rely on expensive closed-loop simulations or experiments, which can become prohibitive when the effects of system uncertainties are accounted for [194].

To mitigate the expensive nature of tuning of advanced controllers, there has been an increasing interest in automatic calibration (aka *auto-tuning* [85, 285, 183, 200, 324]) of complex control structures to achieve desired closed-loop performance. To this end, data-driven optimization methods have been found to be particularly promising since auto-tuning can be interpreted as a black-box problem in which the objective function is expensive to evaluate, potentially non-convex and multi-modal, and whose derivatives either do not exist or cannot be determined. Bayesian optimization (BO) [258, 252] has emerged as a powerful approach for handling these types of black-box problems, even when the measured objective value is corrupted by noise. Several recent works have successfully demonstrated BO for model learning and auto-tuning of model predictive control (MPC) [200, 210, 150, 262] and other complex control structures [73, 124].

Standard BO approaches for auto-tuning rely on non-parametric Gaussian process (GP) models [227], constructed from closed-loop simulation or experimental data, to describe the impact of controller tuning parameters on the closed-loop performance measures; these GP models can be interpreted as probabilistic "surrogate models" for the performance measures of interest. Although GP models are able to account for the effect of system uncertainties (e.g., exogenous disturbances, measurement noise, and/or time-invariant uncertainties in process models used for closed-loop simulations) by optimizing an "effective noise" hyper-parameter, this representation can lead to poor predictions when uncertainties are relatively large. That is, the GP model yields such a large variance in predictions that the mean prediction is dominated by noise, suggesting the GP model is uninformative. In such cases, the BO procedure will become quite fragile and thus will lead to poor overall results. We addressed this challenge in our recent work by introducing an auto-tuning approach, referred to as probabilistically robust Bayesian optimization (PRBO), that provides a probabilistic robustness certificate at every iteration (i.e., every time a new set of tuning parameters is tested) [195]. The key idea in PRBO is to use sample-based estimates of the worst-case performance measures at each iteration. We show how many samples are required — *independent* of the number and probability distribution of the uncertainties — to ensure these worst-case estimates are not violated by other randomly sampled uncertainties within a prespecified probability level. However, since PRBO provides this certification at every iteration, it generally requires a fairly large number of closed-loop simulations/experiments to be performed in order to establish accurate estimates of the worst-case performance measures. This can limit the applicability of PRBO especially when expensive "high-fidelity" process models (or experiments) are utilized for generating closed-loop data.

In this paper, we present an alternative robust BO approach to PRBO that is well-suited for auto-tuning problems that rely on expensive closed-loop simulations with significant time-invariant uncertainties. This type of problem setting appears in a wide variety of applications that use complex process models and model predictive controllers (MPC), including advanced manufacturing and energy systems [67], among many other applications. As opposed to measuring an estimate of the worst-case performance directly (as done in PRBO), the proposed approach, referred to as *adversarially robust BO* (ARBO), looks to solve a problem that *simultaneously* captures the effect of the controller tuning parameters and system uncertainties on the closed-loop performance. In contrast to typical BO approaches where

a GP model is used to approximate the objective as a function of the decision variables (in this case controller tuning parameters), we utilize a GP to approximate the objective explicitly as a function of decision variables and uncertain parameters. In this way, we can directly use this *joint* GP model (where joint refers to the simultaneous consideration of the tuning and uncertain parameters) to predict the location of a minimax solution to the robust auto-tuning problem. We show, however, that using a naive mean-based GP approximation of the performance measure will yield overall poor tuning results, as it lacks the ability to tradeoff between exploration of unknown parts of the *design-uncertainty space* and exploitation of the current estimate of the best tuning parameters. Instead, the proposed ARBO method uses a GP confidence bound-based procedure suggested in [24] to realize a tradeoff between the exploration and exploitation of the design-uncertainty space. In this approach, we alternate between an optimistic prediction of the performance measure to select the next best set of tuning parameters and a pessimistic prediction of the performance measure to select the most likely worst-case uncertainty for the suggested best tuning parameters. By applying this two-step procedure, we only require one (expensive) closed-loop simulation at each iteration of ARBO, which is significantly less than alternatives such as PRBO. Building upon the theory in [24], we also discuss the rate of convergence of the ARBO method, and provide an explicit upper bound on the distance from the best suggested tuning parameters and the true minimax optimal solution, which decays to zero as the number of iterations increases.

We demonstrate the value of the proposed ARBO method on two case studies; an illustrative problem to highlight the key steps and advantages of ARBO and a challenging auto-tuning problem in which a highly nonlinear bioreactor with several unknown parameters is controlled using nonlinear MPC with multiple constraint backoffs that must be tuned. It is important to note that, although we focus on nominal MPC, the ARBO can robustly auto-tune any controller structure including those that incorporate recently proposed safety schemes such as the model-predictive safety (MPS) method [263, 5]. MPS solves several minimax optimization problems offline to identify potential worst-case uncertainty values that may occur online using a nested particle swam optimization (PSO) algorithm. The main disadvantage of PSO is that it requires a very large number of function evaluations, which prevents its usage in applications with very expensive objective functions such as that considered in this work.

4.2 Problem Statement

We are interested in the auto-tuning problem for a general class of controllers, i.e., we want to select the unknown tuning parameters such that we achieve the best possible closed-loop performance, while protecting against potentially adversarial effects of some "external" source of uncertainty. Let $\theta \in \mathbb{R}^{p_1}$ denote the vector of controller tuning parameters and $\delta \in \mathbb{R}^{p_2}$ denote the system (plant) uncertainty vector. Given some scalar measure of the closed-loop performance $f : \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \to \mathbb{R}$ whose structure is unknown, we formulate the auto-tuning problem as the following robust black-box optimization problem

$$\min_{\theta \in \Theta} \max_{\delta \in \Delta} f(\theta, \delta), \tag{4.1}$$

where $\Theta \subset \mathbb{R}^{p_1}$ and $\Delta \subset \mathbb{R}^{p_2}$ are the compact sets of possible tuning parameters and uncertainty realizations, respectively. The controller tuning parameters θ can represent any manipulable value including discrete structural choices (e.g., turning on/off a component) that are modeled with binary variables, as well as parametric choices that are modeled by continuous variables (e.g., increasing a weight value between lower and upper bounds). To account for the effects of uncertainty on controller tuning, we must quantify the impact of different realizations of δ on the performance measure f. Thus, throughout this work, we assume that a *high-fidelity simulator* of the process is available for simulating the effect of specific controller configurations and uncertainty realizations on the closed-loop performance measure f.[1] This allows f to be flexibly specified by the user in terms of any finite-time metric; some common examples include total operating cost or setpoint tracking error, average or maximum constraint violation, and end-of-batch product quality.

We aim to find the (approximate) global solution to the controller auto-tuning problem (4.1). The specific algorithm chosen to solve (4.1) will depend on its underlying characteristics. Thus, we assume that the following characteristics hold, which is generally the case in simulation-based tuning of advanced controllers under uncertainty [194].

Assumption 1 *(1) The worst-case uncertainty $\delta^\star(\theta) \in \operatorname{argmax}_{\delta\in\Delta} f(\theta,\delta)$ cannot be determined from prior knowledge.*

(2) The feasible sets Θ and Δ are known and compact.

(3) The closed-loop performance measure $f(\theta,\delta)$ is fully black-box in nature such that no closed-form expression exists for f and it does not have any known special structure such as convexity or linearity.

(4) The total dimension of the inputs $p = p_1 + p_2$ is typically not too large; $p \leq 20$ is a good rule-of-thumb.

(5) When the closed-loop performance performance is evaluated, we only observe $f(\theta,\delta)$, meaning that first- or second-order derivatives cannot be evaluated.

(6) The observations of $f(\theta,\delta)$ are corrupted by noise. That is, $y = f(\theta,\delta) + \epsilon$, where $\epsilon \in \mathcal{N}(0,\sigma_\epsilon^2)$.

Characteristics (1)-(3) in Assumption 1 imply minimal restrictions on the structure of the to-be-designed controller such that the proposed method for controller auto-tuning can be applied even when the control law is defined implicitly—for example, as is the case in model predictive control (MPC). Characteristic (5) prevents application of derivative-based optimization methods for solving (4.1). For simplicity, characteristic (6) assumes the effective noise ϵ leading to noisy observations y of the closed-loop performance measure is normally distributed with zero mean. The variance of noise can be treated as a hyperparmaeter, as discussed in Section 4.4. Notice that the closed-loop performance measure $f(\theta,\delta)$ is quantified through possibly expensive simulations of the closed-loop system using a process simulator.

[1] We refer to the process simulator as "high-fidelity" to denote the fact that it can be a computationally-expensive model, such as a multiscale model, built from a collection of software codes/packages.

As such, the performance measure can be queried a limited number of times; often on the order of a few hundred of closed-loop simulations.

Remark 1 *Although δ can in principle represent any source of uncertainty, this may lead to a high-dimensional representation of δ due to the time-varying nature of control problems. As such, this may not satisfy characteristic (4) in Assumption 1. Instead, δ should represent the key time-invariant uncertainties (e.g., sensitive model parameters and/or initial conditions) that have the most dominant influence on the performance measure f. If prior knowledge about the dominant time-invariant uncertainties is not available, it can be obtained via global sensitivity analysis [274], which can be facilitated via surrogate modeling [154]. Notice that, although not included in δ, the effect of time-varying process and measurement noise is accounted for through noisy observations of f; see characteristic (6) in Assumption 1.*

The most direct way to solve Problem (4.1) would be via a nested optimization approach wherein an inner maximization is performed for each iteration of an outer minimization algorithm [165]. This approach, however, will expend excessive effort computing the worst-case closed-loop performance for every selected design variables θ, which is not appropriate when dealing with expensive evaluations of f using a high-fidelity process simulator. This also precludes the use of evolutionary algorithms [177], which are popular techniques when the objective function can be evaluated a large number of times. Alternatively, we look to reformulate (4.1) as a *bandit feedback* problem [266]. The main idea is to sequentially select $(\theta_t, \delta_t) \in \Theta \times \Delta$ at every iteration $t = 1, 2, \ldots, N_t$ (here, "iteration" refers to a single closed-loop simulation), and receive the corresponding noisy observations of the cost $y_t = f(\theta_t, \delta_t) + \epsilon_t$. Our regret in this decision can be quantified in terms of the *instantaneous robust-regret* r_t^δ, which is defined as

$$r_t^\delta = \max_{\delta \in \Delta} f(\theta_t, \delta) - \max_{\delta \in \Delta} f(\theta^\star, \delta), \tag{4.2}$$

where $\theta^\star \in \operatorname{argmin}_{\theta \in \Theta} \max_{\delta \in \Delta} f(\theta, \delta)$ is any global solution to (4.1). In words, the robust-regret in (4.2) quantifies how far away our suggested decision θ_t is from the best possible solution $\theta^\star$ (in units of the objective function). This definition is analogous to the standard regret definition $g(\theta_t) - g(\theta^\star)$ in the multi-armed bandit literature [296] when our objective function is defined as the worst-case realization of the performance function $g(\theta) = \max_{\delta \in \Delta} f(\theta, \delta)$. Ideally, we could derive an algorithm that minimizes the *cumulative robust-regret* after T iterations $R_T^\delta = \sum_{t=1}^T r_t^\delta$; however, these quantities cannot be revealed to the algorithm since they require perfect knowledge of the global solution. A viable alternative is to select an algorithm that has *no robust-regret*, i.e., $\lim_{T \to \infty} \frac{1}{T} R_T^\delta = 0$ [24]. The only way that the average robust-regret can approach zero is for the instantaneous robust-regret to approach zero, since $r_t^\delta \geq 0$ must be non-negative. This implies that there exists a $t > 0$ such that $\max_{\delta \in \Delta} f(\theta_t, \delta)$ is arbitrarily close to $\max_{\delta \in \Delta} f(\theta^\star, \delta)$ and the algorithm converges as long as R_T^δ grows sublinearliy with T. In the absence of uncertainty (i.e., the nominal setting of $\Delta = \{\hat{\delta}\}$), we can easily find the point $\{\theta_1, \ldots, \theta_T\}$ that minimizes the (non-robust) regret by selecting the point that produces the smallest value of $f(\theta_t, \hat{\delta})$. This is no longer true in the robust case, however, due to the inclusion of the max operator in (4.2).

Therefore, we require a new recommendation procedure in addition to the selection policy for (θ_t, δ_t). In the next section, we present a variant of the sequential learning algorithm in

[24], referred to as adversarially robust Bayesian optimization, that can achieve the desired no robust-regret property using a combined Gaussian process (GP) model for $f(\theta, \delta)$, which simultaneously models the effect of the design variables and uncertainty realizations on the closed-loop performance measure.

4.3 Adversarially Robust Bayesian Optimization

In this section, we first review Gaussian process (GP) regression for data-driven modeling of the closed-loop performance measure. We will then present the adversarially robust Bayesian optimization (ARBO) algorithm, followed by an overview of established theoretical results [24] related to the robust-regret when solving (4.1) under the conditions specified in Assumption 1.

Gaussian Process Regression

Let $\mathbf{x} = [\theta^\top, \delta^\top]^\top \in \mathcal{X}$ denote the concatenated vector of design variables and uncertainties, where $\mathcal{X} = \Theta \times \Delta \subset \mathbb{R}^p$ and $p = p_1 + p_2$. We interchangeably denote $f(\theta, \delta)$ as $f(\mathbf{x})$ (and vice versa) throughout the paper. Since the structure of f is not known, we cannot make rigid parametric assumptions for f. However, without further assumptions, it would be impossible to achieve sublinear robust-regret for (4.1); for example, f could be discontinuous at every input $\mathbf{x} \in \mathcal{X}$ in the worst-case. Therefore, we assume that a certain degree of smoothness holds in practice, such that we can leverage GP models that enforce smoothness implicitly without making any parametric assumptions. The basic idea underpinning GPs is that the function values $f(\mathbf{x})$, associated with different values of $\mathbf{x}$, are random variables and any finite collection of these random variables have a joint Gaussian distribution [227]. A GP distribution, denoted by $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, is parametrized by a prior mean function $\mu(\mathbf{x})$ and a covariance (or kernel) function $k(\mathbf{x}, \mathbf{x}')$. Without loss of generality, we assume that the prior is zero mean, i.e., $\mu(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$.[2] The chosen class of covariance functions determines the properties of the fitted functions. In this work, we will focus on stationary covariance functions from the Matern class [86], defined as

$$k(\mathbf{x}, \mathbf{x}'; \nu, \Psi) = \zeta^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} r(\mathbf{x}, \mathbf{x}')) B_\nu \left(\sqrt{2\nu} r(\mathbf{x}, \mathbf{x}') \right), \tag{4.3}$$

where $r(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}') L^{-2} (\mathbf{x} - \mathbf{x}')}$ is the scaled Euclidean distance, $L = \text{diag}(l_1, \ldots, l_p)$ is a diagonal scaling matrix composed of length-scale parameters $l_1, \ldots, l_p > 0$, ν is a parameter that dictates smoothness (i.e., the corresponding function is $\lceil \nu/2 - 1 \rceil$ times differentiable), ζ^2 is a scaling factor for the output variance, Γ and B_ν are the Gamma and modified Bessel functions, respectively, and $\Psi = \{l_1, \ldots, l_p, \zeta\}$ are the hyperparameters of the kernel for a fixed value of ν.

Training a GP model corresponds to calibrating $\{\Psi, \sigma_\epsilon\}$ to the available data. For now, we assume the kernel hyperparameters are known and discuss the training procedure further in Section 4.4. Although we focus on kernels of the form (4.3) for simplicity, many other

[2] This can easily be achieved by normalizing the data before training, as discussed in, e.g., [32].

kernels are available and can be used in place of this structure, if needed. Furthermore, one can treat the kernel structure as an additional hyperparameter that is sequentially updated at each step of the Bayesian optimization process. However, since this introduces a set of conditional hyperparameters (corresponding to the internal parameters specific to each kernel), this can substantially increase the GP training cost.

A key advantage of GPs, in addition to their non-parametric nature, is the availability of simple analytic expressions for the posterior distribution of $f(\mathbf{x})$ for any input $\mathbf{x} \in \mathcal{X}$. Let us assume that we have t previous observations of the objective $\mathbf{y}_t = [y_1, \dots, y_t]^\top$ at inputs $\mathbf{X}_t = \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$. The GP model can account for the fact that these measurements are noisy, i.e., $y_t = f(\mathbf{x}_t) + \epsilon_t$ where $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Given that the noise ϵ_t obeys a normal distribution, the posterior $f|\mathbf{X}_t, \mathbf{y}_t$ remains a GP $\mathcal{GP}(\mu_t(\mathbf{x}), k_t(\mathbf{x}, \mathbf{x}'))$ with the following expressions for the mean μ_t, covariance k_t, and variance σ_t^2 [227]

$$\mu_t(\mathbf{x}) = \mathbf{k}_t^\top(\mathbf{x}) \left(\mathbf{K}_t + \sigma_\epsilon^2 \mathbf{I}_t\right)^{-1} \mathbf{y}_t, \tag{4.4a}$$

$$k_t(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_t^\top(\mathbf{x}) \left(\mathbf{K}_t + \sigma_\epsilon^2 \mathbf{I}_t\right)^{-1} \mathbf{k}_t(\mathbf{x}'), \tag{4.4b}$$

$$\sigma_t^2(\mathbf{x}) = k_t(\mathbf{x}, \mathbf{x}), \tag{4.4c}$$

where $\mathbf{k}_t(\mathbf{x}) = [k_j(\mathbf{x}_1, \mathbf{x}), \dots k_j(\mathbf{x}_t, \mathbf{x})]^\top$ contains the covariances between the input $\mathbf{x}$ and observed data points $\mathbf{X}_t$, the covariance matrix $\mathbf{K}_t$ has entries $[\mathbf{K}_t]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ for all $i, j \in \{1, \dots, t\}$, and $\mathbf{I}_t$ is the $t \times t$ identify matrix. The main advantage of the posterior GP expressions in (4.4) is that they can be used to generate *confidence bounds* on the prediction of $f(\theta, \delta)$ for any choice of input. Both the upper and lower confidence bounds will be leveraged in the development of the ARBO algorithm, as described next.

ARBO Algorithm

Given a so-called *exploration parameter* β_t, we can define the following upper and lower confidence bounds on f

$$\text{UCB}_t(\theta, \delta) = \mu_t(\theta, \delta) + \beta_t^{1/2} \sigma_t(\theta, \delta), \tag{4.5a}$$

$$\text{LCB}_t(\theta, \delta) = \mu_t(\theta, \delta) - \beta_t^{1/2} \sigma_t(\theta, \delta), \tag{4.5b}$$

which are readily determined from the posterior GP in (4.4). For sufficiently large choices of β_t, these confidence bounds will be large enough to ensure the no robust-regret property with high probability (see Theorem 1). The ARBO algorithm [24], which relies on the lower and upper confidence bounds (4.5), is presented in Algorithm 1. The suggested θ_t at each iteration is the one that has the minimum "robust" *lower* confidence bound, as given in (4.8). For this choice of θ_t, we must select a feasible uncertainty sample. According to (4.9), we select the uncertainty value δ_t that maximizes the *upper* confidence bound. We can interpret these opposite choices as: (i) optimistic selections under uncertainty for θ_t and (ii) pessimistic selections under uncertainty for the anticipated worst-case point δ_t. While the choice (i) is common to traditional BO algorithms that utilize confidence bounds, the choice (ii) is unique to ARBO to mitigate any possible negative effects caused by the uncertainty. Once the main loop in Algorithm 1 has been completed, a final "recommended" point must be selected from the sequence $\{\theta_1, \dots, \theta_T\}$. Although there are many potential choices, we

choose the one that minimizes a pessimistic bound on the robust-regret in (4.10). To this end, let us assume $f(\theta,\delta) \leq \text{UCB}_{t-1}(\theta,\delta)$ for all $(\theta,\delta) \in \Theta \times \Delta$; this condition will be more formally stated later. Then, we can define the following pessimistic estimate of r_t^δ

$$\bar{r}_t^\delta = \max_{\delta\in\Delta} \text{UCB}_{t-1}(\theta_t,\delta) - f^\star, \tag{4.6}$$

where $f^\star = \max_{\delta\in\Delta} f(\theta^\star,\delta) = \min_{\theta\in\Theta}\max_{\delta\in\Delta} f(\theta,\delta)$, which must satisfy $r_t^\delta \leq \bar{r}_t^\delta$ for all $t \geq 1$ under the above-stated assumption. The main difference between (4.2) and (4.6) is that the algorithm has enough information to identify the index $t^\star$ that minimizes $\bar{r}_t^\delta$ since the global solution does not depend on t. Yet, $\bar{r}_t^\delta$ is related to another important quantity in bandit optimization termed the *simple robust-regret* after T iterations, which is denoted by S_T^δ and defined as

$$S_T^\delta = \min_{t\in\{1,\ldots,T\}} r_t^\delta = \min_{t\in\{1,\ldots,T\}} \max_{\delta\in\Delta} f(\theta_t,\delta) - f^\star. \tag{4.7}$$

It is evident that $S_T^\delta \leq \bar{r}_{t^\star}^\delta$ for all $T \geq 1$. This in turn implies that bounds established on $\bar{r}_{t^\star}^\delta$ immediately transfer to the simple robust-regret S_T^δ, as discussed in the next section. Notice that Algorithm 1 relies on only a single expensive closed-loop simulation run to be performed at every iteration, which is significantly fewer than the vast majority of available alternatives, such as [165, 298].

Upper Bound on Simple Robust-Regret

The ARBO Algorithm 1 requires selection of the exploration parameters $\{\beta_t\}_{t\geq 1}$ that specify the width of the confidence intervals on f. To this end, we rely on a simple result from [266] to select this sequence. We will focus on the case of a finite set $\mathcal{X} = \Theta \times \Delta$ for simplicity, and discuss the extension to a compact and convex set later.

Lemma 1 (Confidence bounds [266]) *Let $f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x},\mathbf{x}'))$ be a sample of a GP for which noisy observations $y_t = f(\mathbf{x}_t) + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(0,\sigma_\epsilon^2)$ are available. Let $\beta_t = 2\log(|\mathcal{X}|t^2\pi^2/(6\alpha))$ for a specified failure probability $\alpha \in (0,1)$ and finite discrete set $|\mathcal{X}| < \infty$. Then, the following bounds on the objective function $f(\mathbf{x})$*

$$f(\mathbf{x}) \in [\text{LCB}_{t-1}(\mathbf{x}), \text{UCB}_{t-1}(\mathbf{x})], \quad \forall \mathbf{x} \in \mathcal{X}, \forall t \geq 1, \tag{4.11}$$

hold with probability (over the GP posterior at every iteration) at least $1-\alpha$.

Next, we define the *maximum information gain* (MIG), which provides a measure of the informativeness of any finite set of sampling points $\mathcal{A} \subset \mathcal{X}$ [55].

Definition 1 *Let $\mathcal{A} \subset \mathcal{X}$ denote any subset of sampling points from $\mathcal{X}$ and let f be a sample of a GP model with the same sampling conditions stated in Lemma 1. The maximum information gain for f under t measurements is defined as*

$$\gamma_t = \max_{\mathcal{A}\subset\mathcal{X}:|\mathcal{A}|=t} \frac{1}{2}\log\det(\mathbf{I}_t + \sigma_\epsilon^{-2}\mathbf{K}_\mathcal{A}), \tag{4.12}$$

where $\mathbf{K}_\mathcal{A} = [k(\mathbf{x},\mathbf{x}')]_{\mathbf{x},\mathbf{x}'\in\mathcal{A}}$ is the kernel matrix. Note that the term inside of the max in (4.12) is the Shannon mutual information between f and the observations at points $\mathbf{x} \in \mathcal{A}$. ◁

Algorithm 1 The robust sequential learning algorithm for ARBO.

Input: The set of the design variables Θ and the uncertainty Δ; kernel k corresponding to GP prior; exploration parameters $\{\beta_t\}_{t\geq 1}$; and total number of iterations T.

1: Initialize the mean and standard deviation $(\mu_0, \sigma_0) \leftarrow (0, k^{1/2})$.
2: **for** $t = 1$ to T **do**
3: Solve the following min-max optimization problem for θ_t

$$\theta_t = \underset{\theta\in\Theta}{\operatorname{argmin}} \max_{\delta\in\Delta} \mathrm{LCB}_{t-1}(\theta, \delta). \tag{4.8}$$

4: Solve the following maximization problem for δ_t

$$\delta_t = \underset{\delta\in\Delta}{\operatorname{argmax}}\, \mathrm{UCB}_{t-1}(\theta_t, \delta). \tag{4.9}$$

5: Run a closed-loop simulation at $\mathbf{x}_t = [\theta_t^\top, \delta_t^\top]^\top$ to compute performance measure $y_t = f(\theta_t, \delta_t) + \epsilon_t$.
6: Perform Bayesian posterior update to estimate μ_t, σ_t, LCB_t and UCB_t using (4.4) and (4.5) by including the latest query of the closed-loop performance measure $\{\mathbf{x}_t, y_t\}$.
7: **end for**
8: Return the point $\theta_{t^\star}$ with the smallest upper confidence bound (our best guess of the optimal design variables)

$$t^\star = \underset{t\in\{1,\ldots,T\}}{\operatorname{argmin}} \max_{\delta\in\Delta} \mathrm{UCB}_{t-1}(\theta_t, \delta). \tag{4.10}$$

We can now state the main theorem that bounds the performance of the ARBO Algorithm 1. We give a brief sketch of the proof of this result, which is a slightly different version of that provided in [24, Supplementary Material].

Theorem 1 (Upper ARBO Performance Bound [24]) *Fix $\alpha \in (0,1)$ and also set $\beta_t = 2\log(|\mathcal{X}|t^2\pi^2/(6\alpha))$, and $T \geq 1$. Running the ARBO algorithm for a sample f of a GP with zero mean and kernel $k(\mathbf{x}, \mathbf{x}')$, the simple robust-regret must satisfy*

$$Pr\left\{ S_T^\delta \leq \bar{r}_{t^\star}^\delta \leq \sqrt{\frac{C_1\beta_T\gamma_T}{T}} \right\} \geq 1-\alpha, \tag{4.13}$$

where $C_1 = 8/\log(1+\sigma_\epsilon^{-2})$.

Proof 1 *From Lemma 1, we know that $\mathrm{LCB}_{t-1}(\mathbf{x}) \leq f(\mathbf{x}) \leq \mathrm{UCB}_{t-1}(\mathbf{x})$ holds for all $\mathbf{x} \in \mathcal{X}, t \geq 1$ with probability greater than or equal to $1-\alpha$. Given this, from the definitions in*

(4.2) *and* (4.6)*, we have*

$$\begin{aligned} r_t^\delta = \max_{\delta\in\Delta} f(\theta_t,\delta) - \min_{\theta\in\Theta}\max_{\delta\in\Delta} f(\theta,\delta) \le \bar{r}_t^\delta &= \max_{\delta\in\Delta} ucb_{t-1}(\theta_t,\delta) - \min_{\theta\in\Theta}\max_{\delta\in\Delta} f(\theta,\delta), \\ &= ucb_{t-1}(\theta_t,\delta_t) - \min_{\theta\in\Theta}\max_{\delta\in\Delta} f(\theta,\delta), \\ &\le ucb_{t-1}(\theta_t,\delta_t) - \min_{\theta\in\Theta}\max_{\delta\in\Delta} lcb_{t-1}(\theta,\delta), \\ &= ucb_{t-1}(\theta_t,\delta_t) - \max_{\delta\in\Delta} lcb_{t-1}(\theta_t,\delta), \\ &\le ucb_{t-1}(\theta_t,\delta_t) - lcb_{t-1}(\theta_t,\delta_t), \\ &= 2\beta_t^{1/2}\sigma_{t-1}(\theta_t,\delta_t), \end{aligned}$$

where the first line follows from the upper bound on f, the second line follows from the definition of δ_t in (4.9)*, the third line follows from the lower bound on f, the fourth line follows from the definition of θ_t in* (4.8)*, the fifth line follows from the fact that* $\max_{\delta\in\Delta} \mathrm{LCB}_{t-1}(\theta_t,\delta) \ge \mathrm{LCB}_{t-1}(\theta_t,\delta_t)$ *for any feasible choice of $\delta_t \in \Delta$, and the sixth line follows from the difference between the confidence bounds in* (4.5)*. Given this bound, we can also see that the following sequence of inequalities must hold with probability $\ge 1-\alpha$*

$$(R_T^\delta)^2 \le T\sum_{t=1}^T (r_t^\delta)^2 \le 4\beta_T \sum_{t=1}^T \sigma_{t-1}^2(\theta_t,\delta_t),$$

where the first step follows from the Cauchy-Schwarz inequality and the second step follows from the monotonicity of the sequence $\{\beta_t\}_{t\ge1}$. Next, we use a special case of [266, Lemma 5.4] to establish a bound on the sum of variances in terms of the MIG (4.12)

$$4\sum_{t=1}^T \sigma_{t-1}^2(\theta_t,\delta_t) \le C_1\gamma_T,$$

for $C_1 = 8/\log(1+\sigma_\epsilon^{-2})$. From these results, it follows that $Pr\{R_T^\delta \le \sqrt{C_1 T\beta_T\gamma_T}\} \ge 1-\alpha$. The assertion in (4.13) *follows by noting that the minimum of a sequence must be less than or equal to the average, i.e., $S_T^\delta \le \frac{1}{T}R_T^\delta$, in addition to the fact that the same inequalities hold for $\bar{r}_t^\delta$ in place of r_t^δ.* ■

As the total number of iterations T increases in (4.13), we observe that the simple robust-regret gets closer to the desired value of zero, implying the global minimax solution has been found in the limit as $T \to \infty$, as long as the numerator $C_1\beta_T\gamma_T \in o(T)$, where o is little-o notation that implies $C_1\beta_T\gamma_T$ decays faster than T. The choice of β_T in Theorem 1 clearly shows logarithmic growth with respect to T. However, we also require bounds on the MIG γ_T to establish convergence. It was shown in [266] that γ_T has sublinear dependence with respect to T for many commonly used kernels, including the Mateŕn class, such that the ARBO algorithm converges to function evaluations near $\theta^\star$ with high probability for sufficiently small choices of α. This is a key advantage of the confidence bound-based ARBO algorithm compared to available alternatives whose theoretical properties have yet to be understood.

4.4 Practical Implementation of ARBO

In this section, we discuss some of the main aspects in practical implementation of the ARBO Algorithm 1, as also considered in the case studies presented in Section 4.5.

Choice of Exploration Constant β_t

Lemma 1 and Theorem 1 only hold for discrete spaces $\mathcal{X}$. However, using the discretization technique introduced in [266], these results can be extended to continuous spaces that are compact and convex. The main added assumption is that the kernel function $k(\mathbf{x}, \mathbf{x}')$ must be chosen such that it ensures the following high probability bounds on the derivatives of f for some constants $a, b > 0$

$$\Pr\left\{\sup_{\mathbf{x}\in\mathcal{X}}\left|\frac{\partial f(\mathbf{x})}{\partial x_i}\right| > L\right\} \leq ae^{-(L/b)^2},\ \forall i = 1,\dots,p, \forall L > 0. \tag{4.14}$$

Whenever this condition holds, the results in Lemma 1 and Theorem 1 can be generalized to any compact and convex set $\mathcal{X} \subset [0, r]^p$ by enlarging the exploration constant

$$\beta_t = 2\log\left(\frac{2\pi^2 t^2}{3\alpha}\right) + 2p\log\left(t^2 pbr\log^{\frac{1}{2}}(4pa/\alpha)\right). \tag{4.15}$$

To the best of our knowledge, these results have not yet been extended to arbitrary non-convex sets. However, this may not pose a challenge in practice since the choices of β_t are generally known to be conservative [117]. In the case studies in Section 4.5, we select $\beta_t = \beta_0 p\log(2t)$ to capture the dominant dependence of the exploration constant on t and p. A tyical value for β_0 is 0.1. An interesting direction for future work includes establishing a more robust way to select $\{\beta_t\}_{t\geq 1}$ for specific applications.

Estimation of GP Hyperparameters

The results in Lemma 1 and Theorem 1 assume that the hyperparameters $\{\Psi, \sigma_\epsilon\}$ of the GP prior for f are known exactly. Since this is often not true in practice, we rely on the maximum likelihood estimation (MLE) framework to determine the optimal hyperparameters $\{\Psi_t^\star, \sigma_\epsilon^\star\}$ that, at every iteration t, maximize the log-likelihood $\mathcal{L}_t(\Psi, \sigma_\epsilon)$ [227]

$$\{\Psi_t^\star, \sigma_{\epsilon,t}^\star\} \in \operatorname*{argmax}_{\Psi,\sigma_\epsilon} \mathcal{L}_t(\Psi, \sigma_\epsilon) = \log(p(\mathbf{y}_t|\mathbf{X}_t, \Psi, \sigma_\epsilon)). \tag{4.16}$$

Based on the GP prior, the measured data vector $\mathbf{y}_t$ must be distributed according to a multivariate Gaussian distribution of the following form

$$\mathbf{y}_t \sim \mathcal{N}(0, \Sigma_t(\Psi, \sigma_\epsilon)), \quad [\Sigma_t(\Psi, \sigma_\epsilon)]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j|\Psi) + \sigma_\epsilon^2\delta_{ij}, \quad \forall i, j \in \{1,\dots,t\}. \tag{4.17}$$

Using this representation, an analytical expression for the log-likelihood function can be derived as

$$\mathcal{L}_t(\Psi, \sigma_\epsilon) = -\mathbf{y}_t^\top\Sigma_t^{-1}\mathbf{y}_t - \frac{1}{2}\log(\det(\Sigma_t)) - \frac{p}{2}\log(2\pi). \tag{4.18}$$

The optimization problem (4.16) is a nonlinear program that can be solved using gradient-based methods (e.g., IPOPT [300]) since (4.18) is a smooth, differentiable function. To ensure the optimizer does not get stuck in a local solution, it is useful to "warm-start" the local solver with the best solution found from a heuristic global optimization method such as

the DIRECT solver [74]. Notice that the "warm-start" approach will introduce an additional step into Algorithm 1 that could be somewhat computationally expensive depending on the size of the optimization (4.16). A simple way to reduce the computational cost associated with hyperparameter estimation is to update the hyperparameters of the GP model only periodically, instead of at every iteration. In this work, we exclusively use the Python package `GPy` [96] to train and make predictions with GP models.

Minimax Optimization for LCB_{t-1}

Our analysis in Section 4.3 assumed that we could exactly optimize the acquisition functions defined in terms of the lower and upper confidence bounds in (4.8) and (4.9). The maximization problem (4.9) resembles the sub-problem that arises in the standard BO, suggesting that the same basic principles can be leveraged to develop a practical solution method for the ARBO Algorithm 1. Here, we propose to use a combination of derivative-free search with a local gradient-based solver for the min-max optimization (4.8) at each itertation. Note that since $\text{LCB}_{t-1}(\theta, \delta)$ may be non-convex with respect to θ and non-concave with respect to δ, we cannot use traditional alternating gradient descent-ascent methods, as they may not even locally converge [114].

The proposed approach partially exploits the differentiability of $\text{LCB}_{t-1}(\theta, \delta)$. Let $g_{t-1}(\theta) = \max_{\delta \in \Delta} \text{LCB}_{t-1}(\theta, \delta)$ denote the optimal objective value for the inner maximization problem. We can then equivalently formulate (4.8) as

$$\min_{\theta \in \Theta} \; g_{t-1}(\theta), \tag{4.19}$$

where g_{t-1} is a black-box function that can only be evaluated by calling an internal algorithm to approximate $g_{t-1}(\theta)$ for any choice of $\theta \in \Theta$. Since, for any fixed θ, $\text{LCB}_{t-1}(\theta, \delta)$ is a smooth function whose derivatives can be efficiently computed, we can rely on gradient-based solvers (e.g., the well-known L-BFGS-B algorithm [323]) to quickly converge to a local optimum. Since we need a good estimate of the *global solution* for the inner maximization, we need some type of globalization strategy. One approach is to apply a random multi-start for several δ points, with the largest converged objective value being returned as our best approximation to $g_{t-1}(\theta)$. The initial guess can be obtained by randomly sampling Δ or by computing a large number of random samples of δ to evaluate $\text{LCB}_{t-1}(\theta, \delta)$ with θ fixed, and choose the ones that lead to the highest $\text{LCB}_{t-1}(\theta, \delta)$ in order to warm start the local solver. We then treat (4.19) as a black-box optimization problem that can be solved with any number of available derivative-free optimization methods. In this work, we rely on BOBYQA [41, 217], which is a local trust region-based approach, but various derivative-free algorithms can be generally applied [17]. We again rely on a random multi-start procedure to protect against local solutions for this outer minimization problem; however, since g_{t-1} is fairly expensive to evaluate, we must carefully select the number of repeats to ensure a solution can be found in a reasonable amount of time.

4.5 Case Studies

In this section, we demonstrate the performance of the ARBO algorithm on two problems. The first case study is an illustrative example that is meant to showcase several implementation details of Algorithm 1. Since the exact knowledge of the function and its min-max solution is available, we can directly compute the key performance assessment measures, such as the simple robust-regret, in the illustrative example. The second case study, on the other hand, focuses on a challenging nonlinear MPC (NMPC) auto-tuning problem. Since this auto-tuning problem involves a nonlinear plant simulator, we do not have exact knowledge of the true solution and thus cannot use simple robust-regret as our performance measure. Instead, we evaluate the solution quality directly in terms of the closed-loop performance and constraint satisfaction profiles. The main goal of this section is to show that ARBO can more reliably find high-performance tuning parameters with significantly fewer closed-loop simulations than alternative methods.

Illustrative Example

Consider a problem in the form of (4.1), with the following analytic expression for f

$$f(\theta, \delta) = \sin(\theta\delta) + \sqrt{\delta}\theta^2 - 0.5\theta, \tag{4.20}$$

where $\Theta = [-1, 2]$ is the feasible set of decision variable and $\Delta = [2, 4]$ is the feasible set of the uncertainty. Throughout this section, (4.20) is unknown to any of the black-box algorithms, and is only used for assessing our the regret-based performance measures. Figure 4.1 shows a plot of $f(\theta, \delta^{(i)})$ versus θ for a large number of random samples $\delta^{(i)} \in \Delta$, with the worst-case function $g(\theta) = \max_{\delta\in\Delta} f(\theta, \delta)$ shown with a black dashed line. From this plot, we can see that $\theta^\star = -0.3573$, which corresponds to an optimal minimax objective value of $f^\star = -0.2961$.

For this illustrative problem we can identify the globally optimal solution in the domain of interest, thus the regret metrics become readily available for assesing the convergence of the proposed algorithm. Therefore, we use the simple robust-regret S_T^δ as our metric since we aim to identify this robust solution in as few iterations as possible. Theorem 1 highlights the importance of the β_t sequence within Algorithm 1, as this is the main tool used to navigate the exploitation-exploration tradeoff in the joint $\{\theta, \delta\}$ space. To better illustrate this point, we compare ARBO to the a purely exploitative approach, namely a *Gaussian process-based robust optimization* (GP-RO) approach. In GP-RO, we completely ignore the variance information provided by the GP model for $f(\theta, \delta)$ and, instead, sample $\theta = \operatorname{argmin}_{\theta\in\Theta} \max_{\delta\in\Delta} \mu_{t-1}(\theta, \delta)$ and $\delta_t = \operatorname{argmax}_{\delta\in\Delta} \mu_{t-1}(\theta_t, \delta)$. Similarly, for the recommendation process, we also rely only on the mean function, i.e., $\theta_{t^\star}$ is returned with $t^\star = \operatorname{argmin}_{t\in\{1,\dots,T\}} \max_{\delta\in\Delta} \mu_{t-1}(\theta_t, \delta)$.

It is well-known that determining the hyperparameters of GP models, as discussed in Section 4.4, is often unreliable for very small datasets. Thus, as opposed to starting Algorithm 1 from iteration 1, it is usually preferred to select the first N_0 points uniformly at random in any BO procedure to ensure a high-degree of exploration initially [37]. In this illustrative problem, we select $N_0 = p^2 - 1$ random points before running Algorithm 1. Since the simple

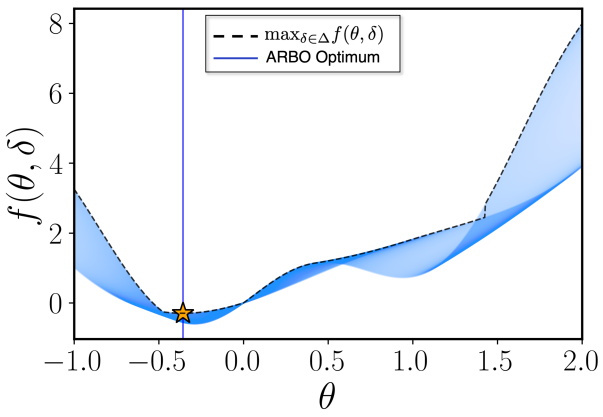

Figure 4.1: Objective function plots for various values of the uncertain parameter δ. The star symbol denotes the true minimax solution, while the vertical blue line represents the best recommended solution discovered by ARBO. The black dashed line represents the point-wise worst-case f while the collection of light-blue lines, represented as a "tube", shows the function evaluated for 1000 random realizations of δ.

robust-regret is a function of these randomly selected initial points, S_T^δ itself is a random quantity, so that showing results for a single initialization is not very informative. Instead, we repeated both the ARBO and GP-RO methods N_r times (under the same random seeds) to construct a sample average estimate for the expected simple robust-regret, i.e.,

$$\mathbb{E}\{S_T^\delta\} \approx \frac{1}{N_r}\sum_{i=1}^{N_r} S_T^{\delta,(i)}, \tag{4.21}$$

where $S_T^{\delta,(i)}$ denotes the simple robust-regret for the i^{th} run of the algorithm starting from the i^{th} set of N_0 random initial points, while $N_r = 10$. Since this estimate is constructed from a finite number of samples, we also report estimated confidence intervals computed as 1.96 times the standard deviation divided by the square root of the number of repeats (also known as the standard error formula).

The simple robust-regret plots for both ARBO and GP-RO are shown in Figure 4.2, with the estimated sample average and corresponding confidence-bound error bars on the left and the individual sample paths $S_T^{\delta,(i)}$ for all $i \in \{1,\dots,N_r\}$ shown on the right. We clearly see that ARBO consistently converges to the true robust globally optimum, within a small tolerance compared to the true $f^\star$, for all considered initial points. This leads to $\mathbb{E}\{S_T^\delta\} \approx 0$ using less than 15 function evaluations. GP-RO, on the other hand, shows considerably worse performance for the individual sample paths as well as the sample average. Figure 4.2(b) is particularly informative, as we see that in several runs, the GP-RO algorithm makes a little to no progress for the entire 30 allotted function evaluations. This behavior is not unexpected, as it is well-known to occur in algorithms that lack any degree of exploration – in this context, there is no clear incentive for GP-RO to sample in unexplored regions of the $\Theta \times \Delta$ space.

To provide additional insights into the improved performance of ARBO over GP-RO, we plot the lower confidence bound contour plots for various iterations of a single run of ARBO

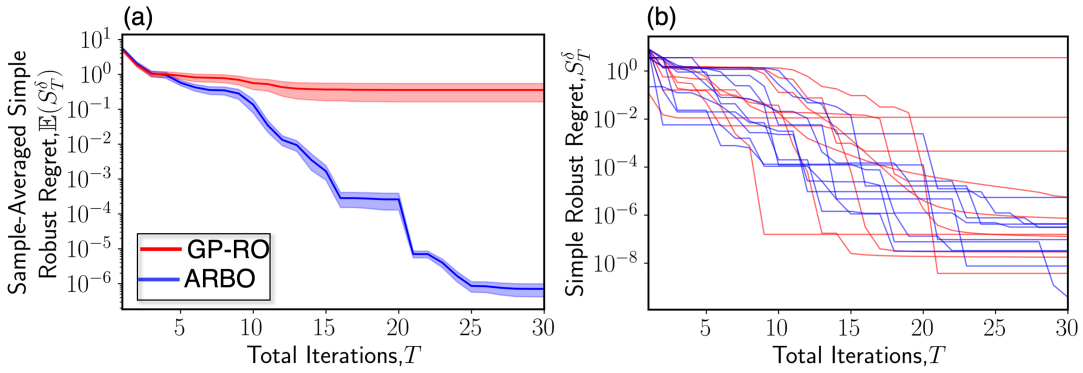

Figure 4.2: The simple robust-regret for ARBO (blue lines) and GP-RO (red lines). The runs are repeated for 10 times and the average simple robust regret is shown along with the 95% confidence intervals in (a). Individual simple robust-regret sample paths for different uncertainty realizations are shown in (b).

in Figure 4.3. We observe that in early iterations, Figure 4.3(a), the lower confidence bound attains relatively high values uniformly in most of the $\Theta \times \Delta$ space, since most of the space is unexplored. As more of the samples suggested by ARBO have been incorporated, shown as the light blue dots, we see that the lower confidence bound is able to filter out regions of the space that are not likely to be near the global minimax solution (e.g., ARBO no longer samples near $\theta = 2$ after it sees large values there). In the later iterations, Figure 4.3(b-c), we observe that the queried points start to form a pattern; The ARBO algorithm samples in a region around θ^*, while the proposed points also start converging to worst-case value for δ.

Lastly, in Figure 4.3(d), we plot the point-wise absolute error between the true function $f(\theta, \delta)$ and the mean value of the GP, i.e., the approximate objective at the final iteration. As expected, the GP at this iteration provides a very good approximation of the unknown true function in a large region around the global minimax solution denoted with a star, since the error approaches zero. Nevertheless, it provides an optimistic prediction of $f(\theta, \delta)$ elsewhere, and, in particular for $\theta > 1$. Since this optimistic prediction is still worse than our known, tested evaluation (we have already queried some points in the regions where, eventually, it is very unlikely to find the global optimum), we can adaptively exclude regions of our search space without wasting the computationally expensive samples. This highlights a fundamentally important point about the BO perspective: it is easier (i.e., fewer samples are needed) to find a globally optimal solution than building a globally accurate surrogate model. Additionally, this is the key missing component in GP-RO (which can be interpreted as ARBO with $\beta_t = 0$), as the mean predictions alone do not posses enough information about the quality of the predictions. This can lead to repeated evaluations at the same uninformative points. Moreover, the gray symbols on Figure 4.3(d), show the evolution of the recommended optimum based on the last step of Algorithm 1, which closely follows the iterative generation of recommended queried points.

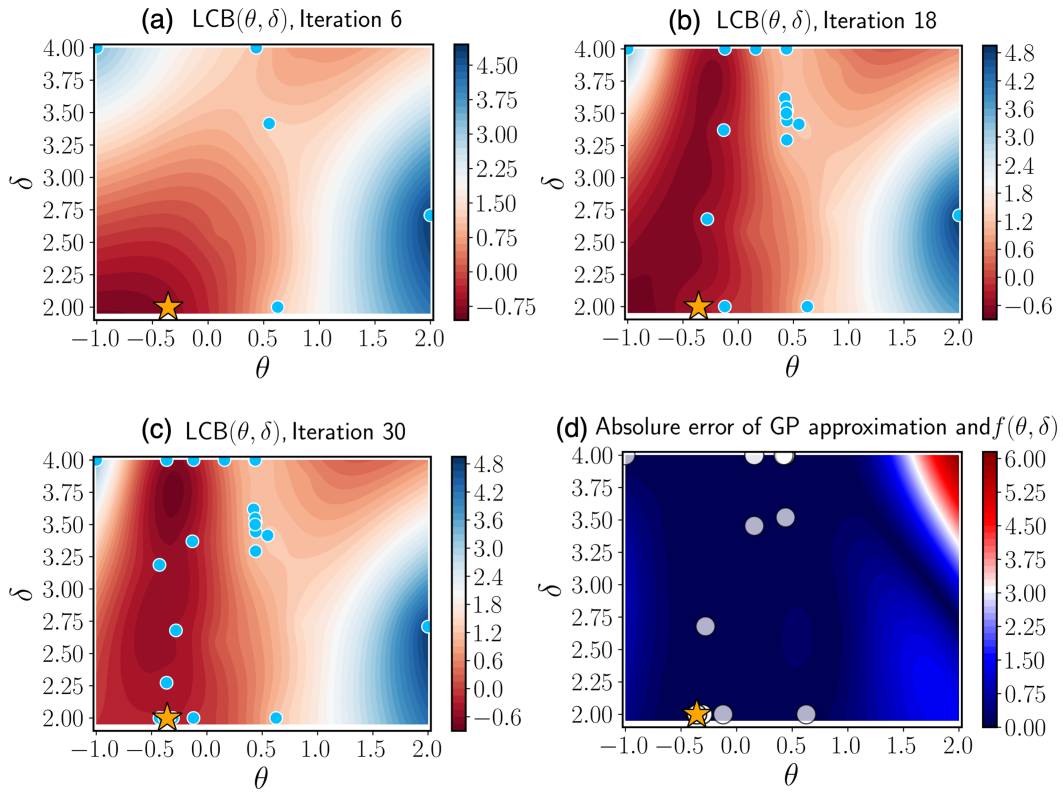

Figure 4.3: (a)-(c) Contour plots showing the convergence of ARBO; the contour plots show the lower confidence bound based on the current iteration's GP, which guides the selection of the queried θ. The sequence of optimal points to be queried $\{\theta_t, \delta_t\}$ is superimposed (light-blue circles), showing the convergence pattern. (d) Point-wise mismatch between the true function and the mean of the GP approximation at the final iteration (contour) and the sequence of recommended optimal points (circles)

Auto-tuning of NMPC for a Bioreactor

Having verified the practical implementation and important theoretical results of ARBO on an illustrative example, we now apply ARBO to the auto-tuning problem of an NMPC strategy for an uncertain bioreactor.

High-fidelity process model

We consider a benchmark continuous bioreactor problem originally presented in [4]. The dynamics of the bioreactor can be modeled by a set of three nonlinear ordinary differential equations given by

$$\dot{X}(t) = -D(t)X(t) + \mu(t)X(t), \qquad X(0) = X_0, \tag{4.22a}$$

$$\dot{S}(t) = D(t)(S_f(t) - S(t)) - \frac{1}{Y_{X/S}}\mu(t)X(t), \qquad S(0) = S_0, \tag{4.22b}$$

$$\dot{P}(t) = -D(t)P(t) + (\alpha\mu(t) + \beta)X(t), \qquad P(0) = P_0, \tag{4.22c}$$

where $X(t)$, $S(t)$, and $P(t)$ denote the biomass, substrate, and product concentration (units of g/L), respectively, with initial conditions $X_0 = 0.3$ g/L, $S_0 = 0.2$ g/L, and $P_0 = 0$ g/L; $D(t)$ is the dilution rate (units of hr^{-1}); $S_f(t)$ is concentration of substrate in the feed (units of g/L); $Y_{X/S}$ is the cell-mass yield (units of g/g); $\mu(t)$ is the specific growth rate (units of hr^{-1}); and α and β are parameters related to the product yield. The specific growth rate is assumed to follow a modified Monod kinetic law that takes into account both substrate and product inhibition

$$\mu(t) = \frac{\mu_{\max}\left(1 - \frac{P(t)}{P_m}\right) S(t)}{K_m + S(t) + \frac{S^2(t)}{K_i}}. \tag{4.23}$$

We consider $\mu_{\max}$ and P_m to be the dominant time-invariant uncertainties in the plant simulator, as explored in several previous case studies [103, 203]. Here, we assume that $\delta = (\mu_{\max}, P_m) \in \Delta = [0.75, 1.35]\ \text{hr}^{-1} \times [1.25, 1.75]$ g/L. The rest of the model parameters are assumed to be constant and are listed in Table 4.1.

Table 4.1: Known parameters for the high-fidelity bioreactor model

Fixed Parameters	**Values**	**Units**
$Y_{X/S}$	0.2	g/g
α	2.5	g/g
β	0.8	hr^{-1}
K_m	1.2	g/L
K_i	20	g/L

The states of the bioreactor model (4.22) are given by $z(t) = (X(t), S(t), P(t))$, while $u(t) = (D(t), S_f(t))$ denote the two manipulated inputs. As such, we can write (4.22) in the following state-space representation

$$\dot{z}(t) = \mathcal{F}(z(t), u(t), \delta), \quad z(0) = z_0, \tag{4.24}$$

where $\mathcal{F} : \mathbb{R}^3 \times \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^3$ is a function that represents the dynamics of the bioreactor in (4.22) and (4.23). The control objective is to maximize the amount of product extracted from the bioreactor over a finite processing time of $t_f = 6$ hr, while satisfying minimum and maximum constraints on the biomass concentration. We can generally denote such state (or path) constraints as $\mathcal{G}(z(t), u(t)) \leq 0$, which reduce to the following in this case study

$$\mathcal{G}(z(t), u(t)) = [X_{LB} - X(t), X(t) - X_{UB}, S(t) - S_{UB}] \leq 0, \tag{4.25}$$

where $X_{LB} = 0.285$ g/L and $X_{UB} = 0.385$ g/L are the lower and upper bounds on the biomass concentration, respectively, while $S_{UB} = 15$ g/L is an upper bound to substrate concentration. The manipulated inputs must also satisfy hard input constraints $u(t) \in \mathbb{U} = [0.1, 1]\ \text{hr}^{-1} \times [10, 20]$ g/L. We assume that the manipulated inputs can be updated every $\delta t = 0.1$ hr in the simulation, such that there are a total of $N_{sim} = 60$ simulation steps to compute the behavior for each uncertainty realization.

Control-relevant model and NMPC formulation

Although a dynamic model of the bioreactor is available here (i.e., we have access to $\mathcal{F}$), this is not always the case in practice. Furthermore, even when plant simulators are available, they may be excessively complex and computationally costly to use for MPC design and implementation. Thus, a more practical approach is often to construct a *control-relevant model* using system identification methods based on either plant simulation data or real plant data. Here, we used a residual neural network approach [220] that learns the "flow-map" function for time-invariant dynamic systems. In particular, this approach learns a transition function $\tilde{\mathcal{F}}(z_k, u_k)$ that can be applied recursively to predict the forward evolution of the states

$$z_{k+1} = \tilde{\mathcal{F}}(z_k, u_k) \tag{4.26}$$

from some initial condition z_0 given a future input sequence. The weights and bias parameters of the neural network representing $\tilde{\mathcal{F}}$ are trained such that the successor state at the next discrete time can be predicted given the current states and control inputs that have been determined from the plant simulator described in Section 4.5. We only collect simulation data for the nominal parameter values $\mu_{\max} = 1.0$ hr^{-1} and $P_m = 1.5$ g/L, though one could treat the unknown parameters as additional inputs to the model during training.

A deep neural network with 3 layers, 20 nodes per layer, and the Swish activation [225] function was used to represent $\tilde{\mathcal{F}}$ in (4.26). The training was efficiently carried out using `Tensorflow` via the `Keras` API [48]. Standard best practices regarding the selection of batch size, weight/bias initialization, and stochastic gradient descent optimizer settings were utilized. As such, not only this case study considers time-invariant parametric uncertainty in the plant simulator, but also plant-model mismatch with respect to the control-relevant model used for the NMPC design at hand.

Given this control-relevant model and the control objective described in Section 4.5, we formulate the following NMPC problem

$$\begin{aligned}
\min_{z_{i|k}, u_{i|k}, \varepsilon_{i|k}} \quad & \textstyle\sum_{i=0}^{N-1} \mathcal{L}(z_{i|k}, u_{i|k}) + \rho \|\varepsilon_{i|k}\|_1, && && (4.27)\\
\text{s.t.} \quad & z_{i+1|k} = \tilde{\mathcal{F}}(z_{i|k}, u_{i|k}), && \forall i = 0, \dots, N-1,\\
& \mathcal{G}(z_{i|k}, u_{i|k}) + \theta \leq \varepsilon_{i|k}, && \forall i = 0, \dots, N-1,\\
& \varepsilon_{i|k} \geq 0, && \forall i = 0, \dots, N-1,\\
& u_{i|k} \in \mathbb{U}, && \forall i = 0, \dots, N-1,\\
& z_{0|k} = z(t_k),
\end{aligned}$$

where N is the prediction horizon; $z_{i|k}$ and $u_{i|k}$ are the predicted state and control inputs i steps ahead of current time k; $z(t_k)$ is the measured state at time t_k (from the plant simulator); $\mathcal{L}(z_{i|k}, u_{i|k}) = -VD_iP_i\delta t$ is the stage cost with reactor volume $V = 10$ L; $\varepsilon_{i|k}$ are slack variables for the state constraints; ρ is a large penalty weight for state constraint violations; and $\theta \in \Theta = [0, 0.1]^2$ are the tunable backoff parameters that can be selected to improve the inherent robustness guarantees in NMPC [193]. Note the stage cost is defined as the negative of the amount of product extracted from the bioreactor over each δt period; the negative arises since we want to maximize product, but formulated our problem in terms

of a minimization. Let $u^{\star}_{0|k}(z(t_k), \theta)$ denote the first element of the solution to (4.27). We can then define the closed-loop system as the combination of (4.24) and the NMPC law

$$u(t_k) = u^{\star}_{0|k}(z(t_k), \theta), \tag{4.28}$$

where the control inputs are constant during each time interval $[t_k, t_{k+1}), \forall k \in \{0, \dots, N_{sim} - 1\}$. We solve the MPC problem by interfacing IPOPT with the CasADi library [299, 7].

Formulation of auto-tuning problem

Given the closed-loop simulation described in Section 4.5, we can now formulate the auto-tuning problem as selecting backoffs $\theta \in \Theta$ such that the worst-case mass of product (with respect to uncertainties $\delta \in \Delta$) is maximized while the biomass constraints (4.25) are not (significantly) violated. We denote the objective function of the NMPC tuning problem as $\phi(\theta, \delta)$.To formulate the auto-tuning problem as in (4.1), we must select $\phi(\theta, \delta)$ to be a weighted combination of productivity and constraint violations. Thus, we mathematically represent the overall control objective in terms of the closed-loop simulation outputs as follows

$$\phi(\theta, \delta) = \sum_{k=1}^{N_{sim}} \mathcal{L}(z(t_k), u(t_k)) + w \| [\mathcal{G}(z(t_k), u(t_k))]^{+} \|_1, \tag{4.29}$$

where $[a]^{+} = \max\{a, 0\}$ denotes the element-wise positive part operator and $w = 20$ is a weight parameter chosen to have a significant penalty associated with constraint violations.

Results and performance comparison

In this subsection we compare the performance of ARBO and GP-RO to the previously described NMPC auto-tuning under uncertainty problem. We allocate a maximum number of 50 iterations for both robust BO variants. We use Latin Hypercube Sampling [148] to generate the first 15 samples in $\Theta \times \Delta$ to ensure sufficient initial coverage of the search space,while the same initial samples are used for both algorithms. There are two factors that mainly determine the overall evolution of querying and recommended points; The initialization of the algorithm, i.e., the initial samples that construct the GP, as well as the discovery of the of true optimal solutions in the robust BO problems. Note that ARBO and GP-RO as defined in this work, involve the solution of one nested (step 3 of Algorithm 1), one simple (step 4) and a sequence of simple (step 8) optimization problems, which, albeit relatively computationally inexpensive, may lead to incorrect solutions if the used local solvers are "trapped" in local minima. Therefore, to examine the consistency of the proposed algorithm, we repeat the optimization procedure $N_r = 5$ times.

From examining the sequence of recommended optimum values, we have observed a similar behavior to the illustrative problem; ARBO consistently explores the decision variables space Θ while GP-RO recommendations do not update in several cases. To this end, we analyze the closed-loop performance for the final recommended point of the ARBO and GP-RO algorithms by evaluating $\{f(\theta_{t^{\star}}, \delta^{(i)})\}_{i=1}^{N_s}$ at $N_s = 150$ LHS-based sampled uncertainty values $\delta^{(i)} \in \Delta$. The results are directly compared with the corresponding performance when

a nominal value for backoffs, i.e., $\theta_0 = [0, 0]$ is used. Note that in this analysis we use all performance samples that were discovered by varying the final recommended point of ARBO and GP-RO as discovered among the 5 replicates.

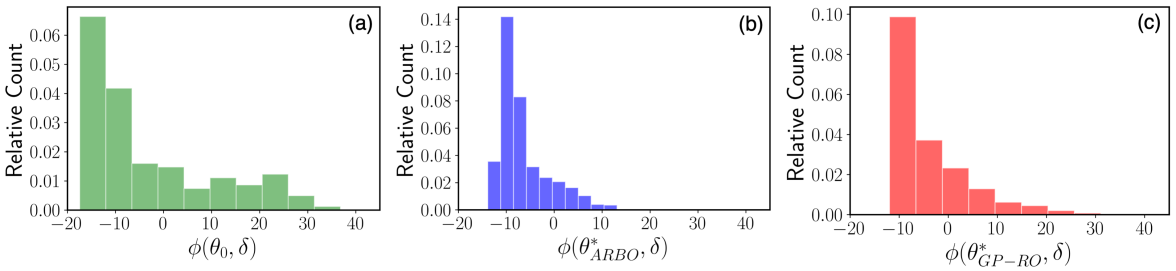

Figure 4.4: Histograms of the total cost $\phi(\theta, \delta)$ given in (4.29) established under 100 realizations of the parametric uncertainties in the plant simulator, for the cases of NMPC with no-backoff, (θ_0), NMPC auto-tuned with ARBO, (θ^*_{ARBO}), and NMPC auto-tuned with GP-RO,(θ^*_{GP-RO})

The resulting histograms are shown in Figure 4.4. We show the relative count since 150 samples have been used for the nominal case, while the BO-based plots use 750 samples (150 samples for 5 trials). The comparison with the nominal setting, in which the backoffs are set to zero, is provided to better highlight the advantages of auto-tuning. As seen from Figure 4.4, the estimated worst-case total cost for ARBO is significantly lower than that for the no-backoff case and GP-RO. This trend is also followed across the entire distributions as, on average, ARBO discovers better solutions than the nominal and GP-RO cases. It is important, though, to note that the algorithm is designed to discover the robust worst case solution; the fact that this solution also leads to an overall improvement in performance in a wide range of uncertainty values is a desired, and intuitively, expected side-effect. We also observe that several solutions of the nominal case actually lead to lower costs than ARBO and GP-RO. This is expected, again, based on minimax nature of the problem as well as given that backoffs overall restrict the performance by reducing the feasible region. They are important, though, as seen, for "protecting" against the worst-case scenario. Quantifying the worst-case closed-loop behavior of the system, the following results have been obtained, as seen in Table 4.2.

Table 4.2: Performance comparison between ARBO and GP-RO across trials

Trial	Worst-Case Cost		Final Recommended θ		Final Worst δ	
	ARBO	GP-RO	ARBO	GP-RO	ARBO	GP-RO
1	13.11	11.98	$(1 \times 10^{-1}, 7.8 \times 10^{-2})$	$(0, 5.4 \times 10^{-2})$	(0.75,1.25)	(0.75,1.75)
2	11.65	41.28	$(0, 6.5 \times 10^{-2})$	$(0, 1 \times 10^{-1})$	(1.35,1.25)	(1.35,1.25)
3	13.09	22.86	$(5.6 \times 10^{-2}, 5 \times 10^{-2})$	$(1 \times 10^{-1}, 1 \times 10^{-1})$	(1.33,1.56)	(1.35,1.25)
4	12.44	19.19	$(0 \times 10^{-1}, 7.2 \times 10^{-2})$	$(6.2 \times 10^{-3}, 8.6 \times 10^{-2})$	(0.75,1.75)	(1.17,1.66)
5	12.13	12.73	$(1.4 \times 10^{-2}, 3.8 \times 10^{-2})$	$(2 \times 10^{-2}, 6.8 \times 10^{-2})$	(1.35,1.25)	(1.35,1.75)

As a point of reference, we should mention that the worst-case cost for the nominal solution is 36.57. From Table 4.2 we can observe that the worst case cost of ARBO is

consistently lower than the nominal and GP-RO cases, except trial 1 where the costs are comparable, although the final recommended point is quite different. Although the GP-RO algorithm provides significantly sub-optimal results compared to ARBO, introducing some backoff generally improves the worst-case performance due to enchanced constraint satisfaction. However, in trial 2, the GP-RO solution is not able to encounter a candidate worst case solution that leads to a cost even higher than the nominal case. Note also that, in the case of ARBO, the values of the worst case costs are quite similar, even when the final recommended θ are not very close in the Θ space. This highlights high degrees of nonlinearity of the performance function in the joint design-uncertainty space.

To obtain more insight into effect of introducing backoff parameters, the dynamic closed-loop behavior of the system is computed. Figure 4.5 shows the closed-loop state profiles for the three cases considered in Figure 4.4. For ARBO we plot the profiles corresponding to trial 1, while for GP-RO the profiles corresponding to trial 2, which are the respective trial for which the algorithms yield the higher worst-case cost.

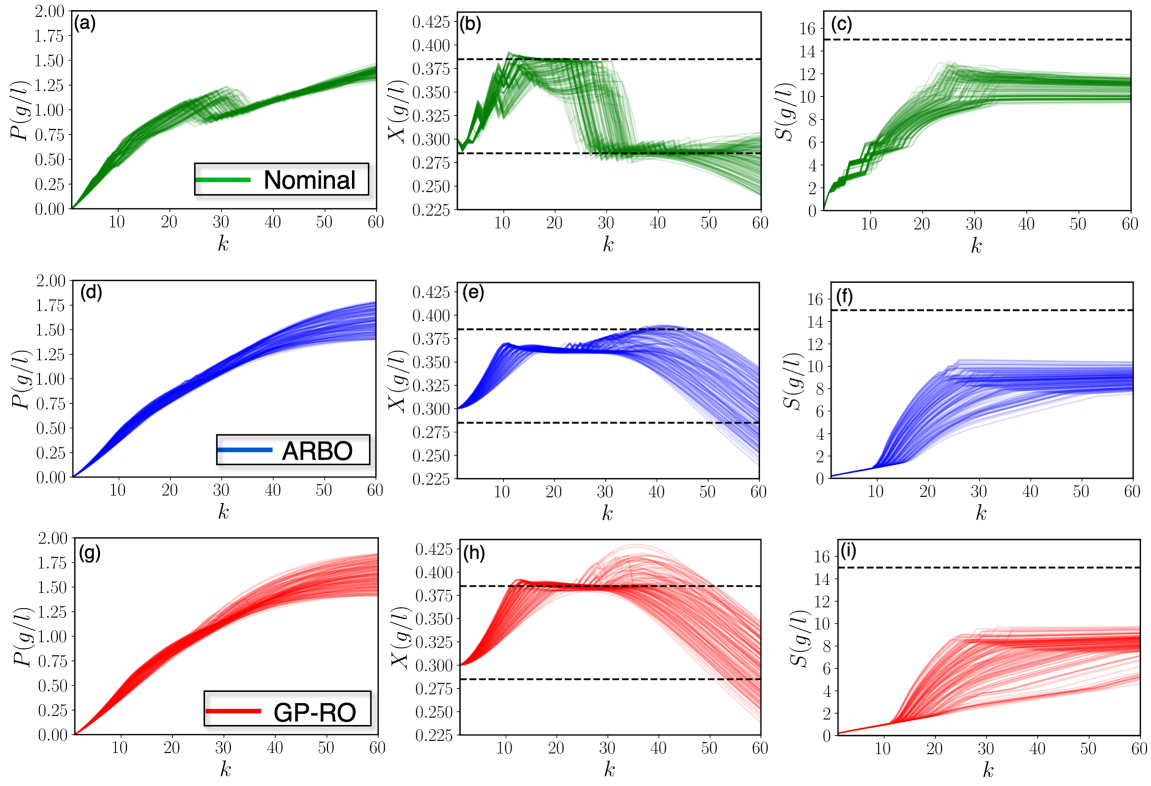

Figure 4.5: Closed-loop state profiles for the biomass, given for 150 realizations of parametric uncertainties in the plant simulator for the cases of NMPC with no-backoff (a-c), NMPC auto-tuned with ARBO (d-f), and NMPC auto-tuned with GP-RO (g-i).

From Figure 4.5(a), it is evident that in the no-backoff case, a large violation of X_{LB} occurs at later instances, thus incurring a large penalty in the performance cost function. This issue is alleviated by using the optimal backoffs calculated by ARBO, which lead to significantly less violation of the lower biomass bound X_{LB}, while not sacrificing on the

amount of product obtained throughout the processing time of interest, as seen by the concentrations of product in Figure 4.5(c & f), which is even higher in the ARBO case. It is important to note that achieving this behavior requires careful *simultaneous* tuning of θ_1 and θ_2, as we need to allow for some level of X_{LB} as well as X_{UB} violation to obtain a large enough amount of product, while staying within the desired bounds for biomass concentration. It would be difficult for a practitioner to infer this careful balance without running an impractically large number of closed-loop simulations. In fact, ARBO was able to uncover this desired balance in an automated fashion using 65 total simulations (15 initial simulations the GP construction and 50 iterations for BO); this is fewer than the 150 simulations we used to estimate the worst-case performance for the final recommended point. Lastly, we also observe that GP-RO attempts to improve the closed-loop behavior of the system by decreasing the amount of X_{LB} violations as ARBO, however, it does so by introducing also more significant X_{UB} violations.

4.6 Conclusions and Future Work

We have presented a robust BO method for auto-tuning of arbitrary complex control structures using a "high-fidelity" plant simulator with significant time-invariant uncertainties. The proposed ARBO method uses a probabilistic Gaussian process surrogate model to jointly describe the effect of the tuning parameters and plant model uncertainties on the closed-loop performance. The Gaussian process model allows for using an alternating confidence-bound procedure to simultaneously select the next candidate tuning and uncertainty realizations. As such, ARBO requires only one (expensive) closed-loop simulation in each iteration, as compared to alternative robust BO approaches to auto-tuning that rely on vastly more closed-loop simulations in each iteration. Our results on two simulation case studies demonstrate the advantages of the confidence bound-based procedure of ARBO in systematically realizing a tradeoff between the exploration and exploitation of the design-uncertainty space relative to Gaussian process surrogate-based robust optimization that lacks an exploration mechanism.

As a remark, we note that there are a series of challenges regarding the applicability of ARBO to similar tuning problems. An important issue is related to the challenges associated with BO in high-dimensional spaces. Since ARBO relies on GPs that jointly learn the decision variable and uncertainty space, it is expected to encounter difficulties in systems that either have many uncertainties or many tunable parameters (or a combination of both). A second challenge is related to the implementation of ARBO and, more specifically, the selection of optimization strategies that solve the inner/outer problemsm including the recommendation procedure. The nested form of the problem makes it critical to ensure that the aforementioned inner/outer problems are solved near global optimality. In the manuscript we propose several practical implementation points, but improvements can be made in this directions. Finally, other practical aspects such as an optimal selection strategy for the exploration constant and convergence criteria should be further examined. Our future work will mainly focus on addressing these challenges, as well as using more complex non-Gaussian noise models for robust BO.

Chapter 5

Gradient-Enhanced Bayesian Optimization Towards Efficient Closed-Loop Policy Search

This chapter discusses the incorporation of gradient observations into Bayesian Optimization (BO). Recent work has shown that BO can be augmented with gradient measurements to further improve its convergence behavior. These approaches mostly rely on standard acquisition functions and indirectly incorporate gradient information into a probabilistic surrogate model of the performance function to improve its local predictions. Here, we present new strategies to simultaneously exploit performance (zeroth-order) and gradient (first-order) data within a single acquisition optimization step. We propose two methods: (i) an acquisition ensemble approach, termed AEGBO, that suggests a point to query by jointly optimizing a zeroth-order and a first-order acquisition and (ii) an approach that enforces the necessary optimality conditions as constraints to the acquisition optimization, termed NOBO. The performance of the proposed methods is demonstrated for closed-loop policy search via reinforcement learning on a benchmark linear-quadratic-regulator (LQR) problem. This chapter is adapted from [155] and [156]. .

5.1 Introdcution

The control of complex systems is often associated with the challenge of optimizing black-box functions that are expensive to evaluate and lack an analytical, closed-form structure. These functions may also be subject to noise, further complicating their optimization. Thus, in many real-world applications, we resort to derivative-free global optimization techniques that can effectively handle these challenges. In recent years, there has been a growing interest in the use of black-box optimization methods for various control applications. In particular, Bayesian optimization (BO) [80, 34] has emerged as an effective strategy for generic controller auto-tuning [197, 159, 124], and direct policy-search reinforcement learning (RL) [178, 139]. The main idea of BO is to convert a challenging black-box optimization problem into a sequence of easier-to-solve sub-problems that aim at iteratively learning and updating our belief about the objective by querying the system performance. This is

achieved via constructing a Gaussian Process (GP) model of the objective given the current set of observations and subsequently optimizing over a utility metric, a so-called acquisition function (AF), to determine where to query the system next. AFs leverage the surrogate objective to suggest new evaluation points, balancing the competing aims of exploration and exploitation.

Although BO is by nature a zeroth-order optimization method, recent work has demonstrated that gradient information, when accessible in practice, can be valuable since it provides additional information about the objective function [311, 308, 253], leading to the development of various, so-called, gradient-enhanced BO. Conditioning the predictions of the function on gradient observations allows for reducing the variance in unexplored points in the domain, yielding a more accurate surrogate, hence accelerating the overall convergence of the algorithm. This gradient-enhanced GP can then be utilized with typical zeroth-order AFs [311] or with first-order AFs [178, 205].

Here, we present a framework for a gradient-enhanced BO method that directly incorporates the gradient information into the AF by leveraging the fact that the first-order optimality conditions must hold for the optimal solution of a BO problem. The first method we propose towards this goal is the use of an ensemble of two acquisition functions. that can be simultaneously maximized (in a Pareto sense) using multi-objective optimization. We initially derive a cheap-to-evaluate gradient-based acquisition function that can identify stationary points of the performance optimization problem. Subsequently, we propose a simple, yet effective strategy for finding query points that optimally tradeoff between a zeroth-order AF and the proposed gradient-based AF via multi-objective optimization. Thus, the proposed acquisition-ensemble, gradient-enhanced BO (AEGBO) method [155] can discover a set of valuable query points that are Pareto optimal with respect to both sources of information. Albeit it's promising results, a few issues with that approach are that (i) it is based on multi-objective optimization which is a challenging problem, (ii) it is harder to generalize to black-box constraint settings and, perhaps most importantly, (iii) it is not evident how to select a point to query from the resulting Pareto set at each iteration, hence, problem specific heuristics need to be developed while rendering convergence analysis cumbersome. To this end, we alleviate seveal issues by also proposing a necessary-optimality BO [156], or NOBO, method (second method) that uses GP surrogates for the partial derivatives of the objective to approximately enforce the first-order optimality conditions as black-box constraints in the AF. These constraints allow for defining a feasible set that explicitly takes into account the uncertainty present in approximating the partial gradients from data, which is updated with observing new data. Thus, the feasible set enables narrowing down the search of the design space to regions that are jointly informative with respect to both zeroth- and first-order information. Unlike AEGBO that relies on an ensemble of acquisition functions with first-order information, the performance of NOBO only depends on scalar exploration hyperparameters that are easier to select. We further analyze the theoretical performance of NOBO based on the cumulative regret metric, connecting it to the kernel properties of the GP. We demonstrate the performance of AEGBO and NOBO for policy-based RL on a benchmark LQR problem, in comparison with standard BO and REINFORCE.

5.2 Problem Statement

Optimization goal and regularity assumptions

We consider the following "black-box" optimization problem

$$\max_{x \in X} f(x), \tag{5.1}$$

where $x \in X$ are decision variables that are restricted to some known compact domain $X \subset \mathbb{R}^d$ and $f : X \to \mathbb{R}$ is an expensive-to-evaluate objective function whose mathematical structure is unknown. We assume that X can be expressed as the level set of a known function $c : \mathbb{R}^d \to \mathbb{R}^c$, i.e.,

$$X = \{x \in \mathbb{R}^d : c(x) \leq 0\}. \tag{5.2}$$

We consider the bandit feedback setting wherein, at iteration t, a query point x_t is selected for which noisy evaluations of $f(x_t)$ and its gradient $\nabla f(x_t)$ can be observed. That is, we observe

$$y_{0,t} = f(x_t) + \epsilon_{0,t}, \tag{5.3a}$$

$$y_{i,t} = \partial_{x_i} f(x_t) + \epsilon_{i,t}, \quad \forall i \in \mathbb{N}_1^d, \tag{5.3b}$$

where $\epsilon_{i,t}$ are i.i.d. R-sub-Gaussian noise terms for a fixed constant $R \geq 0$, meaning they must satisfy

$$\mathbb{E}\left\{e^{\lambda \epsilon_{i,t}} \mid \Sigma_{i,t-1}\right\} \leq e^{\frac{\lambda^2 R^2}{2}}, \ \forall i \in \mathbb{N}_0^d, t \geq 0, \lambda \in \mathbb{R}, \tag{5.4}$$

where $\Sigma_{i,t-1}$ denotes the σ-algebra generated by the random variables $\{x_k, \epsilon_{i,k}\}_{k=1}^{t-1}$ and x_t. This is a standard assumption in the bandit feedback setting, and is relatively mild since it holds for all distributions bounded in $[-R, R]$ [49]. We note that this differs from traditional bandit feedback problems that only assume the availability of zeroth-order information $y_{0,t}$, which can place a limitation on performance. Here, we look to incorporate gradient information, which can often be observed (or estimated) in control applications such as closed-loop policy optimization, as discussed in Section 5.6. We focus on functions f that satisfy the following common "regularity" assumption.

Assumption 2 *Let $H_k(X)$ denote the reproducing kernel Hilbert space (RKHS) of functions $X \to \mathbb{R}$, with a positive semi-definite kernel function $k : X \times X \to \mathbb{R}$. Furthermore, let $\langle \cdot, \cdot \rangle_k$ denote the inner product that obeys the reproducing property $f(x) = \langle f, k(x, \cdot) \rangle_k$ for all $f \in H_k(X)$, which induces the RKHS norm $\|f\|_k = \sqrt{\langle f, f \rangle_k}$. We assume that $\|f\|_{k_0} \leq B_0$ and $\|\partial_{x_i} f\|_{k_i} \leq B_i$ for all $i \in \mathbb{N}_1^d$ have known finite bounds $B_0, \ldots, B_d$ for some known kernels $k_0, \ldots, k_d$.*

Assumption 2 allows for the construction of well-behaved confidence bounds on the target functions and is valid as long as $(f, \nabla f)$ satisfy basic properties such as being bounded, continuous, and at least once differentiable over X. In the absence of such an assumption, these functions could be arbitrarily complex (e.g., realization of a white noise sequence) for which it may be difficult or impossible to exploit past data to make future selections. Furthermore, we assume that the kernel functions are normalized.

Assumption 3 *The kernel functions $k_0, \ldots, k_d$ for f and ∇f satisfy $k_i(x, x) \leq 1$ for all $x \in X$ and $i \in \mathbb{N}_0^d$.*

This assumption can be easily satisfied for any kernel with bounded variance by dividing by a scaling factor and is useful for making the theoretical bounds well-established.

Gaussian process models with gradient information

We place a GP prior over f to build a probabilistic surrogate model that is non-parametric. A GP model is fully specified by its mean function $\mu : X \to \mathbb{R}$ and covariance (or kernel) function $k : X \times X \to \mathbb{R}$. Since the gradient is a linear operator, the gradient of a GP must remain a GP, such that we can create a joint GP model with the following updated mean function $\tilde{\mu}$ and covariance function $\tilde{k}$

$$\tilde{\mu}(x) = \begin{bmatrix} \mu(x) \\ \nabla\mu(x) \end{bmatrix}, \tag{5.5a}$$

$$\tilde{k}(x, x') = \begin{bmatrix} k(x, x') & \nabla_{x'} k(x, x')^\top \\ \nabla_x k(x, x') & \nabla_x (\nabla_{x'} k(x, x')^\top) \end{bmatrix}. \tag{5.5b}$$

The extended mean function $\tilde{\mu} : X \to \mathbb{R}^{d+1}$ maps to a $(d+1)$-dimensional vector, while the extended covariance function $\tilde{k} : X \times X \to \mathbb{R}^{d+1 \times d+1}$ maps to a $(d+1) \times (d+1)$ matrix, which has the ability to capture correlation between the function and its d partial derivatives that make up the gradient vector [305, 311].

The idea behind using such a joint kernel function is that it captures the correlation between the function and the partial derivatives by specific selection of the kernel structure, following the differentiation properties of GPs. Nevertheless, in certain cases, the observed gradients are very noisy or prone to errors in the estimation process. When incorporating such gradient observations into the gradient enhanced kernel, it is possible that they may adversely affect the accuracy of the GP model by introducing misleading information about the true objective function. To mitigate this issue, as discussed in [205], we can create separate GP models for the objective function and each partial derivative. We consider a GP prior $\mathcal{GP}(0, k_i(\cdot, \cdot))$ over the target function f and its partial derivatives $\partial_{x_i} f$ to learn the unknown black-box functions, where k_i is the kernel function associated with the RKHS $H_{k_i}(X)$ (see Assumption 2). Additionally, we adopt an i.i.d. Gaussian zero-mean noise model with variance $\eta_i > 0$. The GP model of f enables us to construct analytic expressions for the posterior mean and covariance functions, as well as the maximum information gain in the bandit feedback problem at hand, that will be useful for the ensuing theoretical analysis.

Given t observations $\boldsymbol{y}_{i,t} = (y_{i,1}, \ldots, y_{i,t})$ under the GP prior, the posterior remains a GP with the following mean $\mu_{i,t}$, kernel $k_{i,t}$, and variance $\sigma_{i,t}^2$ functions for all $i \in \mathbb{N}_0^d$

$$\mu_{i,t}(x) = \boldsymbol{k}_{i,t}^\top(x)(\boldsymbol{K}_{i,t} + \eta_i I)^{-1} \boldsymbol{y}_{i,t}, \tag{5.6a}$$

$$k_{i,t}(x, x') = k_i(x, x') - \boldsymbol{k}_{i,t}^\top(x)(\boldsymbol{K}_{i,t} + \eta_i I)^{-1} \boldsymbol{k}_{i,t}(x'),$$

$$\sigma_{i,t}^2(x) = k_{i,t}(x, x), \tag{5.6b}$$

where $\boldsymbol{k}_{i,t}(x) = [k_i(x_1, x), \ldots, k_i(x_t, x)]^\top$ and $\boldsymbol{K}_{i,t}$ is the positive definite kernel matrix whose elements are given by $[\boldsymbol{K}_{i,t}]_{n,m} = k_i(x_n, x_m)$ for all $n, m \in \mathbb{N}_1^t$. Note that, in principle, one

could replace this set of $d+1$ independent GP models with a joint GP model that captures the correlation between f and ∇f (e.g., see [205]). Here, we consider the case of independent GPs for the following reasons. First, it simplifies the analysis and the complexity of the model substantially. Second, the established results straightforwardly carry over to the joint GP case due to Slepian's comparison lemma [255]. Third, it provides more flexibility in the choice of the kernel structure for the derivative functions.

Next, we define the *maximum information gain* (MIG) for the unknown functions f and ∇f.

Definition 1 *Let $A \subset X$ denote any potential subset of points sampled from X. The maximum information gain for the $(i+1)^{th}$ element of $(f, \nabla f)$ for t noisy measurements is*

$$\gamma_{i,t} = \max_{A \subset X : |A| = t} \frac{1}{2} \log \det \left(I + \eta_i^{-1} \boldsymbol{K}_{i,A} \right), \tag{5.7}$$

where $\boldsymbol{K}_{i,A} = [k_i(x, x')]_{x,x' \in A}$.

Note $\gamma_{i,t}$ depends on both the domain X and the kernel function k_i, and can be interpreted as a measure for the difficulty of the optimization task. Several results exist for bounding the growth of $\gamma_{i,t}$ as a function of the number of iterations t, as used in the theoretical analysis of Section 5.4.

We now summarize a key result that shows how the posterior GP mean is centered around the unknown functions by a multiplicative factor of the posterior standard deviation.

Lemma 2 (Theorem 2, [49]) *Let $X \subset \mathbb{R}^d$, $\{\epsilon_{i,t}\}_{t=1}^{\infty}$ be R-sub-Gaussian noise, and Assumption 2 holds. Then, for any $\delta \in (0,1)$, the following holds for all $x \in X$ and $t \geq 1$*

$$\begin{aligned} &|\mu_{i,t-1}(x) - g_i(x)| \\ &\leq \left(B_i + R\sqrt{2(\gamma_{i,t-1} + 1 + \ln((d+1)/\delta))} \right) \sigma_{i,t-1}(x), \end{aligned} \tag{5.8}$$

with probability at least $1 - \delta/(d+1)$, where g_i denotes the $(i+1)^{th}$ element of $(f, \nabla f)$ and $\mu_{i,t-1}(x)$, $\sigma_{i,t-1}(x)$, and $\gamma_{i,t-1}$ are given in (5.6) and (5.7).

Note that the value of δ in [49, Theorem 2] is replaced by $\delta/(d+1)$ above since we will require joint confidence bounds on $(f, \nabla f)$, similar to [314].

Performance metrics

We now define the key performance metrics that will be used to analyze effectiveness of the proposed approach. As in the standard bandit feedback setting, we look to minimize the gap of $f(x_t)$ to the optimal value $f^\star = \max_{x \in X} f(x)$, i.e., the *instantaneous regret*

$$r_t = f^\star - f(x_t), \tag{5.9}$$

where x_t is the selected query point at iteration $t \geq 1$. Given that gradient information is available, we can also quantify

$$v_t = \|\nabla f(x_t) - \nabla c(x_t) \lambda_t\|_1, \tag{5.10}$$

where $\lambda_t \in \mathbb{R}^c_+$ will be Lagrange multipliers selected by our approach at iteration t. As shown in Section 5.4, v_t is the distance from a first-order stationarity condition being satisfied. Ideally, our approach would be able to achieve zero regret and violation in a single step; however, this is only possible when $(f, \nabla f)$ are perfectly known. In the black-box setting, we aim to minimize the cumulative regret over a time horizon T

$$R_T = \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} \left(f^\star - f(x_t)\right). \tag{5.11}$$

Formally, minimizing R_T requires one to solve an intractable dynamic programming problem (see, e.g., [202, 21]). Thus, this paper presents an efficient, simple-to-implement *no-regret* approach that ensures $R_T/T \to 0$ as $T \to \infty$. The no-regret property not only guarantees vanishing per-round instantaneous regret, but also ensures convergence to the global solution. In a similar fashion, we also define the cumulative violation of stationarity as

$$V_T = \sum_{t=1}^{T} v_t = \sum_{t=1}^{T} \left(\|\nabla f(x_t) - \nabla c(x_t)\lambda_t\|_1\right). \tag{5.12}$$

5.3 Acquisition-Ensemble Gradient-Enhanced Bayesian Optimization (AEGBO)

In this section we describe the first proposed method that is based on the interplay of utilizing zeroth-order and first-order information during the querying stage. Given the probabilistic surrogate model and posteriors as defined in (5.6), we must define an acquisition function $\alpha(x) : X \to \mathbb{R}$ to provide a good measure of the (expected) desirability of querying any point $x \in X$ with respect to our end goal (maximizing the unknown function f). If properly selected, one would like to preferentially sample at the point that produces the highest value of the acquisition function. We can then formally define BO as the sequential learning process of selecting next samples in the following fashion

$$x_{(t+1)} \in \underset{x \in X}{\operatorname{argmax}}\, \alpha(x), \tag{5.13}$$

where $\alpha(\cdot)$ represents the acquisition function induced by the posterior conditioned on data observed up to iteration t, $\mathcal{D}^{(t)}$. Therefore, the main distinction between traditional BO and gradient-enhanced BO is that $\mathcal{D}^{(t)}$ includes derivative information for the latter, which necessitates the use of a more complex GP model. In principle, one could take advantage of any of the previously developed acquisition functions [80], such as expected improvement (EI), upper confidence bound (UCB), or knowledge gradient (KG), by replacing the standard posterior mean and variance predictions for f with those derived from (5.5a). Nevertheless, several computational aspects, tied with the acquisition optimization, reinforce our selection for using separate GPs for each of the objective and partial derivatives; performing hyperparameter training and posterior update using (5.5a) can be computationally demanding. In particular, inverting the covariance matrix $\tilde{\boldsymbol{K}}_t$ for the joint GP model scales as $O((t(d+1))^3)$, which can be challenging when either t or d are large in size. Furthermore, this additional cost can have a big impact on the effort needed to solve (5.13), which requires repeated forward predictions to be made with the joint GP model. To better understand the computational implications, let us discuss the derivative-enabled KG (dKG) function, which is

mathematically defined in [311]. The dKG function is defined as follows

$$\mathrm{dKG}_t(x) = \mathbb{E}_n \left[\max_{x \in X} \tilde{\mu}_{0,(t+1)}(x) | x_{(t+1)} = x \right] - \max_{x \in \mathbb{X}} \tilde{\mu}_{0,t}(x) \tag{5.14}$$

where $\tilde{\mu}_{0,t}(x) = \mathbb{E}_t\{f(x)\}$ is the expectation of $f(x)$ under the GP prior conditioned on data $\mathcal{D}^{(t)}$ (corresponds to the first element of the joint mean vector $\tilde{\mu}_{(T)}(x)$). dKG measures the expected improvement in the maximum value of the mean function given a new observation is taken at $x_{(t+1)} = x$. The use of the mean function, as opposed to the function observations themselves, is needed to filter out any noise present in the observations. Although dKG is a quite effective measure of the value of information, it is very expensive to evaluate due to the internal maximization over the future posterior mean function. To mitigate this computational burden, [311] proposed to only use the best directional derivative at each iteration. In addition to ignoring useful information in the form of the complete set of partial derivatives of the objective function, this approach does not fully address the inherent challenge of the two-level optimization procedure needed to globally solve (5.13) when $\alpha^{(t)}(x) = \mathrm{dKG}_t(x)$.

We now describe the proposed method for efficiently integrating noisy function and gradient information into the BO framework, referred to as AEGBO.

Proposed Acqusition Function

Here, we focus on UCB-style acquisition functions due to their simplicity and established convergence properties [149]. The UCB function is given by

$$\alpha^{(t)}_{\mathrm{UCB}}(x) = \mu_{0,t}(x) + \beta_f \sigma_{0,t}(x), \tag{5.15}$$

where $\beta_f \in \mathbb{R}_+$ is a hyperparameter that balances exploration and exploitation, and $\sigma_{0,t}(x) = [k_{0,t}(x,x)]^{1/2}$ is the standard deviation of the posterior GP for f. Under the independence assumption, the gradient predictions do not directly impact the UCB acquisition such that we need a new strategy for quantifying the value of gradient information. To derive an independent source of information, we recognize that a necessary condition for optimality in (5.1) is $\nabla f(x) = 0$ (assuming the global maximum lies in the interior of X). An equivalent way to represent the solutions to this set of equations is $\min_{x \in X} \|\nabla f(x)\|$, which can also be stated as $\max_{x \in X}(-\|\nabla f(x)\|)$, where $\|\cdot\|$ denotes some vector norm; here, we use the 1-norm. Since the gradient is also an unknown function, we can use BO methods to tackle this optimization problem as a way to efficiently search for stationary points of the original maximization problem (5.1). An important distinction between the gradient norm (GN) problem and (5.1) is that the former involves multiple unknown functions. This is often referred to as a decomposed BO problem, for which standard acquisition functions do not directly apply. We can straightforwardly develop an UCB acquisition function for multi-output problems whenever the objective is defined as a linear transformation of the GP models, as shown in [132]. Norms are nonlinear operators, however, we need a tailored approximation strategy for the gradient norm. We propose the following gradient-based acquisition function analogously to the UCB function (5.15)

$$\alpha^{(t)}_{\mathrm{GN}}(x) = -\mathbb{E}_t\{\|\nabla f(x)\|\} + \beta_g \sqrt{\mathrm{Var}_t\{\|\nabla f(x)\|\}}, \tag{5.16}$$

where $\mathrm{Var}_t\{\cdot\}$ denotes the posterior variance given $\mathcal{D}^(t)$ and $\beta_g \in \mathbb{R}_+$ is a hyperparmeter similar to β_f. We can construct analytic expressions for the mean and variance terms since the absolute value of each partial derivative follows a folded normal distribution [289]. Starting with the mean term, we can derive

$$\begin{aligned}\mathbb{E}_t\{\|\nabla f(x)\|\} &= \textstyle\sum_{i=1}^{d} \mathbb{E}_t\left\{\left|\frac{\partial f(x)}{\partial x_i}\right|\right\}, \\ &= \sum_{i=1}^{d}\left\{2\sigma_{i,t}(x)\phi(z_i^{(t)}) + \mu_{i,t}(x)\left[\Phi(z_i^{(t)}) + \Phi(z_i^{(t)})\right]\right\},\end{aligned} \tag{5.17}$$

where $z_i^{(t)} = \mu_{i,t}(x)/\sigma_{i,t}(x)$, and $\phi(\cdot)$ and $\Phi(\cdot)$ correspond to the standard normal density function and cumulative density function, respectively. We can similarly derive a simple overall expression for the variance

$$\begin{aligned}\mathrm{Var}_t\{\|\nabla f(x)\| &= \textstyle\sum_{i=1}^{d} \mathrm{Var}_t\left\{\left|\frac{\partial f(x)}{\partial x_i}\right|\right\}, \\ &= \sum_{i=1}^{d}\left\{(\mu_{i,t}(x))^2 + (\sigma_{i,t}(x))^2 - \mathbb{E}_t\left\{\left|\frac{\partial f(x)}{\partial x_i}\right|\right\}^2\right\}.\end{aligned} \tag{5.18}$$

Note that closed-form expressions for the inner expectation terms have already been computed in (5.17). As such, our proposed gradient-based acquisition function in (5.16) can be efficiently computed using the d separate GP models for each of the partial derivatives of the objective function. This implies that maximizing $\alpha_{\mathrm{GN}}^{(t)}(x)$ should be at worst a linear factor of the cost required to maximize the cheap-to-evaluate function $\alpha_{\mathrm{UCB}}^{(t)}(x)$ with respect to d. This is a substantial reduction in cost when compared to dKG.

Combining Function and Gradient Information using Acquisition Ensembles

Now, we are equipped with two separate acquisition functions $\alpha_{\mathrm{UCB}}^{(t)}(x)$ and $\alpha_{\mathrm{GN}}^{(t)}(x)$ that, respectively, provide independent sources of zeroth- and first-order information regarding the maxima of f. It is unlikely that the same point maximizes both of these functions simultaneously, meaning we need some procedure to select a common value x_{t+1} that performs reasonably well with respect to both functions. The multi-objective optimization (MOO) framework is suitable for this task since it allows us to systematically tradeoff between multiple objectives.

The main goal of MOO is to characterize the set of points on the so-called *Pareto frontier*, which is the set of Pareto optimal points, i.e., feasible points $x \in X$ in which favorable movement in one objective comes at the expense of at least one other objective. In [46], a related idea is applied to a set of standard BO acquisition functions that showed promising results. Therefore, we look to develop a similar approach using $\boldsymbol{\alpha}_t(x) = \{\alpha_1^{(t)}(x), \alpha_2^{(t)}(x)\}$ as our set of acquisition functions, where the subscripts 1 and 2 will be used as shorthand for the UCB and GN acquisition functions, respectively. We now formally present the AEGBO method in terms of $\boldsymbol{\alpha}_t(x)$ as the following sequential sampling process

$$x_{(t+1)} \in X_t^\star = \{x \in \mathbb{X} : \boldsymbol{\alpha}_t(x) \in \mathcal{P}_t\}, \tag{5.19}$$

where $X_t^\star$ denotes the set of Pareto optimal points given all currently available data $\mathcal{D}^{(t)}$, which is characterized by the Pareto frontier $\mathcal{P}_t$

$$\mathcal{P}_t = \{\boldsymbol{\alpha}_t(x) : \nexists y \in \mathbb{X} \text{ s.t. } \boldsymbol{\alpha}_t(x) \prec \boldsymbol{\alpha}_t(y)\}. \tag{5.20}$$

Here, $\boldsymbol{\alpha}_t(x) \prec \boldsymbol{\alpha}_t(y)$ implies point y *dominates* x, which occurs if and only if $\alpha_i^{(t)}(x) \leq \alpha_i^{(t)}(y)$ for all $i \in \{1, 2\}$ and $\exists i \in 1, 2$ such that $\alpha_i^{(t)}(x) < \alpha_i^{(t)}(y)$. Therefore, $\mathcal{P}_t$ corresponds to the set of points for which there does not exist any feasible point that dominates it.

Although the proposed AEGBO method requires the MOO problem (5.19) be solved at every iteration, this problem involves only two cheap-to-evaluate objective functions and, thus, can be straightforwardly solved (approximately) using established methods such as the NSGA-II algorithm [63]. It is worth noting that all points in $X_n^\star$ are Pareto optimal such that there is no clear metric to select between the candidate points in this set. In general, any selection criteria can be utilized. Uniform random selection criteria (in which all points from $X_n^\star$ are potentially chosen with equal probability) tend to reduce bias that may result from a deterministic selection strategy. Nevertheless, other heuristics such as initially selecting Pareto points that achieve the lowest gradient norm (in absolute value) may work well in practice, as used in this paper.

5.4 Necessary Optimality-constrained Bayesian Optimization (NOBO)

This section summarizes the proposed necessary optimality-constrained Bayesian optimization (NOBO) approach for solving (5.1). The key observation that motivates NOBO is that we can reformulate (5.1) as

$$\max_{x \in X} f(x) \;\; \text{s.t.} \;\; \nabla f(x) \in N_X(x), \tag{5.21}$$

where $N_X(x) = \{z \in \mathbb{R}^d : z^\top (y - x) \leq 0, \forall y \in X\}$ denotes the normal cone to the set X at the point x. The newly added constraint $\nabla f(x) \in N_X(x)$ implies x is a "stationary point," which constitutes the first-order necessary optimality conditions for x to be a (local) maximum as long as the set X ensures constraint qualifications are satisfied. At the first glance, (5.21) may not appear useful since the necessary optimality conditions are typically solved numerically to identify possible solutions to (5.1). This would make $\nabla f(x) \in N_X(x)$ redundant; however, this is only true when the function f is exactly known. In the black-box setting of this work, these constraints provide additional independent information that can be exploited to restrict the set of possible query points.

Assuming the linear independence constraint qualification (LICQ) holds, we can equivalently represent the feasible set of (5.21) using the Karush-Kuhn-Tucker (KKT) conditions

$$\mathcal{F} = \{x \mid \exists \lambda : \nabla f(x) = \nabla c(x)\lambda, \; 0 \leq \lambda \perp c(x) \leq 0\},$$

where the notation "$0 \leq \lambda \perp c(x) \leq 0$" is shorthand for the complementary constraints, i.e., $c(x) \leq 0$, $\lambda \geq 0$, $\lambda^\top c(x) = 0$. Since neither the target function f or the feasible set $\mathcal{F}$ are known in the black-box setting, we rely on constructing high probability relaxations using GP models. To this end, we introduce the lower and upper confidence bound functions.

Definition 2 *The lower and upper confidence bounds for the* $(i+1)^{th}$ *element of* $(f, \nabla f)$ *at iteration* t *are given by*

$$l_{i,t}(x) = \mu_{i,t-1}(x) - \beta_{i,t}^{1/2}\sigma_{i,t-1}(x), \tag{5.22a}$$

$$u_{i,t}(x) = \mu_{i,t-1}(x) + \beta_{i,t}^{1/2}\sigma_{i,t-1}(x), \tag{5.22b}$$

where $\beta_{i,t}^{1/2} = B_i + R\sqrt{2(\gamma_{i,t-1} + 1 + \ln((d+1)/\delta))}$.

Using Lemma 2, we can then establish the following result on the joint relaxation of (5.21).

Theorem 2 *Let the assumptions of Lemma 2 hold. Then, with probability at least* $1-\delta$, *the following bounds hold simultaneously for all* $x \in X$ *and* $t \geq 1$

$$f(x) \in [l_{0,t}(x), u_{0,t}(x)] \text{ and } \mathcal{F} \subseteq \mathcal{F}_t^u, \tag{5.23}$$

where $\mathcal{F}_t^u = \{x \mid \exists \lambda : (x, \lambda) \in \mathcal{R}_t^u\}$ *is the relaxed feasible region defined in terms of the set*

$$\mathcal{R}_t^u = \left\{ \begin{bmatrix} x \\ \lambda \end{bmatrix} \;\middle|\; \begin{array}{c} |\boldsymbol{\mu}_{d,t-1}(x) - \nabla c(x)\lambda| \leq \boldsymbol{\beta}_{d,t}^{1/2}\boldsymbol{\sigma}_{d,t-1}(x) \\ 0 \leq \lambda \perp c(x) \leq 0 \end{array} \right\},$$

with the following definitions

$$\begin{aligned} \boldsymbol{\mu}_{d,t-1}(x) &= (\mu_{1,t-1}(x), \ldots, \mu_{d,t-1}(x)) && \in \mathbb{R}^{d\times 1}, \\ \boldsymbol{\sigma}_{d,t-1}(x) &= (\sigma_{1,t-1}(x), \ldots, \sigma_{d,t-1}(x)) && \in \mathbb{R}^{d\times 1}, \\ \boldsymbol{\beta}_{d,t}^{1/2} &= diag(\beta_{1,t}^{1/2}, \ldots, \beta_{d,t}^{1/2}) && \in \mathbb{R}^{d\times d}. \end{aligned}$$

Proof: The confidence bounds $l_{i,t}(x)$ and $u_{i,t}(x)$ are random variables since they depend on observations $y_{i,t}$ that are corrupted by random noise. Therefore, we can define the following events that the unknown functions respect the confidence bounds for all $x \in X$ and $t \geq 1$

$$\begin{aligned} \mathcal{E}_0 &= \cap_{x\in X} \cap_{t\geq 1} \{l_{0,t}(x) \leq f(x) \leq u_{0,t}(x)\}, \\ \mathcal{E}_i &= \cap_{x\in X} \cap_{t\geq 1} \{l_{i,t}(x) \leq \partial_{x_i} f(x) \leq u_{i,t}(x)\}, \ \forall i \in \mathbb{N}_1^d. \end{aligned}$$

We can then establish the following sequence of inequalities

$$\begin{aligned} \mathbb{P}\left\{\cap_{i=0}^d \mathcal{E}_i\right\} &= 1 - \mathbb{P}\left\{\cup_{i=0}^d \overline{\mathcal{E}_i}\right\} \geq 1 - \textstyle\sum_{i=0}^d \mathbb{P}\left\{\overline{\mathcal{E}_i}\right\} \\ &\geq 1 - \textstyle\sum_{i=0}^d \frac{\delta}{d+1} = 1 - \delta, \end{aligned}$$

where the second inequality follows from Boole's inequality and the third inequality follows from Lemma 2. The first part of (5.23) directly follows. To see that $\mathcal{F} \subset \mathcal{F}_t^u$ must also hold, note that the stationarity condition $\nabla f(x) = \nabla c(x)\lambda$ can be represented by two inequalities $\nabla f(x) - \nabla c(x)\lambda \leq 0$ and $\nabla f(x) - \nabla c(x)\lambda \geq 0$, which can be relaxed by replacing the elements of ∇f by their lower and upper confidence bounds, respectively. After a few algebraic manipulations, one can derive $\mathcal{F}_t^u$ as an equivalent representation. ■

Algorithm 2 The relaxation-based **N**ecessary **O**ptimality-constrained **B**ayesian **O**ptimization (NOBO) algorithm

Input: The compact domain X; GP priors $(\mu_i, k_i)_{i=0}^d$, parameters $\{\beta_{i,t}\}_{i\in\mathbb{N}_0^d, t\geq 1}$; and total number of iterations T.

1: **for** $t = 1$ to T **do**
2: Solve $(x_t, \lambda_t) \in \operatorname{argmax}_{x,\lambda} u_{0,t}(x)$ s.t. $(x, \lambda) \in \mathcal{R}_t^u$.
3: Get noisy observations of f and ∇f at x_t.
4: Update GP posteriors (5.6) with new observations.
5: **end for**

We now state the proposed NOBO approach in Algorithm 2, which is conceptually straightforward in that only a single auxiliary problem is solved at each iteration. This auxiliary problem, shown in line 2, is an instance of a mathematical program with complementarity constraints (MPCC). Given the focus on expensive-to-evaluate functions f, we assume that the cost of solving the MPCC is small relative to the cost of a function query. Also, a direct consequence of Theorem 2 is that the set of global solutions $x^\star$ must be contained within $\mathcal{F}_t^u$ with probability at least $1-\delta$, so that the "size" of $\mathcal{F}_t^u$ provides a measure for progress of NOBO as t increases.

Remark 2 *When $x^\star$ is known to lie strictly in the interior of X, we can directly set $\lambda = 0$, which greatly simplifies the auxiliary optimization problem in line 4 of Algorithm 2. This is equivalent to further simplifying the necessary optimality conditions to $\nabla f(x) = 0$.*

Theoretical Analysis of NOBO

In this section, we analyze the theoretical performance of NOBO (Algorithm 2). Our goal is to establish bounds on the cumulative regret R_T and the stationarity violation V_T that depend on the MIG of the unknown functions and the number of iterations T. We can then use established bounds on the MIG in [267, Theorem 5] to bound the MIG growth over T, which will allow us to establish convergence of NOBO. First, a lemma is introduced to bound r_t and v_t.

Lemma 3 *If the inequalities* (5.23) *hold, then the auxiliary problem in line 4 of Algorithm 2 will always be feasible and the instantaneous regret and stationary violation will satisfy*

$$r_t \leq 2\beta_{0,t}^{1/2}\sigma_{0,t-1}(x_t), \tag{5.24a}$$

$$v_t \leq \textstyle\sum_{i=1}^d 2\beta_{i,t}^{1/2}\sigma_{i,t-1}(x_t), \tag{5.24b}$$

for all $x \in X$ and $t \geq 1$.

Proof: From (5.9) and x_t in line 4 of Algorithm 2, we have

$$\begin{aligned} r_t &\leq u_{0,t}(x^\star) - l_{0,t}(x_t) \\ &\leq u_{0,t}(x_t) - l_{0,t}(x_t) = 2\beta_{0,t}^{1/2}\sigma_{0,t-1}(x_t), \end{aligned}$$

where the first inequality follows from the assumed upper and lower bounds on the target function and the second inequality follows from the fact that x_t maximizes $u_{0,t}(x)$ over a set $\mathcal{F}_t^u \subseteq X$ that contains $x^\star$ under the assumed bounds on the gradient ∇f. Feasibility of $x^\star \in \mathcal{F}_t^u$ directly implies feasibility of the auxiliary problem. We now consider the violation of the stationarity condition. Let $q_t = \nabla c(x_t)\lambda_t \in \mathbb{R}^d$. We can rewrite (5.10) as

$$v_t = \textstyle\sum_{i=1}^d |\partial_{x_i} f(x_t) - [q_t]_i|.$$

We look to bound each element of this sum

$$\begin{aligned}
&|\partial_{x_i} f(x_t) - [q_t]_i| \\
&\leq |\partial_{x_i} f(x_t) - \mu_{i,t-1}(x_t)| + |\mu_{i,t-1}(x_t) - [q_t]_i| \\
&\leq \beta_{i,t}^{1/2}\sigma_{i,t-1}(x_t) + \beta_{i,t}^{1/2}\sigma_{i,t-1}(x_t) = 2\beta_{i,t}^{1/2}\sigma_{i,t-1}(x_t),
\end{aligned}$$

where the first inequality follows from $|a+b| \leq |a| + |b|$ and the second inequality follows from the assumed bounds in (5.23) and the fact that $(x_t, \lambda_t) \in \mathcal{R}_u$. ■

We can now combine these results, along with results from [314], to establish the main theorem on the cumulative regret and stationarity violation for NOBO.

Theorem 3 *Under the assumptions of Lemma 2, we have, with probability at least $1-\delta$, that the sample points $\{x_t\}_{t\geq 1}$ generated by NOBO (Algorithm 2) satisfy*

$$R_T \leq 4\beta_{0,T}^{1/2}\sqrt{(T+2)\gamma_{0,T}}, \tag{5.25a}$$

$$V_T \leq \textstyle\sum_{i=1}^d 4\beta_{i,T}^{1/2}\sqrt{(T+2)\gamma_{i,T}}. \tag{5.25b}$$

Proof: Combining Lemma 2 and Theorem 2, the following event must hold with probability $\geq 1-\delta$

$$\{r_t \leq 2\beta_{0,t}^{1/2}\sigma_{0,t-1}(x_t)\} \cup \{v_t \leq \textstyle\sum_{i=1}^d 2\beta_{i,t}^{1/2}\sigma_{i,t-1}(x_t)\}.$$

From the definition of cumulative regret, we can establish the following inequalities that must hold with probability $\geq 1-\delta$

$$\begin{aligned}
R_T = \textstyle\sum_{t=1}^T r_t &\leq 2\beta_{0,T}^{1/2}\textstyle\sum_{t=1}^T \sigma_{0,t-1}(x_t) \\
&\leq 4\beta_{0,T}^{1/2}\sqrt{(T+2)\gamma_{0,T}},
\end{aligned}$$

where the first inequality follows from the monotonicity of $\{\beta_{0,t}\}_{t\geq 1}$ and the second inequality follows from [314, Lemma 4], which shows that $\sum_{t=1}^T \sigma_{0,t-1}(x_t) \leq \sqrt{4(T+2)\gamma_{0,T}}$. The stated result follows by applying the same analysis to the cumulative stationarity violation V_T. ■

The following corollary to Theorem 3 is immediately established for the convergence rate of NOBO to $f^\star$.

Corollary 1 *Under the assumptions of Theorem 2, we have, with probability at least $1-\delta$, that there exists some $\tilde{x}_T \in \{x_1, x_2, \ldots, x_T\}$ such that*

$$f^\star - f(\tilde{x}_T) \leq \frac{4\beta_{0,T}^{1/2}\sqrt{(T+2)\gamma_{0,T}}}{T} = \mathcal{O}\left(\frac{\gamma_{0,T}}{\sqrt{T}}\right). \tag{5.26}$$

Proof: Let $S_T = \min_{t\in\{1,\dots,T\}} r_t$ be the minimum of the regret sequence. Since r_t is non-negative and the minimum of a sequence must be less than or equal to the average, we have $0 \leq S_T \leq R_T/T$. The claim follows from Theorem 3 and letting $\tilde{x}_T$ be the point that minimizes S_T. ■

We note that the point $\tilde{x}_T$ cannot be identified by minimizing the regret sequence unless the noise variance is zero, i.e., $\eta_0 = 0$. In the noisy case, we can resort to the following recommendation procedure

$$\tilde{x}_T = \underset{x_t\in\{x_1,\dots,x_T\}}{\operatorname{argmax}} \; l_{0,t}(x_t), \tag{5.27}$$

which can be interpreted as a pessimistic estimate of the maximum value of f due to the noise in the observations. This will not affect the result shown in Corollary 1 since Theorem 3 holds for the pessimistic estimate of the regret $r_t \leq \bar{r}_t = f^\star - l_{0,t}(x_t)$. It is also interesting to note that this result implies that NOBO has at least the same worst-case convergence rate as the traditional zeroth-order GP-UCB algorithm. However, since we are optimizing over a restricted set $\mathcal{F}_t^u \subset X$, NOBO is expected to provide a faster convergence rate in practice.

Theorem 3 and Corollary 1 are given in terms of the MIG for general kernels. They imply convergence of $\tilde{x}_T \to x^\star$ as $T \to \infty$ as long as $\gamma_{0,T} = o(\sqrt{T})$, which can be guaranteed for the common types of kernels. This is summarized below.

Lemma 4 (Theorem 5, [267]) *Let X be compact and convex, $d \in \mathbb{N}$, and assume $k(x, x') \leq 1$. Then,*

- *Linear: $\gamma_T = \mathcal{O}(d \log T)$;*
- *Squared exponential: $\gamma_T = \mathcal{O}((\log T)^{d+1})$;*
- *Matern ($\nu > 1$): $\gamma_T = \mathcal{O}(T^{d(d+1)/(2\nu+d(d+1)}(\log T))$.*

Substituting the results of Lemma 4 into (5.26), it is evident that NOBO converges for the linear, squared exponential, and Matern kernel with smoothness parameter $\nu > 1$. These results can also be used to derive kernel specific bounds as a function of T, e.g., $f^\star - f(\tilde{x}_T) = \mathcal{O}((\log T)^{d+1}/\sqrt{T})$ for the squared exponential kernel.

5.5 Episodic Reinforcement Learning

Reinforcement learning (RL) is a semi-supervised learning method in which a so-called "agent" attempts to learn the best way to maximize a long-term reward function through trial-and-error interactions with the "environment." There has been a vast amount of work on RL, which can be roughly viewed as a collection of solution approaches to stochastic optimal control problems of the form

$$\begin{aligned} \max_{\pi_{0:N-1}} \quad & \mathbb{E}_{w_{0:N-1}} \left\{ \textstyle\sum_{k=0}^{N-1} r_k(z_k, u_k, w_k) + r_N(z_N) \right\}, \\ \text{s.t.} \quad & z_{k+1} = g_k(z_k, u_k, w_k), \quad u_k = \pi_k(\tau_k), \end{aligned} \tag{5.28}$$

when the system dynamics are unknown, where z_k, u_k, and w_k are the system state, control input, and disturbance at time k, respectively, $g_k(\cdot)$ is the state transition function that governs the dynamics at time k in response to stochastic disturbances w_k, $r_k(\cdot)$ is the reward gained at time step k, $\pi_k(\cdot)$ is the feedback control policy that can be any feasible function of the observed data trajectory up until time k, i.e., $\tau_k = (u_0, \ldots, u_{k-1}, x_0, \ldots, x_k)$, and N is the time horizon. Thus, RL methods attempt to solve (5.28) in cases where the state transition rules $\{g_k(\cdot)\}$ are unknown by transforming the problem into a learning task. We can then classify different RL methods based on the chosen learning task (see, e.g., [171]).

One of the most popular variants is the so-called class of policy-based RL methods that look to learn the optimal settings for a parametrized stochastic policy function $p(\tau; x)$ where x refers to adjustable policy parameters. We can think of the overall reward function in (5.28) as simply a function of x since this represents the only degrees of freedom remaining in the policy. Let us define $R(\tau)$ as the overall reward function computed over a single dynamic trajectory τ. Due to the exploration noise (or stochasticity) of the policy, τ is random with some probability distribution $p(\tau; x)$ that is parametrized by x such that

$$f(x) = \mathbb{E}_{p(\tau;x)}\{R(\tau)\} = \int R(\tau)p(\tau; x)d\tau, \tag{5.29}$$

matches our starting problem (5.1) since f is unknown. The first key observation is that noisy observations are critically important to handle in policy-based RL problems since we cannot evaluate the integral in (5.29) exactly and must resort to some sampling strategy, e.g., $\frac{1}{N_s}\sum_{i=1}^{N_s} R(\tau^{(i)})$ where $\tau^{(i)} \sim p(\tau; x)$. Traditional BO methods can be applied in such cases, however, this only takes advantage of zeroth-order information. Policy gradient methods are a commonly used alternative that exploit the fact that gradient estimates of the reward can be derived as follows

$$\nabla f(x) = \mathbb{E}_{p(\tau;x)}\{R(\tau)\nabla_x \log p(\tau; x)\}, \tag{5.30}$$

which can be evaluated using only gradients of the policy for Markov processes [276]. Traditional policy gradient methods, such as REINFORCE [307], then apply stochastic gradient ascent to update an initial $x^{(0)}$ using a mini-batch of samples, i.e.,

$$x_{(t+1)} = x_t + \frac{\eta_t}{N_s}\left(\sum_{i=1}^{N_s} R(\tau^{(i)})\nabla_x \log p(\tau^{(i)}; x)\right), \tag{5.31}$$

where η_t is the step size at iteration t (sometimes referred to as a learning rate). However, as observed from (5.31), these types of policy gradient methods only use estimates of the current gradient at each iteration, which neglects valuable information about the current and past reward and gradient estimates. For example, if the current reward value is small, we should prioritize moving to a new region of the policy parameter space, as opposed to wasting closed-loop evaluations at neighboring parameter values that are likely to perform poorly. An efficient sampling strategy is extremely important whenever the closed-loop data collection process is expensive, which will be the case when the system dynamics are defined in terms of a high-fidelity simulator or a time-consuming experimental setup.

The proposed AEGBO method in Section 5.3 is well-suited to take advantage of the complete history of reward and its gradient evaluations at every iteration. Therefore, we

can think of AEGBO as a powerful hybrid strategy that inherits the efficient global search capability of BO as well as the useful local search behavior of REINFORCE. It is worth noting that the cost of evaluating $\nabla_x \log p(\tau^{(i)}; x)$ is roughly the same as a single evaluation of the policy itself, meaning the gradient estimate can be obtained for free whenever the reward evaluation $R(\tau^{(i)})$ is much more expensive than the policy evaluation $p(\tau^{(i)}; x)$, which will often be the case for expensive systems.

5.6 Performance of Gradient-Enhanced BO on RL-based Policy Search

We compare AEGBO to two baseline algorithms on the LQR problem to demonstrate its performance improvements. Since our goal is to identify the policy parameters that maximize the reward function in as few iterations as possible, we use simple regret as our performance metric. `UQLab` [162] is used to construct the the GP surrogates with a zero prior mean and a squared-exponential kernel. To get an accurate estimate of the cost for each input, as well as to reduce the variance of the gradient estimates, a "mini-batch" size of $N_s = 2^8$ samples is used during each episode (training epoch). In addition, we use a baseline value in the estimate of the gradient as discussed in [230, 275].

First, we examine the performance of AEGBO and NOBO in comparison with traditional BO and REINFORCE [230], which is a commonly used reinforcement learning strategy.

BO: The sampled point is $x_{(t+1)} \in \text{argmax}_{x \in X} \alpha^{(t)}_{\text{UCB}}(x)$, which only considers zeroth-order information. We keep all other settings the same as that used in AEGBO.

REINFORCE: The REINFORCE algorithm corresponds to the stochastic gradient ascent update step shown in (5.31), which uses only local first-order information at every iteration. We set the learning rate $\eta_t = 0.1$, which is a commonly used default value (and is the same order as the exploration parameters used in BO and AEGBO).

The initial training dataset is composed of d points (here, $d = 4$) chosen uniformly at random from the design space X. We quantify the average closed-loop performance by repeating each algorithm 100 times for different initial datasets. For standard BO, AEGBO and NOBO, we use $\beta_{0,t} = 0.1 + 10^{-2} d \ln(1 + 10^{-2} t)$ for the acquisition objective. For AEGBO the exploration constant of the GN acquisition is set to $\beta_g = 0$. For the constraints in NOBO we set $\beta_{i,t} = 2 + 5 \times 10^{-2} d \ln(1 + 5 \times 10^{-2} t)$. The selection of the $\beta_{i,t}$ hyperparameters is based the fact that in practice a fixed value is used in BO, while this form allows for a slight increase in subsequent steps, capturing the main effects of d and t [197] and work well in practice. Nevertheless, the results were not found to be particularly sensitive to these choices.

Fig. 5.1 shows the estimated average simple regret $S_T = \min_{t \in \{1,\dots,T\}} r_t$ versus the number of episodes T for BO, REINFORCE, AEGBO and NOBO up to $T = 40$. REINFORCE initially shows a drop in regret but it quickly stalls at a relatively large value after only a few episodes. This can be attributed to fact that REINFORCE is only utilizing noisy gradient information to update parameters x. All of BO, AEGBO and NOBO quickly start to outperform REINFORCE and demonstrate continual improvement (as the number of episodes increases), which can be attributed to the valuable reward observations that

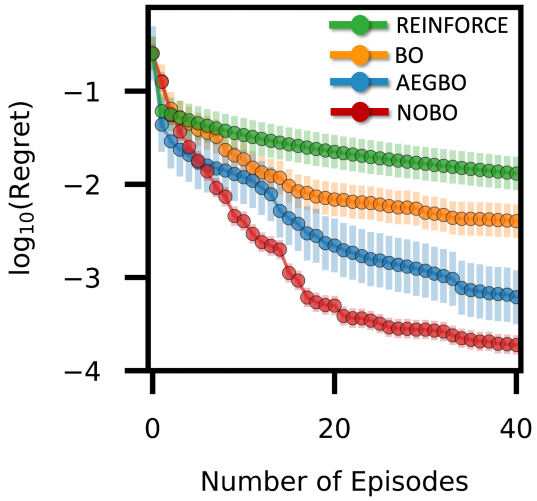

Figure 5.1: Simple regret (current best) for REINFORCE (green), BO (orange), AEGBO (blue) and NOBO (red) over 40 closed-loop episodes. Solid lines represent the average regret over 100 trials with different initial datasets. The shaded regions show the one standard deviation about the average regret.

are interpolated to globally predict the reward function. NOBO consistently outperforms REINFORCE, standard BO and AEGBO over all episodes by up to two orders of magnitude in simple regret, while both gradient-based methods outperform simple BO which suggests that the incorporation of the necessary optimality conditions into the search process leads to improved query point selection at every iteration. For NOBO specifically, it also serves as validation of Theorem 2 since the global solution was not eliminated from the estimated feasible region. It is also worth noting that the differences between NOBO and AEGBO can influenced by particular algorithmic settings, e.g., the heuristic used to select the Pareto point to query in AEGBO.

Pareto Front for AEGBO

To better understand the underlying source of AEGBO's improved performance, Fig. 5.2 shows the evolution of the Pareto frontier in (5.20) over different episodes. In the early episodes, we see that Pareto frontier is fairly elongated since there is a significant amount of uncertainty in the GP predictions. This implies there is significant mismatch between the points that may lead to large reward values and those that are likely to satisfy the necessary optimality conditions given our current information. As more data is collected, we see that the Pareto frontier begins to shrink, indicating lower uncertainty in the predicted maximum point. Furthermore, we see that the proposed GN acquisition function provides us with an independent source of information that helps select high reward points that are also likely to satisfy $\nabla f(x) = 0$. Looking at $e = 30$, for example, we see that several points are predicted to perfectly satisfy the necessary optimality condition while simultaneously having large reward values. Thus, the fusion of zeroth- and first-order information appears to be at the

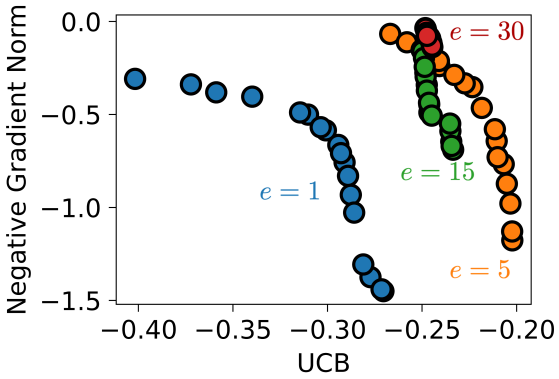

Figure 5.2: Pareto frontiers for the multi-objective acquisition function for four different closed-loop episodes $e \in \{1, 5, 15, 30\}$ in a representative AEGBO run. The x-axis corresponds to the zeroth-order UCB acquisition function in (5.15) and the y-axis corresponds to the first-order GN acquisition function in (5.16).

heart of the improved performance observed in Fig. 5.1.

Feasibility Analayis for NOBO

To further investigate NOBO, we also look to measure the reduction in the volume of the feasible region

$$\varphi_t = \text{Vol}(\mathcal{F}_t^u)/\text{Vol}(X), \tag{5.32}$$

where $\text{Vol}(A) = \int_A dx$ is the volume of a set. The value φ_t, which quantifies the relative size in the feasible region of standard BO can NOBO, can be straightforwardly estimated at any iteration t by Monte Carlo integration. The evolution of φ_t over 50 closed-loop episodes or iterations is shown in Fig. 5.3 (averaged over 100 random initial datasets). Even in the early iterations, φ_t is ~ 0.2, implying only $\sim$20% of the points in X have the potential to satisfy the necessary optimality conditions (i.e., 80% of the points in X have been confidently eliminated from the search process). Furthermore, as NOBO progresses, φ_t continually decreases, which highlights its ability to continually learn from the collected gradient information. As such, NOBO systematically exclude points that are inconsistent with the necessary optimality conditions, leading to a substantial reduction in the search space, without compromising performance.

5.7 Conclusion

This paper presents a gradient-enhanced Bayesian optimization framework for directly using zeroth- and first-order information during the querying process. The first proposed method, AEGBO, defines value following (zeroth-order) and gradient following (first-order) AFs which are jointly optimized via MOO, based on the premise that due to uncertainty those two selection criteria will yield different suggested queries. Several issues related to the

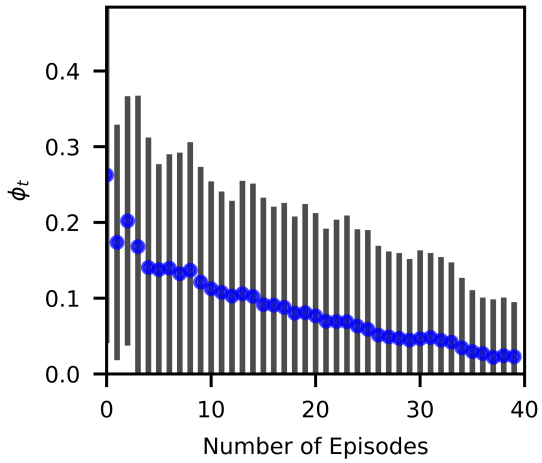

Figure 5.3: The percentage of feasible candidate points in the set $\mathcal{F}_t^u$ based on 30,000 uniform random samples drawn from X over the number of closed-loop episodes T. The circles denote the sample mean based on 100 randomly generated initial datasets, with the error bars representing the sample standard deviation (cutoff at zero).

MOO nature of AEGBO are resolved via our second proposed method, NOBO, that simultaneously leverages zero- and first-order information to sequentially maximize an expensive black-box objective function while imposing optimality constraints, The primary advantage of NOBO is its ability to conduct a more focused search within the design space compared to traditional Bayesian optimization (BO) by excluding points that cannot satisfy necessary optimality conditions (with high probability). We theoretically establish convergence and upper cumulative and simple regret bounds for NOBO. We also demonstrate NOBO's superior performance over traditional BO and policy-based reinforcement learning on a benchmark closed-loop policy optimization problem. Our future work will focus on the incorporation of black-box safety constraints and demonstrations on high-dimensional problems.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The objective of this thesis has been to advance the application of data-driven methods for the dynamical analysis and optimization of complex systems, with emphasis on chemical and biochemical systems. The research encompasses various aspects of flow-map-based dynamical modeling, model-based control, and Bayesian Optimization (BO) techniques, demonstrating their value and applicability in a range of settings, such as uncertainty quantification (UQ), model adaptation, controller auto-tuning, and reinforcement learning (RL).

In **Chapter 2**, we tackled the problem of data-driven model learning. We introduced a flow-map modeling approach based on polynomial-chaos Kriging (PCK) for discovering system dynamics from data. This method facilitated the efficient learning of nonlinear system dynamics and provided a fast-to-evaluate surrogate model for UQ applications. By directly approximating the integration operator of differential equations, the proposed approach addressed the challenge of capturing the dynamical behavior of complex biochemical systems based on limited and possibly noisy data, which is a common situation in real-world applications. The effectiveness of the PCK-based flow-maps was demonstrated in various benchmark systems such as the Lorenz and the Morris-Lecar systems, as well as on more complex systems such a transient co-culture bioreactor and a microbial electrosynthesis reactor. The results emphasized the benefits of this approach in terms of data efficiency and computational efficiency for discovering nonlinear system dynamics and surrogate modeling. In addition, the probabilistic nature of our proposed method is able to provide uncertainty bounds on the predictions, making it useful for safety-critical real-time decision making applications. This method can serve as a foundation for the design and optimization of bioprocesses and integrated biomanufacturing systems.

In the next chapters we shifted the focus to data-driven optimization of closed-loop systems. **Chapter 3** focused on data-driven model-learning in the context of closed-loop control, where the underlying learning objective is not based on prediction accuracy as in **Chapter 2**, but rather relied a system-oriented goal. We introduced a performance-oriented model adaptation approach for model-based control under uncertainty. This approach leveraged the concept of identification for control, in which the underlying model of a controller is adapted over several process runs for maximizing several closed-loop performance measures.

By formulating the model learning problem as a "black-box" multi-objective optimization (MOO) problem solved via BO (hence MOBO), the proposed method systematically accounted for multiple performance measures and demonstrated significant improvements in closed-loop performance. We showed how prior knowledge about the system dynamics can be incorporated into the learning procedure by proposing a composite model structure that allowed for efficient adaptation of the data-driven model in a performance-oriented manner in as few iterations as possible. The method's effectiveness was showcased in a benchmark bioreactor case study, highlighting its potential for multi-objective model learning and autotuning of model-based controllers in processes with finite-time control objectives and high costs associated with each process run. **Chapter 4** also relied on the idea of controller tuning; we addressed the challenge of complex controller structures autotuning under time-invariant uncertainties by utilizing an adversarially robust BO (ARBO) method. The ARBO method relied on a GP model that jointly described the effect of tuning parameters and model uncertainties on closed-loop performance. Using an alternating confidence-bound procedure, ARBO simultaneously selected the next candidate tuning and uncertainty realizations, requiring only one expensive closed-loop simulation per iteration. This approach demonstrated the advantages of systematically exploring and exploiting the joint design-uncertainty space, leading to improved closed-loop performance in the presence of uncertainties. The ARBO method was applied to two case studies, including an illustrative problem and the autotuning of a nonlinear model predictive controller (MPC) for an uncertain bioreactor. Finally, in **Chapter 5** we investigated the incorporation of gradient information into BO by presenting a gradient-enhanced BO framework. Two methods, were proposed to simultaneously exploit performance (zeroth-order) and gradient (first-order) observations within a single acquisition optimization step. Both methods rely on the premise that a first-order based criterion aids in directing the search process of a zeroth-order oracle that suggests query points that maximize the system performance based on our current belief. The first method is called acquisition ensemble gradient-based BO (AEGBO). It utilizes the gradient information to define a gradient-based acquisition function that is optimal if the norm of the gradient at some queried point becomes zero (i.e., the first order optimality condition is satisfied, assuming that the optimal solution is in the interior of the domain). This acquisition is optimized along with a standard zeroth-order acquisition function, in our case the upper confidence bound, using MOO. Several challenges associated with this method are tackled via directly using gradient information into the acquisition optimization in the form of constraints that mimicked the necessary optimality conditions for the original global optimization problem, restricting the allowable search space of BO and leading to less exploration compared to zeroth-order methods. This method is called Necessary Optimality-constrained BO (NOBO). Moreover, theoretical convergence and regret bounds were established for NOBO. The superior performance of the gradient-enhanced BO methods over traditional BO was demonstrated on policy-based RL of a benchmark closed-loop linear quadratic regulator (LQR) system. A rigorous comparison between AEGBO and NOBO would require more careful analysis of the exploration hyperparameters associated with each method.

6.2 Future Work

Despite the successful application of these data-driven methods in various settings, there remain open challenges and opportunities for future work. In this section we present several directions that are relevant to the research theme of this thesis, both from a methods development point of view as well as from the scope of application of these methods to space biomanufacturing.

Integrated Design, Scheduling and Control (iDSC)

Design, scheduling and control of multiple interdependent processes has been a focal point of process systems enginering (PSE), with special attention to integrated approaches of all these elements. The aim is to optimize the overall performance of an integrated system by concurrently considering various aspects of design, scheduling, and control, in contrast to a traditional approach where these tasks are often performed sequentially or in isolation, leading to suboptimal outcomes. Typically, most of research in this field is concerned with integration of a pair of elements (out of design, scheduling and control). For instance, the works of [295, 75, 191], as well as the review of [317], present novel methods and discuss the state-of-the-art on integrated design and control (iDC), where dynamic mixed-integer non-linear programming (MINLP) is the basis for formulating and tackling such problems. The work of [212] highlights the difficulty in adopting an integrated view of all iDSC components, while providing sophisticated multi-parametric programming solvers for mixed continuous/integer problems that serve as a framework for this task. Although MINLP methods along with physics-based models have been the powerhouse of PSE, recently more novel approaches based on machine learning methods and data-driven models have shown promising results [175, 123]. The work of [65] presents a framework for overall iDSC where surrogate models are used to represent black-box constraints and subsequently perform a feasibility analysis, while [125] shows how neural networks and GPs are jointly used within an MINLP framework for problems relevant to process synthesis. The review of [8] discusses several surrogate-enhanced approaches for integrated design, planning and control. To this end, surrogate models that are able to capture uncertainties, such as the polynomial chaos-based Gaussian Process model presented in **Chapter 2**, can be utilized for iDSC with embedded uncertainty quantification for analyzing the impact of uncertainties into the optimal system design. Since the task of iDSC is inherently dynamic, surrogate models that predict the dynamics can be embedded into a robust MINLP optimization framework that yields solutions for a safe iDSC analysis.

Moreover, we envision that this iDSC point of view is of paramount importance for space biomanufacturing. Given that several resources and hardware (e.g., reactors, materials, available crew time) are limited in this context and that operations are time critical, optimally allocating equipment while ensuring that the system performs as expected is essential for the success of a human-based mission and survival of crew members.

High-Dimensional and Mixed Variable Systems

BO has proven to be a powerful method for solving expensive black-box problems (with or without gradient observations incorporated). Nevertheless, addressing the scalability of the presented methods to high-dimensions or their applicability to mixed-integer spaces is non-trivial and an open challenge in the literature. High-dimensional problems are ubiquitous in real-world applications. In **Chapter 3** we alluded to this issue and performed sensitivity analysis to select only a few parameters to tune in our MOBO framework. However, this is a special case of a heuristic utilized for choosing the parameters to tune, equipped with an "importance" cutoff criterion for ultimately selecting the tunable parameters. Hence, more structured and rigorous approaches are need to tackle BO of high-dimensional problems. Methods that address this issue typically rely on dimensionality reduction techniques [173, 140, 301], or on the properties of the acquisition functions [226]. Thus, a possible future research direction is to focus on the problem of policy search that has been presented in **Chapter 5**, in the context of using highly-parametrized policies, for instance as in the case of using neural network policies in deep RL [9]. In particular, it is interesting to explore how methods that handle the aspect of high-dimensionality perform on policy learning tasks using deep RL, as well as large scale optimization problems in the context of biomanufacturing.

Furthermore, MINLP black-box problems are common in industrial settings, as discussed earlier in the context of scheduling, design and control. BO methods that address mixed variable problems have been proposed in the literature [281, 60, 238],including problems with integer, binary and categorial variables. Nevertheless, their combination with gradient-enhanced approaches has not been explored to the best of our knowledge. This could be an interesting extension as many policy search problems include a mix of continuous and integer/binary actions, while problems of this nature are relevant for design and control for deep space biomanufacturing.

Safety-Critical Applications

In the majority of real-world applications, safety restrictions play a crucial role in determining the overall utility of the system. The integration of black-box safety constraints into the optimization has given rise to constrained BO formulations designed for model learning and auto-tuning of controllers in safety-critical applications [133, 84]. The gradient-enhanced BO framework presented in **Chapter 5** includes a necessary optimality constraint, however, the framework, especially the NOBO method, can be extended beyond this to include black-box equality and inequality constraints using the Karush-Kuhn-Tucker (KKT) conditions that were mentioned. Thus, we are interested in the development of algorithms that can efficiently handle multiple safety constraints especially in high-dimensional problems where the complexity of the constraints increases. We further envision that techniques such as transfer learning [6, 280] can greatly assist in incorporating any available knowledge about the system performance and safe operating points in the BO search. Therefore, extending the proposed gradient-enhanced BO methods to incorporate black-box constraints would be a valuable contribution to the field of safe RL [83]. From an application standpoint, safety is paramount in the context of deep space biomanufacturing. As discussed in **Chapter 1**, space bioprocess engineering seeks to apply the principles of PSE to space biomanufacturing,

where unique design and operational constraints emerge due to the highly uncertain nature of unexplored environments. For examples, those constraints can be related to unexpected environmental conditions and changes [2] that impede the efficiency of photovoltaics, which are the primary energy source for the end-to-end biomanufacturing system. Black-box design optimization under safety considerations is generally paramount for the entire mission as discussed in [108].

Bibliography

[1] Anthony J Abel and Douglas S Clark. "A comprehensive modeling analysis of formate-mediated microbial electrosynthesis". In: *ChemSusChem* 14.1 (2021), pp. 344–355.

[2] Anthony J Abel et al. "Photovoltaics-driven power production can support human exploration on Mars". In: *Frontiers in Astronomy and Space Sciences* (2022), p. 78.

[3] National Aeronautics. *NASA's Journey to Mars: Pioneering Next Steps in Space Exploration*. Government Printing Office, 2016.

[4] Pramod Agrawal, George Koshy, and Michael Ramseier. "An algorithm for operating a fed-batch fermentor at optimum specific-growth rate". In: *Biotechnology and Bioengineering* 33.1 (1989), pp. 115–125.

[5] Taha Mohseni Ahooyi et al. "Model-predictive safety system for proactive detection of operation hazards". In: *AIChE Journal* 62.6 (2016), pp. 2024–2042.

[6] Haitham Bou Ammar, Rasul Tutunov, and Eric Eaton. "Safe policy search for lifelong reinforcement learning with sublinear regret". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2361–2369.

[7] Joel A E Andersson et al. "CasADi – A software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36.

[8] Oswaldo Andrés-Martınez and Luis A Ricardez-Sandoval. "Integration of planning, scheduling, and control: A review and new perspectives". In: *The Canadian Journal of Chemical Engineering* 100.9 (2022), pp. 2057–2070.

[9] Kai Arulkumaran et al. "Deep reinforcement learning: A brief survey". In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 26–38.

[10] Helon Vicente Hultmann Ayala and Leandro dos Santos Coelho. "Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator". In: *Expert Systems with Applications* 39.10 (2012), pp. 8968–8974.

[11] Maximilian Balandat et al. "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization." In: *NeurIPS*. 2020.

[12] Julio R Banga et al. "Dynamic optimization of bioprocesses: Efficient and robust numerical strategies". In: *Journal of Biotechnology* 117.4 (2005), pp. 407–419.

[13] Mohammed Saad Faizan Bangi and Joseph Sang-Il Kwon. "Deep hybrid modeling of chemical process: Application to hydraulic fracturing". In: *Computers & Chemical Engineering* 134 (2020), p. 106696.

[14] Somil Bansal et al. "Goal-driven dynamics learning via Bayesian optimization". In: *Proceedings of the 56th IEEE Conference on Decision and Control*. Melbourne, 2017, pp. 5168–5173.

[15] Atilim Gunes Baydin et al. "Automatic differentiation in machine learning: a survey". In: *Journal of Marchine Learning Research* 18 (2018), pp. 1–43.

[16] Hugo Bellamy et al. "Batched bayesian optimization for drug design in noisy environments". In: *Journal of Chemical Information and Modeling* 62.17 (2022), pp. 3970–3981.

[17] Damien van de Berg et al. "Data-driven optimization for process systems engineering applications". In: *Chemical Engineering Science* 248 (2022), p. 117135.

[18] Felix Berkenkamp et al. "Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 4661–4666.

[19] Aaron J Berliner et al. "Space bioprocess engineering on the horizon". In: *Communications Engineering* 1.1 (2022), p. 13.

[20] Aaron J Berliner et al. "Towards a Biomanufactory on Mars". In: *Frontiers in Astronomy and Space Sciences* 8 (2021), p. 120. ISSN: 2296-987X.

[21] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*. Vol. 1. Athena scientific, 2012.

[22] Christopher M Bishop. "Pattern recognition". In: *Machine learning* 128.9 (2006).

[23] Géraud Blatman and Bruno Sudret. "Adaptive sparse polynomial chaos expansion based on least angle regression". In: *Journal of computational Physics* 230.6 (2011), pp. 2345–2367.

[24] I. Bogunovic et al. "Adversarially robust optimization with Gaussian processes". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Montréal, Canada, 2018, pp. 5765–5775.

[25] Josh Bongard and Hod Lipson. "Automated reverse engineering of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences* 104.24 (2007), pp. 9943–9948.

[26] Angelo D Bonzanini, David B Graves, and Ali Mesbah. "Learning-Based SMPC for Reference Tracking Under State-Dependent Uncertainty: An Application to Atmospheric Pressure Plasma Jets for Plasma Medicine". In: *IEEE Transactions on Control Systems Technology* (2021).

[27] Angelo D Bonzanini et al. "Fast approximate learning-based multistage nonlinear model predictive control using Gaussian processes and deep neural networks". In: *Computers & Chemical Engineering* 145 (2021), p. 107174.

[28] E Borgonovo. "A new uncertainty importance measure". In: *Reliability Engineering \& System Safety* 92.6 (2007), pp. 771–784. ISSN: 0951-8320.

[29] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[30] George EP Box and Henry L Lucas. "Design of experiments in non-linear situations". In: *Biometrika* 46.1/2 (1959), pp. 77–90.

[31] George EP Box et al. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[32] E. Bradford, A. M. Schweidtmann, and A. Lapkin. "Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm". In: *Journal of Global Optimization* 71.2 (2018), pp. 407–438.

[33] William Bradley and Fani Boukouvala. "Two-Stage Approach to Parameter Estimation of Differential Equations Using Neural ODEs". In: *Industrial & Engineering Chemistry Research* 60.45 (2021), pp. 16330–16344.

[34] Eric Brochu, Vlad M Cora, and Nando De Freitas. "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning". In: *arXiv preprint arXiv:1012.2599* (2010).

[35] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[36] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the national academy of sciences* 113.15 (2016), pp. 3932–3937.

[37] A. D. Bull. "Convergence rates of efficient global optimization algorithms." In: *Journal of Machine Learning Research* 12.10 (2011).

[38] Richard L Burden, J Douglas Faires, and Annette M Burden. *Numerical analysis*. Cengage learning, 2015.

[39] Russel E Caflisch. "Monte Carlo and Quasi-Monte Carlo methods". In: *Acta numerica* 7 (1998), pp. 1–49.

[40] R.H. Cameron and W T Martin. "The Orthogonal Development of Non-Linear Functionals in Series of Fourier-Hermite Functionals". In: *Annals of Mathematics* 48.2 (1947), pp. 385–392.

[41] Coralia Cartis et al. "Improving the flexibility and robustness of model-based derivative-free optimization solvers". In: *ACM Transactions on Mathematical Software (TOMS)* 45.3 (2019), pp. 1–41.

[42] Kathryn Chaloner and Isabella Verdinelli. "Bayesian experimental design: A review". In: *Statistical science* (1995), pp. 273–304.

[43] Kathleen Champion et al. "Data-driven discovery of coordinates and governing equations". In: *Proceedings of the National Academy of Sciences* 116.45 (2019), pp. 22445–22451.

[44] K. Chan, G. Makrygiorgos, and A. Mesbah. "Towards Personalized Plasma Medicine via Data-Efficient Adaptation of Fast Deep Learning-based MPC Policies". In: *2023 Annual American Control Conference (ACC)*. Vol. Accepted. 2023.

[45] Jorge Chang et al. "Efficient closed-loop maximization of carbon nanotube growth rate using bayesian optimization". In: *Scientific reports* 10.1 (2020), p. 9040.

[46] Jixiang Chen, Fu Luo, and Zhenkun Wang. "Dynamic multi-objective ensemble of acquisition functions in batch Bayesian optimization". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2022, pp. 479–482.

[47] Yingjie Chen and Marianthi Ierapetritou. "A framework of hybrid model development with identification of plant-model mismatch". In: *AIChE Journal* 66.10 (2020), e16996.

[48] Francois Chollet et al. *Keras*. 2015. URL: `https://github.com/fchollet/keras`.

[49] Sayak Ray Chowdhury and Aditya Gopalan. "On kernelized multi-armed bandits". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 844–853.

[50] Victor Churchill et al. "Robust modeling of unknown dynamical systems via ensemble averaged learning". In: *Journal of Computational Physics* 474 (2023), p. 111842.

[51] Douglas S Clark and Harvey W Blanch. *Biochemical engineering*. CRC press, 1997.

[52] Jacques Cohen. "The crucial role of CS in systems and synthetic biology". In: *Communications of the ACM* 51.5 (2008), pp. 15–18.

[53] Luca Console, Claudia Picardi, and D Theseider Duprè. "Temporal decision trees: Model-based diagnosis of dynamic systems on-board". In: *Journal of artificial intelligence research* 19 (2003), pp. 469–512.

[54] Noel Cressie. "The origins of kriging". In: *Mathematical geology* 22.3 (1990), pp. 239–252.

[55] V. Dani, T. P. Hayes, and S. M. Kakade. "Stochastic linear optimization under bandit feedback". In: *21st Annual Conference on Learning Theory* (2008).

[56] Bryan C Daniels and Ilya Nemenman. "Efficient inference of parsimonious phenomenological models of cellular dynamics using S-systems and alternating regression". In: *PloS one* 10.3 (2015), e0119821.

[57] George B Dantzig. "Maximization of a linear function of variables subject to linear inequalities". In: *Activity analysis of production and allocation* 13 (1951), pp. 339–347.

[58] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. "Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization". In: *arXiv preprint arXiv:2006.05078* (2020).

[59] Richard A Davis, Keh-Shin Lii, and Dimitris N Politis. "Remarks on some nonparametric estimates of a density function". In: *Selected Works of Murray Rosenblatt*. Springer, 2011, pp. 95–100.

[60] Erik Daxberger et al. "Mixed-variable Bayesian optimization". In: *arXiv preprint arXiv:1907.01329* (2019).

[61] S Feyo De Azevedo, B Dahm, and FR Oliveira. "Hybrid modelling of biochemical processes: A comparison with the conventional approach". In: *Computers & Chemical Engineering* 21 (1997), S751–S756.

[62] Kalyanmoy Deb. "Multi-objective optimization". In: *Search methodologies*. Springer, 2014, pp. 403–449.

[63] Kalyanmoy Deb et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197.

[64] G Deman et al. "Using sparse polynomial chaos expansions for the global sensitivity analysis of groundwater lifetime expectancy in a multi-layered hydrogeological model". In: *Reliability Engineering and System Safety* 147 (2016), pp. 156–169. ISSN: 09518320.

[65] Lisia S Dias and Marianthi G Ierapetritou. "Integration of planning, scheduling and control problems using data-driven feasibility analysis and surrogate models". In: *Computers & Chemical Engineering* 134 (2020), p. 106714.

[66] Pierre Dubois et al. "Data-driven predictions of the Lorenz system". In: *Physica D: Nonlinear Phenomena* 408 (2020), p. 132495.

[67] Thomas F Edgar and Efstratios N Pistikopoulos. "Smart manufacturing and energy systems". In: *Computers & Chemical Engineering* 114 (2018), pp. 130–144.

[68] Bradley Efron et al. "Least angle regression". In: *Ann. Stat.* 32.2 (Apr. 2004), pp. 407–499. ISSN: 00905364.

[69] Michael Emmerich, Nicola Beume, and Boris Naujoks. "An EMO algorithm using the hypervolume measure as selection criterion". In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2005, pp. 62–76.

[70] Michael TM Emmerich, André H Deutz, and Jan Willem Klinkenberg. "Hypervolume-based expected improvement: Monotonicity properties and exact computation". In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE. 2011, pp. 2147–2154.

[71] Valerii Vadimovich Fedorov. *Theory of optimal experiments*. Elsevier, 2013.

[72] Matthias Feurer and Frank Hutter. "Hyperparameter optimization". In: *Automated Machine Learning*. Springer, Cham, 2019, pp. 3–33.

[73] Marcello Fiducioso et al. "Safe contextual Bayesian optimization for sustainable room temperature PID control tuning". In: *arXiv preprint arXiv:1906.12086* (2019).

[74] D. Finkel. *DIRECT optimization algorithm user guide*. Tech. rep. North Carolina State University. Center for Research in Scientific Computation, 2003.

[75] Antonio Flores-Tlacuahuac and Lorenz T Biegler. "Simultaneous mixed-integer dynamic optimization for integrated design and control". In: *Computers & chemical engineering* 31.5-6 (2007), pp. 588–600.

[76] Marco Forgione, Dario Piga, and Alberto Bemporad. "Efficient calibration of embedded MPC". In: *arXiv preprint arXiv:1911.13021* (2019).

[77] Alexander IJ Forrester, András Sóbester, and Andy J Keane. "Multi-fidelity optimization via surrogate modelling". In: *Proceedings of the royal society a: mathematical, physical and engineering sciences* 463.2088 (2007), pp. 3251–3269.

[78] Urban Forssell and Lennart Ljung. "Closed-loop identification revisited". In: *Automatica* 35.7 (1999), pp. 1215–1241.

[79] Gaia Franceschini and Sandro Macchietto. "Model-based design of experiments for parameter precision: State of the art". In: *Chemical Engineering Science* 63.19 (2008), pp. 4846–4872.

[80] Peter I Frazier. "A tutorial on Bayesian optimization". In: *arXiv preprint 1807.02811* (2018).

[81] Lukas P Fröhlich, Melanie N Zeilinger, and Edgar D Klenske. "Cautious Bayesian Optimization for Efficient and Scalable Policy Search". In: *Learning for Dynamics and Control*. PMLR. 2021, pp. 227–240.

[82] Ken-ichi Funahashi and Yuichi Nakamura. "Approximation of dynamical systems by continuous time recurrent neural networks". In: *Neural networks* 6.6 (1993), pp. 801–806.

[83] Javier Garcıa and Fernando Fernández. "A comprehensive survey on safe reinforcement learning". In: *Journal of Machine Learning Research* 16.1 (2015), pp. 1437–1480.

[84] Jacob R Gardner et al. "Bayesian Optimization with Inequality Constraints". In: *Proceedings of the International Conference on Machine Learning*. Beijing, 2014, pp. 937–945.

[85] Jorge L Garriga and Masoud Soroush. "Model predictive control tuning methods: A review". In: *Industrial & Engineering Chemistry Research* 49.8 (2010), pp. 3505–3515.

[86] Marc G Genton. "Classes of kernels for machine learning: a statistics perspective". In: *Journal of machine learning research* 2.Dec (2001), pp. 299–312.

[87] M. Gevers. "Identification for Control: From the Early Achievements to the Revival of Experiment Design". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. 2005, pp. 12–12.

[88] Michel Gevers. "Towards a joint design of identification and control?" In: *Essays on Control*. Springer, 1993, pp. 111–151.

[89] Debanjan Ghosh et al. "Hybrid modeling approach integrating first-principles models with subspace identification". In: *Industrial & Engineering Chemistry Research* 58.30 (2019), pp. 13533–13543.

[90] Paul Gilmore and Carl T Kelley. "An implicit filtering algorithm for optimization of functions with many local minima". In: *SIAM Journal on Optimization* 5.2 (1995), pp. 269–285.

[91] Agathe Girard et al. "Gaussian process priors with uncertain inputs: application to multiple-step ahead time series forecasting". In: (2003).

[92] Andrew Golightly and Darren J Wilkinson. "Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo". In: *Interface focus* 1.6 (2011), pp. 807–820.

[93] Aldair E Gongora et al. "A Bayesian experimental autonomous researcher for mechanical design". In: *Science advances* 6.15 (2020), eaaz1708.

[94] Joan Gonzalvez et al. "Financial applications of Gaussian processes and Bayesian optimization". In: *arXiv preprint arXiv:1903.04841* (2019).

[95] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[96] GPy. *GPy: A Gaussian process framework in python*. 2012.

[97] Ignacio E Grossmann and Arthur W Westerberg. "Research challenges in process systems engineering". In: *AIChE Journal* 46.9 (2000), pp. 1700–1703.

[98] Andreia P Guerreiro, Carlos M Fonseca, and Luis Paquete. "The hypervolume indicator: Problems and algorithms". In: *arXiv preprint arXiv:2005.00515* (2020).

[99] Nyoman Gunantara. "A review of multi-objective optimization: Methods and its applications". In: *Cogent Engineering* 5.1 (2018), p. 1502242.

[100] Nikolaus Hansen and Andreas Ostermeier. "Completely derandomized self-adaptation in evolution strategies". In: *Evolutionary computation* 9.2 (2001), pp. 159–195.

[101] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015.

[102] Markus Heinonen et al. "Learning unknown ODE models with Gaussian processes". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1959–1968.

[103] Michael A. Henson and Dale E. Seborg. "Nonlinear control strategies for continuous fermenters". In: *Chemical Engineering Science* 47.4 (1992), pp. 821–835. ISSN: 0009-2509.

[104] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. "Predictive entropy search for efficient global optimization of black-box functions". In: *Advances in neural information processing systems* 27 (2014).

[105] Lukas Hewing, Juraj Kabzan, and Melanie N Zeilinger. "Cautious model predictive control using gaussian process regression". In: *IEEE Transactions on Control Systems Technology* 28.6 (2019), pp. 2736–2743.

[106] Lukas Hewing et al. "Learning-based model predictive control: Toward safe learning in control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020), pp. 269–296.

[107] Lukas Hewing et al. "On simulation and trajectory prediction with gaussian process dynamics". In: *Learning for Dynamics and Control*. PMLR. 2020, pp. 424–434.

[108] Davian Ho et al. "Towards an extension of equivalent system mass for human exploration missions on Mars". In: *npj Microgravity* 8.1 (2022), p. 30.

[109] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[110] Xun Huan and Youssef M Marzouk. "Simulation-based optimal Bayesian experimental design for nonlinear systems". In: *Journal of Computational Physics* 232.1 (2013), pp. 288–317.

[111] Zhongling Huang, Zongxu Pan, and Bin Lei. "Transfer Learning with Deep Convolutional Neural Network for SAR Target Classification with Limited Labeled Data". In: *Remote Sensing* 9.9 (2017). ISSN: 2072-4292.

[112] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration". In: *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*. Springer. 2011, pp. 507–523.

[113] Bertrand Iooss and Paul Lemaıtre. "A review on global sensitivity analysis methods". In: *Uncertainty management in simulation-optimization of complex systems*. Springer, 2015, pp. 101–122.

[114] C. Jin, P. Netrapalli, and M. Jordan. "What is local optimality in nonconvex-nonconcave minimax optimization?" In: *Proceedings of the International Conference on Machine Learning*. 2020, pp. 4880–4889.

[115] Donald R Jones, Matthias Schonlau, and William J Welch. "Efficient global optimization of expensive black-box functions". In: *Journal of Global optimization* 13.4 (1998), p. 455.

[116] Motonobu Kanagawa et al. "Gaussian processes and kernel methods: A review on connections and equivalences". In: *arXiv preprint arXiv:1807.02582* (2018).

[117] K. Kandasamy, J. Schneider, and B. Póczos. "High dimensional Bayesian optimisation and bandits via additive models". In: *Proceedings of the International Conference on Machine Learning*. 2015, pp. 295–304.

[118] Kirthevasan Kandasamy et al. "Gaussian process optimisation with multi-fidelity evaluations". In: *Proceedings of the 30th/International Conference on Advances in Neural Information Processing Systems (NIPS'30)*. 2016.

[119] Rituraj Kaushik, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. "Multi-objective model-based policy search for data-efficient learning with sparse rewards". In: *Conference on Robot Learning*. PMLR. 2018, pp. 839–855.

[120] M Kazemi, D Biria, and H Rismani-Yazdi. "Modelling bio-electrosynthesis in a reverse microbial fuel cell to produce acetate from CO2 and H2O". In: *Phys. Chem. Chem. Phys.* 17.19 (2015), pp. 12561–12574.

[121] James Kennedy and Russell Eberhart. "Particle swarm optimization". In: *Proceedings of ICNN'95-international conference on neural networks*. Vol. 4. IEEE. 1995, pp. 1942–1948.

[122] Marc C Kennedy and Anthony O'hagan. "Bayesian calibration of computer models". In: *J. R. Statist. Soc. B* 63 (2001), pp. 425–464.

[123] Nadia Khan and Syed Ali Ammar Taqvi. "Machine learning an intelligent approach in process industries: A perspective and overview". In: *ChemBioEng Reviews* (2023).

[124] Mohammad Khosravi et al. "Performance-driven cascade controller tuning with Bayesian optimization". In: *IEEE Transactions on Industrial Electronics* 69.1 (2021), pp. 1032–1042.

[125] Sun Hye Kim and Fani Boukouvala. "Surrogate-based optimization for mixed-integer nonlinear problems". In: *Computers & Chemical Engineering* 140 (2020), p. 106847.

[126] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. "Optimization by simulated annealing". In: *science* 220.4598 (1983), pp. 671–680.

[127] Anton Kleywegt, Alexander Shapiro, and Tito Homem-De-Mello. "The Sample Average Approximation Method for Stochastic Discrete Optimization". In: *SIAM Journal on Optimization* 12 (Jan. 2001), pp. 479–.

[128] Michał Komorowski et al. "Bayesian inference of biochemical kinetic parameters using the linear noise approximation". In: *BMC bioinformatics* 10.1 (2009), pp. 1–10.

[129] Abdullah Konak, David W Coit, and Alice E Smith. "Multi-objective optimization using genetic algorithms: A tutorial". In: *Reliability engineering & system safety* 91.9 (2006), pp. 992–1007.

[130] Stefan Körkel et al. "Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes". In: *Optimization Methods and Software* 19.3-4 (2004), pp. 327–338.

[131] Michalis Koutinas et al. "Bioprocess systems engineering: transferring traditional process engineering principles to industrial biotechnology". In: *Computational and structural biotechnology journal* 3.4 (2012), e201210022.

[132] Akshay Kudva, Farshud Sorouifar, and Joel A Paulson. "Efficient robust global optimization for simulation-based problems using decomposed Gaussian processes: Application to MPC calibration". In: *Proceedings of the American Control Conference*. 2022, pp. 2091–2097.

[133] Akshay Kudva, Farshud Sorourifar, and Joel A Paulson. "Constrained robust Bayesian optimization of expensive noisy black-box functions with guaranteed regret bounds". In: *AIChE Journal* 68.12 (2022), e17857.

[134] S Narendra Kumpati, Parthasarathy Kannan, et al. "Identification and control of dynamical systems using neural networks". In: *IEEE Transactions on neural networks* 1.1 (1990), pp. 4–27.

[135] J Nathan Kutz et al. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016. ISBN: 978-1-61197-449-2.

[136] Rémi Lam et al. "Advances in Bayesian optimization with applications in aerospace engineering". In: *2018 AIAA Non-Deterministic Approaches Conference*. 2018, p. 1656.

[137] Christopher E Lawson et al. "Machine learning for metabolic engineering: A review". In: *Metabolic Engineering* 63 (2021), pp. 34–60.

[138] Jay H Lee, Joohyun Shin, and Matthew J Realff. "Machine learning: Overview of the recent progresses and implications for the process systems engineering field". In: *Computers & Chemical Engineering* 114 (2018), pp. 111–121.

[139] Benjamin Letham and Eytan Bakshy. "Bayesian Optimization for Policy Search via Online-Offline Experimentation." In: *J. Mach. Learn. Res.* 20 (2019), pp. 145–1.

[140] Benjamin Letham et al. "Re-examining linear embeddings for high-dimensional Bayesian optimization". In: *Advances in neural information processing systems* 33 (2020), pp. 1546–1558.

[141] Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

[142] Junyi Li and R Russell Rhinehart. "Heuristic random optimization". In: *Computers & chemical engineering* 22.3 (1998), pp. 427–444.

[143] Qiaohao Liang and Lipeng Lai. "Scalable bayesian optimization accelerates process optimization of penicillin production". In: *NeurIPS 2021 AI for Science Workshop*. 2021.

[144] Qiaohao Liang et al. "Benchmarking the performance of Bayesian optimization across multiple experimental materials science domains". In: *npj Computational Materials* 7.1 (2021), p. 188.

[145] Xiaoping Liao, Hongwu Ma, and Yinjie J Tang. "Artificial intelligence: a solution to involution of design–build–test–learn cycle". In: *Current Opinion in Biotechnology* 75 (2022), p. 102712.

[146] Haitao Liu, Jianfei Cai, and Yew-Soon Ong. "Remarks on multi-output Gaussian process regression". In: *Knowledge-Based Systems* 144 (2018), pp. 102–121.

[147] Jun S Liu and Rong Chen. "Sequential monte carlo methods for dynamic systems". In: *Journal of the American Statistical Association* 93.443 (Sept. 1998), pp. 1032–1044. ISSN: 1537274X.

[148] Wei-Liem Loh. "On Latin hypercube sampling". In: *The annals of statistics* 24.5 (1996), pp. 2058–2080.

[149] Congwen Lu and Joel A Paulson. "No-Regret Bayesian Optimization with Unknown Equality and Inequality Constraints using Exact Penalty Functions". In: *IFAC Papers On Line* 55.7 (2022), pp. 895–902.

[150] Qiugang Lu, Ranjeet Kumar, and Victor M. Zavala. *MPC Controller Tuning using Bayesian Optimization Techniques*. 2020. arXiv: `2009.14175 [eess.SY]`.

[151] Nora Lüthen, Stefano Marelli, and Bruno Sudret. "Sparse polynomial chaos expansions: Literature survey and benchmark". In: *SIAM/ASA Journal on Uncertainty Quantification* 9.2 (2021), pp. 593–649.

[152] Benjamin P MacLeod et al. "Self-driving laboratory for accelerated discovery of thin-film materials". In: *Science Advances* 6.20 (2020), eaaz8867.

[153] Chu V Mai and Bruno Sudret. "Surrogate models for oscillatory systems using sparse polynomial chaos expansions and stochastic time warping". In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017), pp. 540–571.

[154] Georgios Makrygiorgos, Giovanni Maria Maggioni, and Ali Mesbah. "Surrogate modeling for fast uncertainty quantification: Application to 2D population balance models". In: *Computers & Chemical Engineering* 138 (2020), p. 106814.

[155] Georgios Makrygiorgos, Joel A Paulson, and Ali Mesbah. "Gradient-Enhanced Bayesian Optimization via Acquisition Ensembles with Application to Reinforcement Learning". In: *IFAC-PapersOnLine, Accepted* (2023).

[156] Georgios Makrygiorgos, Joel A Paulson, and Ali Mesbah. "No-Regret Bayesian Optimization with Gradients using Local Optimality-based Constraints: Application to Closed-loop Policy Search". In: *2023 IEEE 62nd Conference on Decision and Control (CDC), Submitted* (2023).

[157] Georgios Makrygiorgos et al. "Data-driven flow-map models for data-efficient discovery of dynamics and fast uncertainty quantification of biological and biochemical systems". In: *Biotechnology and Bioengineering* 120.3 (2023), pp. 803–818.

[158] Georgios Makrygiorgos et al. "Fast Probabilistic Uncertainty Quantification and Sensitivity Analysis of a Mars Life Support System Model". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 7268–7273.

[159] Georgios Makrygiorgos et al. "Performance-oriented model learning for control via multi-objective Bayesian optimization". In: *Computers & Chemical Engineering* 162 (2022), p. 107770.

[160] Alonso Marco et al. "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 1557–1563.

[161] Andrew Kato Marcus, C Torres, and B Rittmann. "Conduction-based modeling of the biofilm anode of a microbial fuel cell". In: *Biotechnology and Bioengineering* 98 (2007).

[162] Stefano Marelli and Bruno Sudret. "UQLab: A framework for uncertainty quantification in Matlab". In: *Vulnerability, uncertainty, and risk: quantification, mitigation, and management*. 2014, pp. 2554–2563.

[163] R Timothy Marler and Jasbir S Arora. "Survey of multi-objective optimization methods for engineering". In: *Structural and multidisciplinary optimization* 26.6 (2004), pp. 369–395.

[164] Marc Martin-Casas and Ali Mesbah. "Discrimination between competing model structures of biological systems in the presence of population heterogeneity". In: *IEEE Life Sciences Letters* 2.3 (2016), pp. 23–26.

[165] J. Marzat, E. Walter, and H. Piet-Lahanier. "A new expected-improvement algorithm for continuous minimax optimization". In: *Journal of Global Optimization* 64.4 (2016), pp. 785–802.

[166] Robert M May. "Biological populations with nonoverlapping generations: stable points, stable cycles, and chaos". In: *Science* 186.4164 (1974), pp. 645–647.

[167] David Q Mayne. "Model predictive control: Recent developments and future promise". In: *Automatica* 50.12 (2014), pp. 2967–2986.

[168] Ali Mesbah. "Stochastic model predictive control with active uncertainty learning: A survey on dual control". In: *Annual Reviews in Control* 45 (2018), pp. 107–117.

[169] Ali Mesbah. "Stochastic model predictive control: An overview and perspectives for future research". In: *IEEE Control Systems* 36.6 (2016), pp. 30–44.

[170] Ali Mesbah and Stefan Streif. "A probabilistic approach to robust optimal experiment design with chance constraints". In: *IFAC-PapersOnLine* 48.8 (2015), pp. 100–105.

[171] Ali Mesbah et al. "Fusion of Machine Learning and MPC under Uncertainty: What Advances Are on the Horizon?" In: *Proceedings of the American Control Conference*. IEEE. 2022, pp. 342–357.

[172] Carmen G Moles, Pedro Mendes, and Julio R Banga. "Parameter estimation in biochemical pathways: a comparison of global optimization methods". In: *Genome research* 13.11 (2003), pp. 2467–2474.

[173] Riccardo Moriconi, Marc Peter Deisenroth, and KS Sesh Kumar. "High-dimensional Bayesian optimization using low-dimensional feature spaces". In: *Machine Learning* 109 (2020), pp. 1925–1943.

[174] Catherine Morris and Harold Lecar. "Voltage oscillations in the barnacle giant muscle fiber". In: *Biophysical journal* 35.1 (1981), pp. 193–213.

[175] Max Mowbray et al. "Industrial data science–a review of machine learning applications for chemical and process industries". In: *Reaction Chemistry & Engineering* (2022).

[176] Max Mowbray et al. "Machine learning for biochemical engineering: A review". In: *Biochemical Engineering Journal* 172 (2021), p. 108054.

[177] D. M. Mukhopadhyay et al. "Genetic algorithm: A tutorial review". In: *International Journal of Grid and Distributed Computing* 2.3 (2009), pp. 25–32.

[178] Sarah Müller, Alexander von Rohr, and Sebastian Trimpe. "Local policy search with Bayesian optimization". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 20708–20720.

[179] Habib N Najm. "Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics". In: *Annual Review of Fluid Mechanics* 41.1 (2009), pp. 35–52.

[180] Habib N Najm et al. "Uncertainty quantification in chemical systems". In: *International journal for numerical methods in engineering* 80.6-7 (2009), pp. 789–814.

[181] Harini Narayanan et al. "A new generation of predictive models: the added value of hybrid models for manufacturing processes of therapeutic proteins". In: *Biotechnology and Bioengineering* 116.10 (2019), pp. 2540–2549.

[182] John A Nelder and Roger Mead. "A simplex method for function minimization". In: *The computer journal* 7.4 (1965), pp. 308–313.

[183] Matthias Neumann-Brosig et al. "Data-Efficient Autotuning With Bayesian Optimization: An Industrial Control Study". In: *IEEE Transactions on Control Systems Technology* 28.3 (2019), pp. 730–740.

[184] Thanh Thi Nguyen et al. "A multi-objective deep reinforcement learning framework". In: *Engineering Applications of Artificial Intelligence* 96 (2020), p. 103915.

[185] Majid Nour, Zafer Cömert, and Kemal Polat. "A novel medical diagnosis model for COVID-19 infection detection based on deep features and Bayesian optimization". In: *Applied Soft Computing* 97 (2020), p. 106580.

[186] Jared O'Leary, Joel A Paulson, and Ali Mesbah. "Stochastic physics-informed neural ordinary differential equations". In: *Journal of Computational Physics* 468 (2022), p. 111466.

[187] S Oladyshkin and W Nowak. "Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion". In: *Reliab. Eng. Syst. Saf.* 106 (2012), pp. 179–190.

[188] Lars F Olsen and Anita Lunding. "Chaos in the peroxidase–oxidase oscillator". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.1 (2021), p. 013119.

[189] Samuel E Otto and Clarence W Rowley. "Koopman operators for estimation and control of dynamical systems". In: *Annual Review of Control, Robotics, and Autonomous Systems* 4 (2021), pp. 59–87.

[190] Samay Pande et al. "Fitness and stability of obligate cross-feeding interactions that emerge upon gene loss in bacteria". In: *The ISME journal* 8.5 (2014), pp. 953–962.

[191] Christos S Patilas and Ioannis K Kookos. "A novel approach to the simultaneous design & control problem". In: *Chemical Engineering Science* 240 (2021), p. 116637.

[192] J. A. Paulson and A. Mesbah. "Data-Driven Scenario Optimization for Automated Controller Tuning With Probabilistic Performance Guarantees". In: *IEEE Control Systems Letters* 5.4 (2020), pp. 1477–1482.

[193] J. A. Paulson and A. Mesbah. "Nonlinear model predictive control with explicit backoffs for stochastic systems under arbitrary uncertainty". In: *IFAC-PapersOnLine* 51.20 (2018), pp. 523–534.

[194] J. A. Paulson and A. Mesbah. "Shaping the closed-loop behavior of nonlinear systems under probabilistic uncertainty using arbitrary polynomial chaos". In: *Proceedings of the IEEE Conference on Decision and Control*. 2018, pp. 6307–6313.

[195] J. A. Paulson, K. Shao, and A. Mesbah. "Probabilistically Robust Bayesian Optimization for Data-Driven Design of Arbitrary Controllers with Gaussian Process Emulators". In: *Proceedings of the IEEE Conference on Decision and Control*. 2021, Accepted.

[196] Joel A Paulson, Edward A Buehler, and Ali Mesbah. "Arbitrary polynomial chaos for uncertainty propagation of correlated random variables in dynamic systems". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 3548–3553.

[197] Joel A Paulson, Georgios Makrygiorgos, and Ali Mesbah. "Adversarially robust Bayesian optimization for efficient auto-tuning of generic control structures under uncertainty". In: *AIChE Journal* 68.6 (2022), e17591.

[198] Joel A Paulson, Marc Martin-Casas, and Ali Mesbah. "Fast uncertainty quantification for dynamic flux balance analysis using non-smooth polynomial chaos expansions". In: *PLoS computational biology* 15.8 (2019), e1007308.

[199] Joel A Paulson, Marc Martin-Casas, and Ali Mesbah. "Optimal Bayesian experiment design for nonlinear dynamic systems with chance constraints". In: *Journal of Process Control* 77 (2019), pp. 155–171.

[200] Joel A Paulson and Ali Mesbah. "Data-Driven Scenario Optimization for Automated Controller Tuning with Probabilistic Performance Guarantees". In: *IEEE Control Systems Letters* 5.4 (2020), pp. 1477–1482.

[201] Joel A Paulson, Ketong Shao, and Ali Mesbah. "Probabilistically robust Bayesian optimization for data-driven design of arbitrary controllers with Gaussian process emulators". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 3633–3639.

[202] Joel A Paulson, Farshud Sorouifar, and Ankush Chakrabarty. "Efficient Multi-Step Lookahead Bayesian Optimization with Local Search Constraints". In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE. 2022, pp. 123–129.

[203] Joel A. Paulson, Marc Martin-Casas, and Ali Mesbah. "Input design for online fault diagnosis of nonlinear systems with stochastic uncertainty". In: *Industrial & Engineering Chemistry Research* 56.34 (2017), pp. 9593–9605.

[204] "PC-Kriging: a new metamodelling method combining polynomial chaos expansions and kriging". In: *Proc. 2nd Int. Symp. Uncertain. Quantif. Stoch. Model.* (2014).

[205] Santosh Penubothula, Chandramouli Kamanchi, Shalabh Bhatnagar, et al. "Novel first order Bayesian optimization with an application to reinforcement learning". In: *Applied Intelligence* 51.3 (2021), pp. 1565–1579.

[206] Fábio Henrique Pereira, Pedro HT Schimit, and Francisco Elânio Bezerra. "A deep learning based surrogate model for the parameter identification problem in probabilistic cellular automaton epidemic models". In: *Computer Methods and Programs in Biomedicine* 205 (2021), p. 106078.

[207] Panagiotis Petsagkourakis and Federico Galvanin. "Safe model-based design of experiments using Gaussian processes". In: *Computers & Chemical Engineering* 151 (2021), p. 107339.

[208] Panagiotis Petsagkourakis et al. "Reinforcement learning for batch bioprocess optimization". In: *Computers & Chemical Engineering* 133 (2020), p. 106649.

[209] CL Pettit and PS Beran. "Spectral and multiresolution Wiener expansions of oscillatory stochastic processes". In: *Journal of Sound and Vibration* 294.4-5 (2006), pp. 752–779.

[210] Dario Piga et al. "Performance-oriented model learning for data-driven MPC design". In: *IEEE Control Systems Letters* 3.3 (2019), pp. 577–582.

[211] José Pinto et al. "A bootstrap-aggregated hybrid semi-parametric modeling framework for bioprocess development". In: *Bioprocess and Biosystems Engineering* 42.11 (2019), pp. 1853–1865.

[212] Efstratios N Pistikopoulos and Nikolaos A Diangelakis. "Towards the integration of process design, control and scheduling: Are we getting closer?" In: *Computers & Chemical Engineering* 91 (2016), pp. 85–92.

[213] R-E Plessix. "A review of the adjoint-state method for computing the gradient of a functional with geophysical applications". In: *Geophysical Journal International* 167.2 (2006), pp. 495–503.

[214] Kyriakos Polymenakos, Alessandro Abate, and Stephen Roberts. "Safe Policy Search with Gaussian Process Models". In: *arXiv* (2017).

[215] Wolfgang Ponweiser et al. "Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2008, pp. 784–794.

[216] Michael JD Powell. "A new algorithm for unconstrained optimization". In: *Nonlinear programming*. Elsevier, 1970, pp. 31–65.

[217] Michael JD Powell. "The BOBYQA algorithm for bound constrained optimization without derivatives". In: *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge* (2009), pp. 26–46.

[218] Luc Pronzato and Éric Walter. "Robust experiment design via stochastic approximation". In: *Mathematical Biosciences* 75.1 (1985), pp. 103–120.

[219] S Joe Qin. "An overview of subspace identification". In: *Computers & chemical engineering* 30.10-12 (2006), pp. 1502–1513.

[220] Tong Qin, Kailiang Wu, and Dongbin Xiu. "Data driven governing equations approximation using deep neural networks". In: *Journal of Computational Physics* 395 (2019), pp. 620–635. ISSN: 0021-9991.

[221] Tong Qin et al. *Deep learning of parameterized equations with applications to uncertainty quantification*. 2020.

[222] Christopher Rackauckas et al. "Universal differential equations for scientific machine learning". In: *arXiv preprint arXiv:2001.04385* (2020).

[223] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707.

[224] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. "Multistep neural networks for data-driven discovery of nonlinear dynamical systems". In: *arXiv* (2018).

[225] Prajit Ramachandran, Barret Zoph, and Quoc V Le. "Searching for activation functions". In: *arXiv preprint arXiv:1710.05941* (2017).

[226] Santu Rana et al. "High dimensional Bayesian optimization with elastic Gaussian process". In: *International conference on machine learning*. PMLR. 2017, pp. 2883–2891.

[227] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Cambridge, MA: MIT Press, 2006. ISBN: 026218253X. eprint: 026218253X.

[228] Carl Edward Rasmussen. "Gaussian processes in machine learning". In: *Summer school on machine learning*. Springer. 2003, pp. 63–71.

[229] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*. Vol. 2. Nob Hill Publishing Madison, WI, 2017.

[230] Benjamin Recht. "A tour of reinforcement learning: The view from continuous control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019), pp. 253–279.

[231] Rommel G Regis and Christine A Shoemaker. "A stochastic radial basis function method for the global optimization of expensive functions". In: *INFORMS Journal on Computing* 19.4 (2007), pp. 497–509.

[232] Gilberto Reynoso-Meza et al. "Controller tuning using evolutionary multi-objective optimisation: current trends and applications". In: *Control Engineering Practice* 28 (2014), pp. 58–73.

[233] Ehecatl Antonio del Rio-Chanona et al. "Deep learning-based surrogate modeling and optimization for microalgal biofuel production and photobioreactor design". In: *AIChE Journal* 65.3 (2019), pp. 915–923.

[234] Luis Miguel Rios and Nikolaos V Sahinidis. "Derivative-free optimization: a review of algorithms and comparison of software implementations". In: *Journal of Global Optimization* 56 (2013), pp. 1247–1293.

[235] Diogo Rodrigues, Georgios Makrygiorgos, and Ali Mesbah. "Tractable Global Solutions to Bayesian Optimal Experiment Design". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 1614–1619.

[236] Diogo Rodrigues, Georgios Makrygiorgos, and Ali Mesbah. "Tractable global solutions to chance-constrained Bayesian optimal experiment design for arbitrary prior and noise distributions". In: *Journal of Process Control* 116 (2022), pp. 1–18.

[237] Sascha Rollié, Michael Mangold, and Kai Sundmacher. "Designing biological systems: systems engineering meets synthetic biology". In: *Chemical Engineering Science* 69.1 (2012), pp. 1–29.

[238] Binxin Ru et al. "Bayesian optimisation over multiple continuous and categorical inputs". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8276–8285.

[239] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. "Latent ordinary differential equations for irregularly-sampled time series". In: *Advances in neural information processing systems* 32 (2019).

[240] Samuel H Rudy, J Nathan Kutz, and Steven L Brunton. "Deep learning of dynamics and signal-noise decomposition with time-stepping constraints". In: *Journal of Computational Physics* 396 (2019), pp. 483–506.

[241] Philipp Rumschinski et al. "Set-based dynamical parameter estimation and model invalidation for biochemical reaction networks". In: *BMC systems biology* 4.1 (2010), pp. 1–14.

[242] Andrea Saltelli et al. "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index". In: *Computer physics communications* 181.2 (2010), pp. 259–270.

[243] Syusuke Sano et al. "Application of Bayesian optimization for pharmaceutical product development". In: *Journal of Pharmaceutical Innovation* 15 (2020), pp. 333–343.

[244] Kumara Sastry, David Goldberg, and Graham Kendall. "Genetic algorithms". In: *Search methodologies*. Springer, 2005, pp. 97–125.

[245] Thomas Savage et al. "Data-driven optimization for process systems engineering applications". In: *Chemical Engineering Science* (2021), p. 117135.

[246] Claudia Schillings et al. "Efficient characterization of parametric uncertainty of complex (bio) chemical networks". In: *PLoS Computational Biology* 11.8 (2015), e1004457.

[247] Michael Schmidt and Hod Lipson. "Distilling free-form natural laws from experimental data". In: *science* 324.5923 (2009), pp. 81–85.

[248] Michael D Schmidt et al. "Automated refinement and inference of analytical models for metabolic networks". In: *Physical biology* 8.5 (2011), p. 055011.

[249] Johan Schoukens and Lennart Ljung. "Nonlinear System Identification: A User Oriented Roadmap". In: *arXiv e-prints*, arXiv:1902.00683 (Feb. 2019), arXiv:1902.00683. arXiv: `1902.00683 [cs.SY]`.

[250] Jörg Schubert et al. "Bioprocess optimization and control: Application of hybrid modelling". In: *Journal of biotechnology* 35.1 (1994), pp. 51–68.

[251] B. Shahriari et al. "Taking the Human Out of the Loop: A Review of Bayesian Optimization". In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175.

[252] Bobak Shahriari et al. "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175.

[253] Shubhanshu Shekhar and Tara Javidi. "Significance of gradient information in Bayesian optimization". In: *International Conference on Artificial Intelligence and Statistics*. 2021, pp. 2836–2844.

[254] Jonas Sjöberg et al. "Nonlinear black-box modeling in system identification: a unified overview". In: *Automatica* 31.12 (1995), pp. 1691–1724.

[255] David Slepian. "The one-sided barrier problem for Gaussian noise". In: *Bell System Technical Journal* 41.2 (1962), pp. 463–501.

[256] Ralph C Smith. *Uncertainty quantification: theory, implementation, and applications*. Vol. 12. SIAM, 2013.

[257] Edward Snelson and Zoubin Ghahramani. "Local and global sparse Gaussian process approximations". In: *Artificial Intelligence and Statistics*. PMLR. 2007, pp. 524–531.

[258] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems*. 2012, pp. 2951–2959.

[259] András Sobester, Alexander Forrester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[260] Ilya M Sobol. "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates". In: *Mathematics and computers in simulation* 55.1-3 (2001), pp. 271–280.

[261] Jialin Song, Yuxin Chen, and Yisong Yue. "A general framework for multi-fidelity bayesian optimization with gaussian processes". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 3158–3167.

[262] Farshud Sorourifar et al. "A data-driven automatic tuning method for MPC under uncertainty using constrained Bayesian optimization". In: *IFAC-PapersOnLine* 54.3 (2021), pp. 243–250.

[263] Masoud Soroush et al. "Model-predictive safety optimal actions to detect and handle process operation hazards". In: *AIChE Journal* 66.6 (2020), e16932.

[264] Colin Sparrow. *The Lorenz equations: bifurcations, chaos, and strange attractors*. Vol. 41. Springer Science & Business Media, 2012.

[265] Jost Tobias Springenberg et al. "Bayesian optimization with robust Bayesian neural networks". In: *Advances in neural information processing systems* 29 (2016).

[266] N. Srinivas et al. "Gaussian process optimization in the bandit setting: No regret and experimental design". In: *Proceedings of the International Conference on Machine Learning*. 2015, pp. 2171–2180.

[267] Niranjan Srinivas et al. "Information-theoretic regret bounds for gaussian process optimization in the bandit setting". In: *IEEE transactions on information theory* 58.5 (2012), pp. 3250–3265.

[268] George Stephanopoulos, Aristos A Aristidou, and Jens Nielsen. "Metabolic engineering: principles and methodologies". In: (1998).

[269] Roy Sterritt et al. "Sustainable and autonomic space exploration missions". In: *2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06)*. IEEE. 2006, 8–pp.

[270] Stefan Streif et al. "Optimal experimental design for probabilistic model discrimination using polynomial chaos". In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 4103–4109.

[271] Stefan Streif et al. "Robustness analysis, prediction, and estimation for uncertain biochemical networks: An overview". In: *J. Process Control* 42 (2016), pp. 14–34. ISSN: 09591524.

[272] Wei-Hung Su, Ching-Shan Chou, and Dongbin Xiu. "Deep Learning of Biological Models from Data: Applications to ODE Models". In: *Bulletin of Mathematical Biology* 83.3 (2021), pp. 1–19.

[273] B Sudret, S Marelli, and J Wiart. "Surrogate models for uncertainty quantification: An overview". In: *2017 11th European Conference on Antennas and Propagation (EUCAP)*. Mar. 2017, pp. 793–797.

[274] B. Sudret. "Global sensitivity analysis using polynomial chaos expansions". In: *Reliability Engineering & System Safety* 93.7 (2008), pp. 964–979.

[275] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[276] Richard S Sutton et al. "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in neural information processing systems* 12 (1999).

[277] Johan AK Suykens. "Support vector machines and kernel-based learning for dynamical systems modelling". In: *IFAC Proceedings Volumes* 42.10 (2009), pp. 1029–1037.

[278] Shinya Suzuki et al. "Multi-objective Bayesian optimization using Pareto-frontier entropy". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9279–9288.

[279] Eric Taw and Jeffrey Neaton. "Accelerated Discovery of CH4 Uptake Capacity MOFs using Bayesian Optimization". In: *ChemRxiv* (2021).

[280] Matthew E Taylor and Peter Stone. "Transfer learning for reinforcement learning domains: A survey." In: *Journal of Machine Learning Research* 10.7 (2009).

[281] Alexander Thebelt et al. "Tree ensemble kernels for Bayesian optimization with known constraints over mixed-feature spaces". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 37401–37415.

[282] Michalis Titsias. "Variational learning of inducing variables in sparse Gaussian processes". In: *Artificial intelligence and statistics*. PMLR. 2009, pp. 567–574.

[283] César I Torres et al. "Kinetic experiments for evaluating the Nernst- Monod model for anode-respiring bacteria (ARB) in a biofilm anode". In: *Environmental science & technology* 42.17 (2008), pp. 6593–6597.

[284] Dustin Tran et al. "Bayesian layers: A module for neural network uncertainty". In: *arXiv preprint arXiv:1812.03973* (2018).

[285] Q. N. Tran et al. "A model-free approach for auto-tuning of model predictive control". In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 2189–2194.

[286] Neythen J Treloar et al. "Deep reinforcement learning for the control of microbial co-cultures in bioreactors". In: *PLoS computational biology* 16.4 (2020), e1007783.

[287] Rohit Tripathy, Ilias Bilionis, and Marcial Gonzalez. "Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation". In: *Journal of Computational Physics* 321 (2016), pp. 191–223.

[288] Rohit K Tripathy and Ilias Bilionis. "Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification". In: *Journal of Computational Physics* 375 (2018), pp. 565–588. ISSN: 0021-9991.

[289] Michail Tsagris, Christina Beneki, and Hossein Hassani. "On the folded normal distribution". In: *Mathematics* 2.1 (2014), pp. 12–28.

[290] Evgenii Tsymbalov, Maxim Panov, and Alexander Shapeev. "Dropout-based active learning for regression". In: *International conference on analysis of images, social networks and texts*. Springer. 2018, pp. 247–258.

[291] Matteo Turchetta, Andreas Krause, and Sebastian Trimpe. "Robust model-free reinforcement learning with multi-objective Bayesian optimization". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 10702–10708.

[292] Paul MJ Van Den Hof and Ruud JP Schrama. "Identification and control—closed-loop issues". In: *Automatica* 31.12 (1995), pp. 1751–1770.

[293] J Vanlier et al. "Parameter uncertainty in biochemical models described by ordinary differential equations". In: *Mathematical biosciences* 246.2 (2013), pp. 305–314.

[294] Joep Vanlier et al. "Optimal experiment design for model selection in biochemical networks". In: *BMC systems biology* 8.1 (2014), pp. 1–16.

[295] Pastora Vega et al. "Integrated design and control of chemical processes–Part I: Revision and classification". In: *Computers & chemical engineering* 71 (2014), pp. 602–617.

[296] Joannes Vermorel and Mehryar Mohri. "Multi-armed bandit algorithms and empirical evaluation". In: *European conference on machine learning*. Springer. 2005, pp. 437–448.

[297] Moritz Von Stosch et al. "Hybrid semi-parametric modeling in process systems engineering: Past, present and future". In: *Computers & Chemical Engineering* 60 (2014), pp. 86–101.

[298] K. P. Wabersich and M. Toussaint. "Automatic testing and minimax optimization of system parameters for best worst-case performance". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2015, pp. 5533–5539.

[299] Andreas Wächter and Lorenz T Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical programming* 106.1 (2006), pp. 25–57.

[300] Andreas Wächter and Lorenz T Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical programming* 106.1 (2006), pp. 25–57.

[301] Ziyu Wang et al. "Bayesian optimization in a billion dimensions via random embeddings". In: *Journal of Artificial Intelligence Research* 55 (2016), pp. 361–387.

[302] Peter E Wellstead. "Non-parametric methods of system identification". In: *Automatica* 17.1 (1981), pp. 55–69.

[303] Wolfgang Wiechert and Stephan Noack. "Mechanistic pathway modeling for industrial biotechnology: challenging but worthwhile". In: *Current opinion in biotechnology* 22.5 (2011), pp. 604–610.

[304] Christopher KI Williams and David Barber. "Bayesian classification with Gaussian processes". In: *IEEE Transactions on pattern analysis and machine intelligence* 20.12 (1998), pp. 1342–1351.

[305] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[306] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. "A data-driven approximation of the koopman operator: Extending dynamic mode decomposition". In: *Journal of Nonlinear Science* 25.6 (2015), pp. 1307–1346.

[307] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine Learning* 8.3 (1992), pp. 229–256.

[308] Anqi Wu, Mikio C Aoi, and Jonathan W Pillow. "Exploiting gradients and Hessians in Bayesian optimization and Bayesian quadrature". In: *arXiv preprint arXiv:1704.00060* (2017).

[309] Jian Wu. "Knowledge gradient methods for Bayesian optimization". PhD thesis. Cornell University, 2017.

[310] Jian Wu and Peter Frazier. "The parallel knowledge gradient method for batch Bayesian optimization". In: *Advances in neural information processing systems* 29 (2016).

[311] Jian Wu et al. "Bayesian optimization with gradients". In: *Advances in Neural Information Processing Systems* 30 (2017).

[312] Dongbin Xiu and George Em Karniadakis. "Modeling uncertainty in flow simulations via generalized polynomial chaos". In: *Journal of Computational Physics* 187.1 (May 2003), pp. 137–167. ISSN: 00219991.

[313] Dongbin Xiu and George Em Karniadakis. "The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations". In: *SIAM J. Sci. Comput.* 24.2 (2002), pp. 619–644. ISSN: 1064-8275.

[314] Wenjie Xu, Yuning Jiang, and Colin N Jones. "Constrained Efficient Global Optimization of Expensive Black-box Functions". In: *arXiv preprint arXiv:2211.00162* (2022).

[315] Masaki Yamawaki et al. "Multifunctional structural design of graphene thermoelectrics by Bayesian optimization". In: *Science advances* 4.6 (2018), eaar4192.

[316] Shu Yang et al. "Hybrid Modeling in the Era of Smart Manufacturing". In: *Computers & Chemical Engineering* (2020), p. 106874.

[317] Zhihong Yuan et al. "State-of-the-art and progress in the optimization-based simultaneous design and control for chemical processes". In: *AIChE Journal* 58.6 (2012), pp. 1640–1659.

[318] Stelios H Zanakis and James R Evans. "Heuristic "optimization": Why, when, and how to use it". In: *Interfaces* 11.5 (1981), pp. 84–91.

[319] Dongda Zhang et al. "Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization". In: *Biotechnology and bioengineering* 116.11 (2019), pp. 2919–2930.

[320] Dongda Zhang et al. "Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization". In: *Biotechnology and Bioengineering* 116.11 (2019), pp. 2919–2930.

[321] Yan Zhang and Nikolaos V Sahinidis. "Uncertainty quantification in CO2 sequestration using surrogate models from polynomial chaos expansion". In: *Industrial & Engineering Chemistry Research* 52.9 (2013), pp. 3121–3132.

[322] M. Zhao and J. Li. "Tuning the hyper-parameters of CMA-ES with tree-structured Parzen estimators". In: *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*. 2018, pp. 613–618.

[323] Ciyou Zhu et al. "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization". In: *ACM Transactions on mathematical software (TOMS)* 23.4 (1997), pp. 550–560.

[324] Mengjia Zhu, Alberto Bemporad, and Dario Piga. "Preference-based MPC calibration". In: *2021 European Control Conference (ECC)*. IEEE. 2021, pp. 638–645.

[325] Olek C Zienkiewicz and Robert L Taylor. *The finite element method for solid and structural mechanics*. Elsevier, 2005.