REAL-TIME CONTROL AND OPTIMAL PLANT DESIGN FOR RENEWABLE ENERGY SYSTEMS USING BAYESIAN OPTIMIZATION

by

Ali Khayat Baheri Irani

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Mechanical Engineering

Charlotte

2018

Approved by:

Dr. Christopher Vermillion

Dr. Praveen Ramaprabhu

Dr. Scott Kelly

Dr. Wesley Williams

Dr. Navid Goudarzi

ABSTRACT

ALI KHAYAT BAHERI IRANI. Real-Time Control and Optimal Plant Design for Renewable Energy Systems Using Bayesian Optimization. (Under the direction of DR. CHRISTOPHER VERMILLION)

The overarching objective of this research is to use machine learning tools to address the design and real-time control of active systems, focusing specifically by tethered airborne wind energy and ocean current energy systems. In both applications, along with numerous other engineering applications, data is expensive to generate. In particular, generating new plant designs is costly, and any adjustments to the controller must be performed in an environment that is continually changing. To address these challenges, we leverage a data-driven optimization strategy called Bayesian Optimization, which is specifically tailored to optimization problems for which a model does not exist (necessitating expensive data collection) or must be controlled with expensive experiments. This dissertation extends the current state of the art in Bayesian Optimization to enable Bayesian Optimization to be performed in real-time, in a spatiotemporally-varying environment. The techniques described in this dissertation have been applied using real-data, to the real-time altitude and depth optimization of airborne wind and ocean current turbine energy systems, respectively. Furthermore, the techniques have been applied to the nested co-design (combined plant/controller design) of an airborne wind energy system.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Vermillion for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better advisor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my dissertation committee members: Dr. Ramaprabhu, Dr. Kelly, Dr. Williams, and Dr. Goudarzi for their insightful comments and encouragement.

Finally, but by no means least, thanks go to my wife, mum, and dad for unbelievable support. This dissertation would not have been possible without their warm love and continued patience. They are the most important people in my world and I dedicate this dissertation to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Introduction

This chapter discusses the motivation of this research, summarizes the main contributions, and outlines the structure of this thesis by giving a chapter-by-chapter overview.

1.1.1 Renewable Energy Systems: From Sky to Ocean

One of the most urgent challenges faced by society is the search for cheap renewable energy resources to displace finite and polluting fossil fuel resources. In fact, renewable energy sources such as wind, solar power, hydro-electric power and ocean current are a viable alternative to fossil fuels [3]. Among many types of renewable energy systems, in this thesis we focus on two specific systems, namely airborne wind energy (AWE) system and ocean current turbines (OCTs).

1.1.2 Airborne Wind Energy Systems

As one of the fastest growing energy technologies for the past decade, wind energy is one of the best candidate renewable resources for displacing traditional fossil sources [4]. However, today's wind turbines operate at heights of only about 150m or less, where wind is weaker and more variable than at higher altitudes. AWE systems represent a new paradigm for wind turbines in which the structural elements of conventional wind turbines are replaced with tethers and a lifting body (a kite, rigid

(a) Altaeros Buoyant Airborne Turbine

(b) Surf kite from ETH

(c) EnerKite

(d) Makani Power-GoogleX

Figure 1: Different types of Airborne Wind Energy systems

wing, or aerostat) to harvest wind power at significantly increased altitudes (typically up to 600m or more depending on regulatory limits). At those altitudes, winds are often 5-10 times stronger and more consistent than ground-level winds. Figure 1 shows several different types of AWE systems.

There are two major options for harnessing wind energy and generating electricity with an AWE system:

- Ground-based generation: Here, the generator is on the ground and connected to a winch that holds the cable(s). During the power generating phase, electricity is produced by flying the lifting body at a high speed, in a crosswind motion,

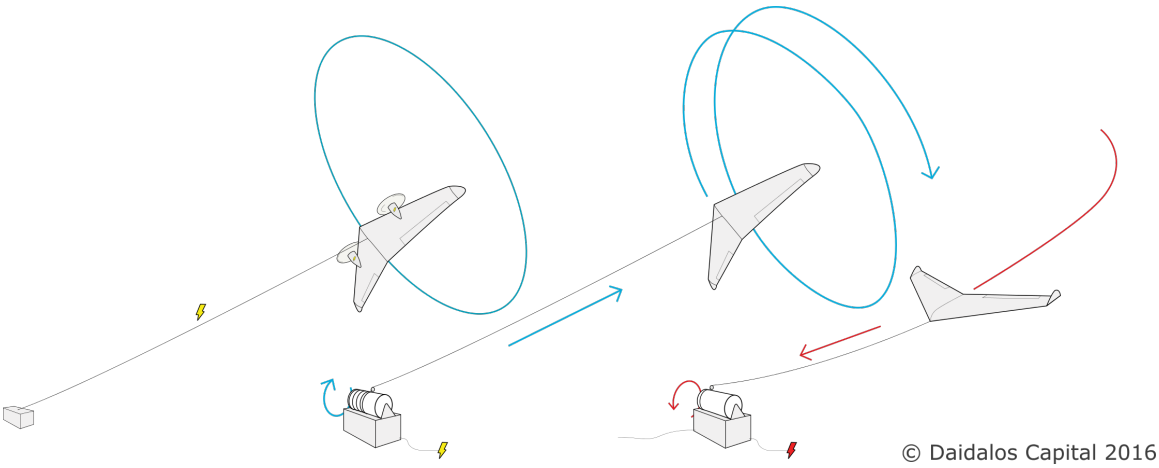


Figure 2: Electricity generation in the air (first picture) and on the ground (second and third picture). Image credit: [2]

generating a high lift force and consequently a high traction force on the cable(s).

- Airborne generation: Here, the generators are on board the lifting body. These are driven by rotors that are in turn driven by the wind speed. The energy is transferred to the ground through a conductive cable.

Inspired by the huge potential of AWEs, several academic and commercial groups have been developing AWEs and building working prototypes. Figure 3 illustrates the worldwide commercial and academic AWE research and development activity in 2013.

1.1.3 Ocean Current Turbine Array

Comparable to fossil fuels in terms of consistency and reliability, ocean currents are another candidate source of clean and renewable power. The oceans of the earth provide a vast source of renewable energy. The Gulf Stream, whose 1-2 m/s flow speeds carry over 65 million cubic meters per second of flow off the shores of Florida and North Carolina, possesses an estimated total power of 25 GW [67]. In general,

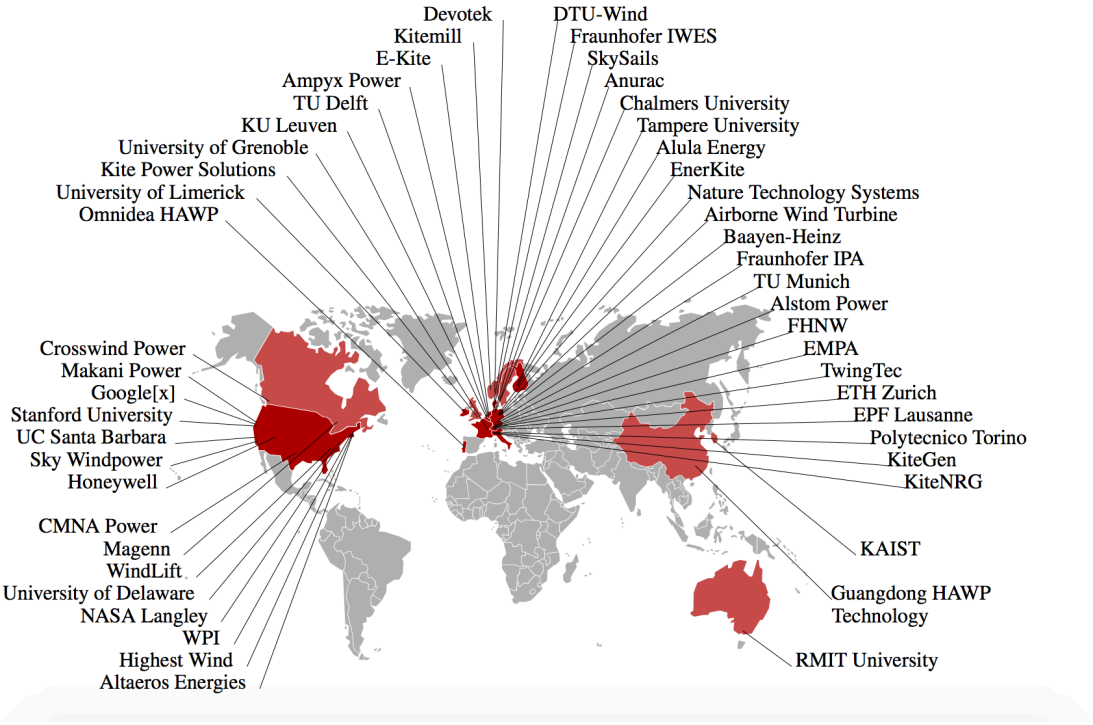


Figure 3: Companies and research institutes currently active in Airborne Wind Energy systems [8]

ocean energy can be divided into six forms: ocean wave, tidal range, tidal current, ocean current, ocean thermal energy, and salinity gradient [74]. Ocean current is caused by the gravitational pull of the moon and the sun (for the tidal current) and the temperature difference of ocean water (for the tropical circulation). In this thesis, we focus on ocean current turbines (OCTs) for our case studies. OCTs are hydro kinetic turbines that generate electricity from ocean current flow. Because of the deep waters where the strongest current resources can be found, most OCT concepts are comprised of tethers (rather than towers) and an underwater lifting body similar to AWE systems. Figure 4 shows some conceptual designs of OCTs.

1.2 Challenges

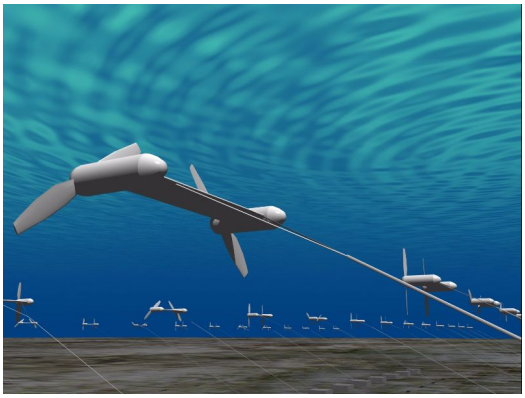
To achieve economic feasibility of both AWE and OCT systems, several important challenges need to be addressed. Many of these challenges relate to the fact that the economical realization of AWE and OCT systems involves the component-level and control system design of dynamically complex systems that operate in highly variable environments.

Broadly speaking, this thesis aims to address two main challenges:

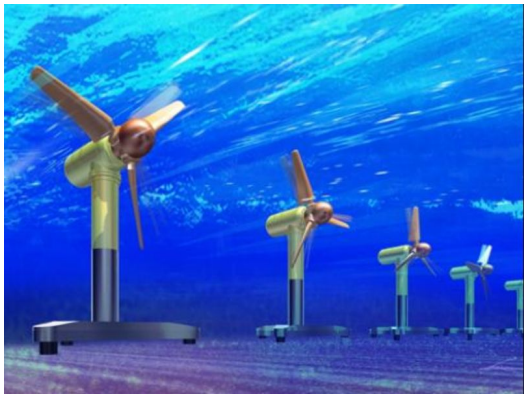
- In both the plant optimization and real-time control of AWE and OCT systems, data is expensive to generate. In particular, generating new plant designs is costly, and any in-situ adjustments to the controller must be performed in an environment that is continually changing. With respect to the latter (control) challenges, this means that any attempts to adapt the controller online must be performed more quickly than the time scales of the environment.

- There is a simultaneous plant optimization and real-time control challenge associated with both AWE and OCT systems. In particular, the plant and control design challenges have been shown to be coupled, as the optimal controller depends upon the plant and vice versa.

The ultimate goal of this thesis is to show how tools from machine learning can be used to introduce novel plant and control design mechanisms that address both aforementioned challenges. To achieve this goal, we present a data-driven optimization strategy called Bayesian Optimization. This optimization strategy is tailored



(a) Rendering of tethered ocean current turbines



(b) Rendering of underwater ocean current turbines

Figure 4: Conceptual design of OCTs. Image credit: [6]

toward instances where a closed-form expression for the objective function as a function of the decision variables is unobtainable, making expensive simulations and/or experiments necessary for convergence to an optimal design or control systems. In such situations, Bayesian Optimization is employed to find the optimum of an unknown and expensive-to-evaluate objective function. Bayesian Optimization lies at the heart of this thesis and is a unifying feature in both mentioned applications.

1.3 Renewable Energy Systems in Machine Learning Era

Arthur Samuel in 1959 defined machine learning as "the field of study that gives computers the ability to learn without being explicitly programmed" [61]. In the context of this thesis, when a physical model is not available or perfect, large amounts of amount of data can help to understand, model and predict physical behaviors of energy resources. In recent years, many researchers have found a great opportunity in applications of machine learning algorithms in the energy business. Nowadays, the energy-related data are broadly available [7]. These data carry statistically significant

information about how sources of energy are generated and/or consumed. Machine learning has proven to serve as promising tool for understanding the complex nature of renewable energy systems. In the existing literature, the application of machine learning to renewable energy systems mainly falls into the following categories [65]:

- **Forecasting renewable energy generation**

  Forecasting power output from a renewable energy power plant is very important. Machine learning is a promising tool for achieving this goal, which several machine learning tools considered in the literature. In this section we will review a few machine learning techniques used to forecast renewable energy generation [65]. Auto-regressive moving average (ARMA) and auto-regressive integrated moving average (ARIMA) models were presented in [48] for wind speed forecasting and wind power forecasting through the analysis of time-series data. The recurrent multi-layer perceptron model, a variant of ANN, was proposed in [57], and employs a Kalman filter based back-propagation network. In [59], a support vector machine (SVM) using Gaussian kernels was used to predict the wind speed. Fuzzy models are another way of using machine learning for prediction. In [30], a fuzzy model with a spatial correlation was used for wind power generation prediction. Artificial neural networks (ANNs) were used in [58] to build an equivalent model of a wind farm under normal operating conditions.

- **Determining plant location, size, and configuration**

  Renewable power plants continue to expand. It has been shown that determining their optimal sizes, locations and configurations will have a significant

impact on their performance. Several studies have been done to optimize the configuration of renewable energy systems such as solar photovoltaic (PV) plants and wind farms. For example, [71] optimized the configuration of power generating systems on an isolated island using a genetic algorithm. A flexible neuro-fuzzy approach for location optimization of solar plants was described in [13].

1.4 Thesis Contributions

The overarching objective of this research is to use machine learning tools to address plant and control system optimization challenges faced specifically by tethered wind (AWE) and ocean energy (OCT) systems. A unifying theme in the proposed research is the use of *Bayesian Optimization* to address these challenges. Bayesian Optimization is ideally suited for applications in which data is expensive to obtain. This conforms to our situation, since plant redesigns are expensive, and control system reconfigurations must be performed more quickly than the environmental conditions are changing. Because Bayesian Optimization plays a central role in the proposed research, Chapter 2 is dedicated to describing the mathematical details of Bayesian Optimization. Leveraging and expanding upon Bayesian Optimization tools, this thesis makes the following contributions:

1.4.1 Real-time control via Bayesian Optimization (Chapter 3)

We present a framework by which Bayesian Optimization can be used for real-time altitude optimization of an AWE system, for the purpose of maximizing net energy production. We then provide the details of how Bayesian Optimization is utilized

for an in-situ layout optimization of an OCT array. Because both real-time control problems involve an environment that is continually changing, we present a novel context-dependent Bayesian Optimization formulation where time is treated as the context in order to address the spatiotemporally varying nature of the environment. This mathematical formulation is accompanied by an original convergence result.

1.4.2 Combined plant and controller optimization (co-design) via Bayesian Optimization (Chapter 4)

We present a nested optimization framework that aims to solve the problem of coupling between plant and controller optimization for an AWE application. In the nested framework, adjustments to the plant design are performed iteratively, in batches, and control parameters are adjusted in real-time, during the experiments or simulations that are performed at each iteration. In this chapter, Bayesian Optimization is employed to solve the nested optimization problem at both the plant and controller design levels. Furthermore, we assess economies of scale as a function of batch size.

CHAPTER 2: BAYESIAN OPTIMIZATION

2.1 Introduction

In this chapter, our intent is to provide the mathematical details of Bayesian Optimization and the underlying Gaussian Process (GP) modeling tools, upon which the proposed algorithms in this thesis are built on.

2.2 Bayesian Optimization

Suppose that we want to maximize an unknown, black-box, and expensive-to-evaluate objective function. Due to the cost associated with evaluating this function, it is crucial to select the location of each new evaluation point deliberately. Bayesian Optimization is a powerful tool for finding the extrema of unknown and expensive-to-evaluate objective functions. If f is an unknown objective function defined by:

$$f(x) : \mathcal{X} \to \mathbb{R} \tag{1}$$

the goal of Bayesian Optimization is to find

$$x^* = \arg\max_{x \in \mathcal{X}} f(x) \tag{2}$$

However, if there is no analytical expression for $f(x)$, what can we do? Bayesian optimization solves this problem by maintaining a probabilistic model f and designing a so-called *acquisition function* to determine where to evaluate the function next.

Broadly speaking, Bayesian Optimization involves two steps. First, at each iteration, we update a model that characterizes our best guess of the objective function vs. design parameter, along with a quantification of the uncertainty in that guess. This phase of Bayesian Optimization is referred to as the *learning phase*. Second, we choose an *acquisition function*, which guides the optimization by determining the next point to evaluate. The selection of the next evaluation point in an effort to maximize the acquisition function is referred to as the *optimization phase*. In the rest of this chapter, we go into more details regarding each of these two steps.

2.2.1 Gaussian Process Regression

One area of supervised learning algorithms is regression, where the ultimate goal is to generalize an unknown function and predict its value over a continuous domain. As an attractive choice for non-parametric regression in machine learning, the ultimate goal of GP models is to find an approximation of a complex map from an input vector to the objective function value. In general, GP models are able to characterize complex phenomena, such as nonlinear effects and interactions between covariates, in a systematic fashion. GP models assume that the objective function values associated with different inputs follow a joint Gaussian distribution [68]. While this is not true of all objectives, it is often possible to apply a transformation to the objective function to more closely satisfy this assumption while preserving the location of the maximizer.

In general, a GP is fully specified by its mean function, $\mu(x)$, and covariance function, $k(x, x')$:

$$f(x) \sim \mathcal{GP}\Big(\mu(x), k(x, x')\Big). \tag{3}$$

The GP framework is used to predict the function value, $f(x^*)$, at a candidate design parameter x^*, based on a set of t past observations, $\mathcal{D}_{1:t} = \Big\{ x_{1:t}, f(x_{1:t}) \Big\}$. The unknown function value, $f(x^*)$, is modeled to follow a multivariate Gaussian distribution [68]:

$$\begin{bmatrix} f(x_{1:t}) \\ f(x^*) \end{bmatrix} \sim \mathcal{N}\Big(0, \begin{bmatrix} K_t + \sigma_\epsilon^2 I_t & k_t^T \\ k_t & k(x^*, x^*) \end{bmatrix}\Big), \tag{4}$$

where $f(x_{1:t}) = \Big\{ f(x_1), \cdots, f(x_t) \Big\}$ is the vector of observed function values. The vector $k_t(x) = [k(x^*, x_1), \cdots, k(x^*, x_t)]$ encodes the covariances between the candidate input, x^*, and the past data points, $x^{1:t}$. The past-data covariance matrix, with entries $[K_t]_{(i,j)} = k(x_i, x_j)$ for $i, j \in \Big\{ 1, \cdots, t \Big\}$, characterizes the covariances between pairs of past data points. The identity matrix is represented by I_t, and σ_ϵ represents the noise variance [68].

Individual elements of the covariance matrix, namely $k(x_i, x_j)$, encode the correlations between pairs of different evaluation points. In order to characterize this correlation in a simple, closed-form manner, a *covariance kernel* is used. This covariance kernel provides a relatively simple parametric structure for the values of $k(x_i, x_j)$. A very commonly used kernel function is known as the Squared Exponential (SE) covariance kernel. For two observations, x_i and x_j, the SE kernel is parameterized as:

$$k(x_i, x_j) = \sigma_0^2 \exp\Big(-\frac{1}{2}(x_i - x_j)^T \Lambda^{-2} (x_i - x_j) \Big) + \sigma_\epsilon \delta_{ij}, \tag{5}$$

where δ_{ij} represents the Kronecker delta operator:

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \tag{6}$$

The quantitative correlation between different data points is characterized by *hyper-parameters*, which are denoted by $\theta = \Big\{\sigma_0, \Lambda, \sigma_\epsilon\Big\}$. Kernel hyper-parameters are identified by maximizing the marginal log-likelihood of the existing observed data, $\mathcal{D}$ [68]:

$$\theta^* = \arg\max_{\theta} \log\, p(f(x_t) \mid x^{1:t}, \theta), \tag{7}$$

where

$$\log\, p(f(x_t) \mid x^{1:t}, \theta) = \Big(-\frac{1}{2} f(x_t)^T K_t^{-1} f(x_t) - \frac{1}{2}\log \mid K_t \mid -\frac{t}{2}\log 2\pi \Big), \tag{8}$$

Once the hyper-parameters are optimized, the predictive mean and variance at x^*, conditioned on these past observations, are expressed as:

$$\mu_t(x^* \mid \mathcal{D}) = k_t(x)\Big(K_t + I_t\sigma_\epsilon^2\Big)^{-1} f(x_t)^T, \tag{9}$$

$$\sigma_t^2(x^* \mid \mathcal{D}) = k(x,x) - k_t(x)\Big(K_t + I_t\sigma_\epsilon^2\Big)^{-1} k_t^T(x), \tag{10}$$

2.2.2 Use of Acquisition Functions for the Optimization Phase of Bayesian Optimization

Bayesian Optimization bases its choice of the next operating point on the maximization of an *acquisition function*. Specifically, the acquisition function uses the predictive mean and variance $\big($Eqs. (9-10)$\big)$ to combine exploration (visiting high-

variance regions) and exploitation (visiting high-mean regions) in selecting the next operating point. We denote the acquisition function by $\alpha(.)$ in this thesis.

To-date, several acquisition function structures have been proposed in the literature, including probability of improvement, expected improvement, upper confidence bound, information gain, and entropy-based improvements. All acquisition functions have a common theme: they incorporate both the mean, μ_t, and the variance, σ_t^2, of the GP model prediction, in order to manage the trade-off between exploration and exploitation. In the rest of this chapter, we will introduce three acquisition function structures.

- **Maximum Probability of Improvement (MPI)**

The first and most intuitive acquisition function proposed for Bayesian Optimization is Maximum Probability of Improvement (MPI) [55]. This acquisition selects the next operating point as the one that is most likely to improve upon the best value of objective function value so far. Mathematically, the acquisition function takes the following form:

$$\alpha_{PI}(x) \triangleq p\Big(f(x) \geq f(x)^{\max}\Big), \tag{11}$$

where $f(x)^{\max}$ represents the best value of objective function so far. Thus, the MPI acquisition function can be expressed as:

$$\alpha_{PI}(x) = \phi\Big(\frac{\mu_t(x) - f(x)^{\max}}{\sigma_t(x)}\Big), \tag{12}$$

where $\phi(.)$ denotes the normal cumulative distribution function (CDF). The point

Algorithm 1 Bayesian Optimization (BO)

procedure GENERIC VERSION OF BAYESIAN OPTIMIZATION
    $\mathcal{D} \leftarrow$ *Initialize*: $\{x_{1:2}, f(x_{1:2})\}$
    for each iteration **do**
        Train a GP model from $\mathcal{D}$
        Compute mean and variance of GP (Eqs. 9-10)
        Compute acquisition function (see Sec. 2.2.2)
        Find x^* that optimizes acquisition function
        Append $\{x^*, f(x^*)\}$ to $\mathcal{D}$
    end for
end procedure

with the highest probability of improvement is selected as the best subsequent point to evaluate.

- **Expected Improvement (EI)**

In many practical applications, the MPI acquisition function results in local optima, since it does not take into account the size of the improvement in its formulation. As an alternative to the MPI acquisition function, the Expected Improvement (EI) acquisition function [25, 50] considers the *amount* of anticipated improvement when choosing the next operating point. To construct the EI acquisition function, the improvement function is defined by:

$$I(x) \triangleq \max\left\{0, f(x) - f(x)^{\max}\right\}, \tag{13}$$

If the prediction is higher than the best value of the objective function so far, then the improvement is positive. Otherwise, it is set to zero. The inability for the acquisition function to assume negative values reflects the fact that if the performance *worsens* from one iteration to the next, then it is possible to simply revert to the

previous best point. This incentivizes exploration, since the system is not "punished" for testing a new design point that may lead to poor performance. The EI acquisition function is given by:

$$\alpha_{EI}(x) = \mathbb{E}(I(x)) = \mathbb{E}[\max\Big\{0, f(x) - f(x)^{\max}\Big\}], \tag{14}$$

which can be expressed as:

$$\alpha_{EI}(x) = \int_{f(x)^{\max}}^{\infty} \underbrace{\max\Big\{0, f(x) - f(x)^{\max}]\Big\}}_{\text{value of improvement}} \cdot \underbrace{\phi\Big(\frac{\mu_t(x) - f(x)^{\max}}{\sigma_t(x)}\Big)}_{\text{probability of improvement}} \mathrm{d}P \tag{15}$$

The first term in the integrand of Eq. (15) is the amount of improvement, and the second term represents the probability of that improvement. Calculating the integral of Eq. (15) using integration by parts leads to an expression for the EI acquisition function [51] that is given by:

$$\alpha_{EI}(x) = \begin{cases} \Big(\mu_t(x) - f(x)^{\max}\Big)\Phi(Z) + \sigma_t(x)\phi(Z), & \sigma_t(x) > 0 \\ 0, & \sigma_t(x) = 0 \end{cases} \tag{16}$$

where:

$$Z = \frac{\mu_t(x) - f(x)^{\max}}{\sigma_t(x)}, \tag{17}$$

and $\Phi(.)$ denotes the probability density function (PDF) for the normal distribution. The next operating point is chosen as the one that maximizes α_{EI}.

- **Upper Confidence Bound (UCB)**

The final candidate acquisition function considered in this thesis is known as the *upper confidence bound* (UCB) acquisition function [29], which incorporates the mean

and variance *explicitly* at a particular location:

$$\alpha_{UCB}(x) \triangleq \mu_t(x) + \sqrt{\beta_t}\sigma_t(x), \tag{18}$$

β_t plays a crucial role in Eq. (18), as it determines the trade-off between exploration and exploitation. A special case of the UCB acquisition is GP-UCB, where β_t at each iteration is automatically set according to [73]:

$$\beta_t = 2\log\Big(\frac{|D|t^2\pi^2}{6\delta}\Big). \tag{19}$$

Here, t is the number of past evaluations of the objective function, $\delta \in (0, 1)$ is a calibration parameter, and D is the number of decision variables.

We summarize the conventional Bayesian Optimization in Algorithm 1. The algorithm is initialized by two previously evaluated observations and corresponding function values (line 2). Then, at each step, a GP model is trained (line 4) to compute the predictive mean and variance (line 5). These statistical quantities are used to construct the acquisition function (line 6). Next, the point that maximizes the acquisition function is selected as the next operating point (line 7). Finally, data from the chosen operating point is added to the historical data (line 8), and the process is repeated.

CHAPTER 3: REAL-TIME CONTROLLER DESIGN USING BAYESIAN OPTIMIZATION

3.1 Introduction

This chapter provides the details of how Bayesian Optimization can be tailored to solve real-time optimal control problems, focusing specifically on altitude optimization of an AWE system and in-situ layout optimization of an OCT array.

In the first application, the objective is to adjust altitude to maximize net energy production in a spatiotemporally-varying wind environment. In the second application, the objective is to use tethers and control surfaces to adjust the layout of an array OCTs to maximize total power output of the array in a varying flow conditions. To accomplish these objectives, we have extended the conventional Bayesian Optimization algorithm of Chapter 2 to capture time-dependent patterns of the objective function. Because this adjustment to the algorithm is a unifying feature in the two applications, we first describe the adjustment in Section 3.2. We accompany this description with an original convergence result. We then detail the two applications in Section 3.3 and 3.4.

3.2 Context-Dependent Bayesian Optimization

Time-varying objective functions are common in many Parameters Estimation for State-of-Power (SOP) Prediction of Lithium-ion Batteries: A Bayesian Optimization Approach engineering applications, including the AWE and OCT systems considered

Algorithm 2 Context-Dependent Bayesian Optimization (CDBO)

procedure CONTEXT-DEPENDENT BAYESIAN OPTIMIZATION (CDBO)
  $\mathcal{D} \leftarrow$ *Initialize*: $\big\{(s_{1:2}, z_{1:2}), f(s_{1:2})\big\}$
  for each time step **do**
    Set the context into the current time
    Construct a composite kernel
    Train a GP model from $\mathcal{D}$
    Compute predictive mean and variance of GP
    Compute acquisition function
    Find s^* that optimizes acquisition function
    Append $\big\{(s^*, z_{\text{current context (time)}}), f(s^*)\big\}$ to $\mathcal{D}$
  end for
end procedure

in this thesis. The time-varying nature of our objective function compels us to modify the conventional Bayesian Optimization method. In fact, the generic version of Bayesian Optimization presented in Chapter 2 assumes that the objective function's dependence on the control variables does not change during the optimization procedure. One way to cope with the dynamic nature of objective function is through contextual GP modeling [54]. Contextual GP modeling allows for the possibility that the objective function depends on both the decision variable(s) and *other environmental variables* (termed the "context"). To deal with time-varying nature of our objective function, we consider *time* as the context. With generic Bayesian Optimization setting (Algorithm 1), the kernel of the GP is defined only in terms of decision variable(s). With contextual GP, the kernel is also allowed to vary as a function of the context (time):

$$k\Big((s, z), (s', z')\Big) = k_s(s, s') + k_z(z, z'), \tag{20}$$

where:

$$k_s(s, s') = \sigma_{0s}^2 \exp\Big(-\frac{1}{2}(s - s')^T \Lambda_s^{-2}(s - s')\Big) \tag{21}$$

and

$$k_z(z, z') = \sigma_{0z}^2 \exp\Big(-\frac{1}{2}(z - z')^T \Lambda_z^{-2}(z - z')\Big). \tag{22}$$

In Eq. (57) and (58), $\theta_s = \Big\{\sigma_{0s}, \Lambda_s\Big\}$ and $\theta_z = \Big\{\sigma_{0z}, \Lambda_z\Big\}$ denote the hyper-parameters of k_s and k_z, respectively.

Context-Dependent Bayesian Optimization is summarized in Algorithm 2. First, at the very first two time steps, the procedure is initialized by two (arbitrary) candidates and the corresponding function values (line 2). Then, at each time step, the context (time) is fixed at the current instance (line 4). Next, the composite kernel is constructed based upon Eq. (20) (line 5). The GP model, along with its predictive mean and variance, is computed (line 6-7). Then, the acquisition is computed by one of the methods outlined in Section 2.2.2, and optimization is done only over the decision variable(s) (line 8-9). Finally, the selected candidate and the current time are augmented to the data (line 10), and the process is restarted.

3.2.1 Convergence Rate of Contextual Expected Improvement

In this section, we derive a regret bound for contextual EI acquisition function. Regret refers to the difference between the optimal objective function value and its current value.

In each iteration, we receive a context $z_t \in Z$ from a set of Z context and our goal is to identify an action (i.e., decision variable) $s_t \in S$. The function value (i.e., payoff) is $\mathbf{y}_t = f(\mathbf{s}, \mathbf{z}_t) + \epsilon_t$, where $f : S \times Z \to \mathbb{R}$ is an unkwon function and ϵ_t is zero random

noise. We use $X = S \times Z$ to refer to the set of all action-context pairs. We choose *regret* as a metric to measure the performance of Bayesian Optimization. Specifically, the cumulative regret, R_T, after T iterations is the sum of the instantaneous regret, r_t:

$$R_T = \sum_{t=1}^{T} r_t \tag{23}$$

where $r_t = f(\mathbf{s}^*, \mathbf{z}_t) - f(\mathbf{s}, \mathbf{z}_t)$. Furthermore, we use the *information gain* concept to measure the reduction in uncertainty (measured in terms of differential Shannon entropy) in the underlying function gained by revealing a set of observations, $\mathbf{y}_A$:

$$\mathcal{I}\Big(\mathbf{y}_A; f(.)\Big) = H(\mathbf{y}_A) - H(\mathbf{y}_A | f(.)) \tag{24}$$

Consequently, the *maximum information gain*, γ_T, after T decision rounds is defined by:

$$\gamma_T = \max_{A \subset S} \mathcal{I}\Big(\mathbf{y}_A; f(.)\Big) \tag{25}$$

Before presenting the main theory, first we introduce several lemmas.

Lemma 1. (Lemma 1 of [63]) Let $z = \frac{\mu_{t-1}(\mathbf{s},\mathbf{z}_t) - y_{t-1}^{max}}{\sigma_{t-1}(\mathbf{s},\mathbf{z})}$. The EI acquisition function can be expressed as $\alpha_t^{EI}(\mathbf{s}, \mathbf{z}_t) = \sigma_{t-1}(\mathbf{s}, \mathbf{z}_t)\tau\big(z_{t-1}(\mathbf{s}, \mathbf{z}_t)\big)$ and $\alpha_t^{EI}(\mathbf{s}, \mathbf{z}_t) \leq \tau\big(z_{t-1}(\mathbf{s}, \mathbf{z}_t)\big)$, where $\tau(z) = z\Phi(z) + \phi(z)$ with Φ and ϕ are the c.d.f and p.d.f of the standard normal distribution.

Lemma 2. (Lemma 2 of [63]) The EI acquisition function at the selected point should be positive for a valid optimization, i.e., $\forall X_t \in \mathcal{D}_t, \alpha_t^{EI}(\mathbf{s}, \mathbf{z}_t) \geq \kappa > 0$, where κ is a small positive constant.

Lemma 3. (Lemma 5.2 of [73]) Let $\beta_t = 2log\big(\frac{|S_t|\pi_t}{\delta}\big)$ and $\delta \in (0,1)$. Then

$$p\Big(\forall t, \forall X \in \mathcal{X}, |\mu_{t-1}(\mathbf{s},\mathbf{z}_t) - f(\mathbf{s},\mathbf{z}_t)| \leq {\beta_t}^{1/2}\sigma_{t-1}(\mathbf{s},\mathbf{z}_t)\Big) \geq 1-\delta \tag{26}$$

Lemma 4. (Lemma 9 of [82]) The improvement function $I_t(\mathbf{s},\mathbf{z}_t) = \Big\{0, f(\mathbf{s},\mathbf{z}_t) - y_{t-1}^{max}\Big\}$ and the acquisition function $\alpha_t^{EI}(\mathbf{s},\mathbf{z}_t) = \mathbb{E}[I_t(\mathbf{s},\mathbf{z}_t)]$ satisfy the following inequality: $I_t(\mathbf{s},\mathbf{z}_t) - \beta_t^{1/2}\sigma_{t-1}(\mathbf{s},\mathbf{z}_t) \leq \alpha_t^{EI}(\mathbf{s},\mathbf{z}_t)$.

Lemma 5. (Lemma 5.4 of [73]) The sum of the predictive variances is bounded by the maximum information gain γ_T. That is $\sum_{t=1}^{T}\sigma_{t-1}^2(\mathbf{s},\mathbf{z}_t) \leq \frac{2}{\log(1+\sigma^{-2})}\gamma_T$.

Lemma 6. (Lemma 8 of [63]) Let $\kappa > 0$ be a pre-defined threshold on the acquisition function $\alpha_{t-1}^{EI}(\mathbf{s},\mathbf{z}_t)$, if $y_{t-1}^{max} - \mu_{t-1}(\mathbf{s},\mathbf{z}_t) > 0$. We then have $y_{t-1}^{max} - \mu_{t-1}(\mathbf{s},\mathbf{z}_t) \leq \sigma_{t-1}(\mathbf{s},\mathbf{z}_t)C^{1/2}$ where $C \triangleq \log[\frac{1}{2\pi\kappa^2}]$.

Lemma 7. (Lemma 9 of [63]) Let $\kappa > 0$ be a pre-defined threshold, $z = \frac{\mu_{t-1}(\mathbf{s},\mathbf{z}_t) - y_{t-1}^{max}}{\sigma_{t-1}(\mathbf{s},\mathbf{z}_t)}$ and $\tau(z) = z\Phi(z) + \phi(z)$, we have $\tau(-z_{t-1}(\mathbf{s},\mathbf{z}_t)) \leq 1 + C^{1/2}$ where $C \triangleq \log[\frac{1}{2\pi\kappa^2}]$.

***Theorem* 1**. Let $\kappa > 0$ be a pre-defined threshold, γ_T be the maximum information gain for the squared exponential kernel, σ^2 be the measurement noise variance, $C \triangleq \log[\frac{1}{2\pi\kappa^2}]$, $\beta_t = 2log\big(\frac{|S_t|\pi_t}{\delta}\big)$, where $\sum \pi_t^{-1} = 1, \pi_t > 0$, and $\delta \in (0,1)$. Then, with a probability of at least $1-\delta$, the cumulative regret for contextual EI obeys the following rate:

$$R_T \leq \sqrt{\frac{2T\gamma_T}{\log(1+\sigma^{-2})}}\Big(\sqrt{3(\beta_T + 1 + C)} + \beta_T^{1/2}\Big) \tag{27}$$

Proof:

The roadmap for our proof is analogous to the proof of Theorem 1 of [63], which drives a convergence rate for EI acquisition function (without context). Using the results from the legacy contextual GP paper ([54]), our results extend the previous work of [63] in the presence of context.

Proof.

$$r_t = f(\mathbf{s}^*, \mathbf{z}_t) - f(\mathbf{s}, \mathbf{z}_t) = f(\mathbf{s}^*, \mathbf{z}_t) - y_{t-1}^{max} + y_{t-1}^{max} - f(\mathbf{s}, \mathbf{z}_t) \tag{28}$$

Now, we bound r_t. Using Lemma 3 for the first two terms of Eq. (28), we write:

$$\begin{aligned} f(\mathbf{s}^*, \mathbf{z}_t) - y_{t-1}^{max} \leq \alpha^{EI}(\mathbf{s}^*, \mathbf{z}_t) + \beta^{1/2}\sigma_{t-1}(\mathbf{s}^*, \mathbf{z}_t) \leq \alpha^{EI}(\mathbf{s}, \mathbf{z}_t) + \beta^{1/2}\sigma_{t-1}(\mathbf{s}^*, \mathbf{z}_t) \\ = \sigma_{t-1}(\mathbf{s}, \mathbf{z}_t)\tau(z_{t-1}(\mathbf{s}, \mathbf{z}_t)) + \beta_t^{1/2}\sigma_{t-1}(\mathbf{s}^*, \mathbf{z}_t) \end{aligned} \tag{29}$$

For the last two terms of Eq. 28, $y_{t-1}^{max} - f(\mathbf{s}, \mathbf{z}_t)$, we write:

$$\begin{aligned} y_{t-1}^{max} - f(\mathbf{s}, \mathbf{z}_t) &= y_{t-1}^{max} - \mu_{t-1}(\mathbf{s}, \mathbf{z}_t) + \mu_{t-1}(\mathbf{s}, \mathbf{z}_t) - f(\mathbf{s}, \mathbf{z}_t) \\ &\leq \sigma_{t-1}(\mathbf{s}, \mathbf{z}_t)(-z_{t-1}(\mathbf{s}, \mathbf{z}_t)) + \sigma_{t-1}(\mathbf{s}, \mathbf{z}_t)\beta_t^{1/2} \\ &= \sigma_{t-1}(\mathbf{s}, \mathbf{z}_t)[\tau(-z_{t-1}(\mathbf{s}, \mathbf{z}_t)) + \beta_t^{1/2} - \tau(z_{t-1}(\mathbf{s}, \mathbf{z}_t))] \end{aligned} \tag{30}$$

Thus, we can write:

$$r_t \leq \sigma_{t-1}(\mathbf{s}, \mathbf{z}_t)[\beta_t^{1/2} + \tau(z_{t-1}(\mathbf{s}, \mathbf{z}_t))] + \beta_t^{1/2}\sigma_{t-1}(\mathbf{s}^*, \mathbf{z}_t) \tag{31}$$

Using the bound of $\tau(-z_{t-1}(\mathbf{s}, \mathbf{z}_t))$ in Lemma 7, we obtain

$$r_t \leq \underbrace{\sigma_{t-1}(\mathbf{s}, \mathbf{z}_t)(\beta_t^{1/2} + 1 + C^{1/2})}_{L_t} + \underbrace{\beta_t^{1/2}\sigma_{t-1}(\mathbf{s}^*, \mathbf{z}_t)}_{U_t} \tag{32}$$

Using Cauchy-Schwartz inequality, $\Big((a+b+c)^2 \leq 3(a^2+b^2+c^2)\Big)$, we bound L_t:

$$\sum_{t=1}^{T} L_t^2 \leq \sum_{t=1}^{T} \sigma_{t-1}^2(\mathbf{s}, \mathbf{z}_t)(\beta_T^{1/2}+1+C^{1/2})^2 \leq 3(\beta_T+1+C)\sum_{t=1}^{T} \sigma_{t-1}^2(\mathbf{s}, \mathbf{z}_t) \leq \frac{6(\beta_T+1+C)\gamma_T}{\log(1+\sigma^{-2})} \tag{33}$$

Therefore

$$\sum_{t=1}^{T} L_t \leq \sqrt{T}\sqrt{\sum_{t=1}^{T} L_t^2} \leq \sqrt{\frac{6T(\beta_T+1+C)\gamma_T}{\log(1+\sigma^{-2})}} \tag{34}$$

Using Lemma 5 we bound U_t

$$\sum_{t=1}^{T} U_t \leq \beta_T \sum_{t=1}^{T} \sigma_{t-1}(\mathbf{s}^*, \mathbf{z}_t) \leq \sqrt{\frac{2T\beta_T\gamma_T}{\log(1+\sigma^{-2})}} \tag{35}$$

Finally, the cumulative regret bound, $\Big(R_T \leq \sum_{t=1}^{T}(L_t + U_t)\Big)$, can be expressed as:

$$R_T \leq \sqrt{\frac{2T\gamma_T}{\log(1+\sigma^{-2})}}\Big(\sqrt{3(\beta_T+1+C)} + \beta_T^{1/2}\Big) \tag{36}$$

which completes the proof. □

3.3 Real-Time Altitude Optimization Using Bayesian Optimization

We now present a framework by which Bayesian Optimization can be used for real-time altitude optimization of an AWE system, for the purpose of maximizing net energy production. In particular, we leverage the extension of context-dependent Bayesian Optimization to capture time-dependent patterns of a realistic wind shear profile.

3.3.1 Background and Literature Survey

AWE systems provide control degrees of freedom that allow the systems to *adjust* their operating altitudes and intentionally induce *crosswind motions* to enhance

power output. To-date, the majority of research in the area of AWE system control has focused on the latter problem, i.e., crosswind motion control [79, 38, 27, 49, 83]. Comparatively fewer studies have focused on the impact of adjusting altitude [77, 15]. Even within the smaller body of literature that addresses altitude optimization, many of the results in the literature make greatly simplifying but often unrealistic assumptions. For example, [79, 38] make the assumption of a monotonic wind profile that conforms to a power law model, an assumption that greatly simplifies the altitude optimization problem but is seldom satisfied in the *instantaneous* wind shear (wind speed vs. altitude) profile (as we shall see in this chapter). A more recent publication, [15], does take into account the possibility of non-monotonic wind shear profiles but uses an extremum seeking (ES) formulation that only achieves convergence to *local*, rather than global, optima. The authors recently extended their work in [14] to consider more complex control architectures such as hierarchical structure and "globalized" extremum seeking to improve the overall performance of the AWE system. Still, the extremum seeking algorithms only leverage recent data in adjusting the amplitude of the perturbation signal.

Only recently has any effort been undertaken to take into account the *stochastic* nature of the wind shear profile in the context of a *global* altitude optimization. An initial attempt to address this problem using model predictive control (MPC) is detailed in [24]. In the context of the altitude optimization problem, MPC attempts to balance exploration with exploitation in optimizing the altitude setpoint. However, this MPC formulation requires an offline characterization of the statistical properties (conditional mean and conditional variance) of the wind shear (wind speed vs. al-

titude) profile, thereby necessitating the collection of a very substantial amount of data offline in order to kick-start the optimization.

For the altitude optimization problem at hand, it is desirable to employ a control system that can learn the statistical properties of the wind shear profile online, thereby alleviating the need for offline data collection prior to running the control system. One of the most well-studied problems in the machine learning community is the design of optimization algorithms for a real-world applications using scarce data. In existing literature, this problem has been studied in the context of sequential decision-making problems where the goal is to learn the behavior of an objective function (called exploration) while simultaneously trying to maximize or minimize the objective function (called exploitation). Bayesian Optimization is a fitting approach for addressing this problem, as it blends exploration and exploitation in such a way that it finds the global optimum of an unknown, expensive-to-evaluate, and black-box function within only a few evaluations. As mentioned in Chapter 2, one popular approach is to model the unknown function as a Gaussian Process (GP), where Bayesian Optimization puts prior belief the overall structure of that objective function. At every step of Bayesian Optimization, the next operating point is selected to maximize some *acquisition function*, which characterizes (i) how much will be learned by visiting a candidate point (exploration) and (ii) what the likely performance level will be at that next candidate point (exploitation) [72], [25].

In the case of the altitude optimization problem at hand, the decision variable is the operating altitude at the next time step, denoted by z_{k+1}, where k is the current time step. The objective is net power output, taking under consideration the power

required to control the system at a given altitude and the power required to adjust the altitude. In this chapter, we focus specifically on the Altaeros Energies Buoyant Airborne Turbine (BAT), pictured in Fig. 5. The BAT maximizes power output by *hunting* for the optimal altitude.

3.3.2 Wind Shear Profile, Altitude Adjustment, and Energy Generation Models

To maximize generated energy, it is of interest to operate the AWE system at altitudes where the wind velocity, V_{wind}, is as close as possible to the rated wind speed, V_{rated}. Consistent with intuition, if $V_{\text{wind}} < V_{\text{rated}}$, the net power production will diminish because the turbine is producing less power. Less obviously, the net power production will also diminish whenever $V_{\text{wind}} > V_{\text{rated}}$ because more power will be expended controlling the system. While it is possible to install wind profiling equipment to continuously monitor V_{wind} as a function of altitude (z), such profiling equipment is very costly. Table 1 shows an estimated cost comparison between different wind profiling technologies, including fixed LiDAR and SODAR units, as well as a weather balloon that continually is raised and lowered through the domain of allowable altitudes. These cost estimates account for capital expenses, along with operational expenses that include the cost of the energy required to operate the system. The cost estimate for the weather balloon system includes not only the balloon, but also the on-board instrumentation and winch system that continually actuates the weather balloon. Given these costs, it is of economic interest to use the AWE system itself to periodically explore the wind shear environment, taking into account the spatiotemporal variability of the wind in deciding which altitude to explore next. In the case

of the Altaeros BAT (used as a case study in this work), the wind speed is measured through the use of an ultrasonic anemometer, as discussed in [80]. Anemometer measurements (which provide apparent wind speed) are used in conjunction with GPS data to compute the true wind speed. This alleviates the need for expensive equipment but only provides a wind speed measurement at the AWE system's operating altitude. This introduces partial observability to the problem, leading to a tradeoff between exploration and exploitation.

To conduct our study, we used data obtained from a Doppler radar-based wind profiler in Cape Henlopen State Park (located in Lewes, Delaware). The data includes wind speed at 30 minute intervals, at 48 altitudes up to $3000m$, over the course of one year [11]. Due to regulatory and technical constraints on the AWE system flight altitude, we used the portion of data involving altitudes up to roughly $1600m$ (one mile) above ground level for the study at hand.

Figure 5: Altaeros Buoyant Airborne Turbine (BAT), Image Credit: [1]

The progression of the wind shear profile over the course of one day, at 3 hour intervals, is shown in Fig. 6. One can infer from this figure that the wind profile at any given time does not follow a power law or logarithmic law wind profile (or even a monotonic trend in many instances). Thus, in order to optimize net energy

Table 1: 20-year cost comparison for different technologies of measuring wind shear profile

Technology	Capital cost	Annual operational cost	Total cost
SODAR	990,000	36,000	1,710,000
LiDAR	190,000	32,000	830,000
Weather Balloon + Winch System	50,000	1,000	70,000

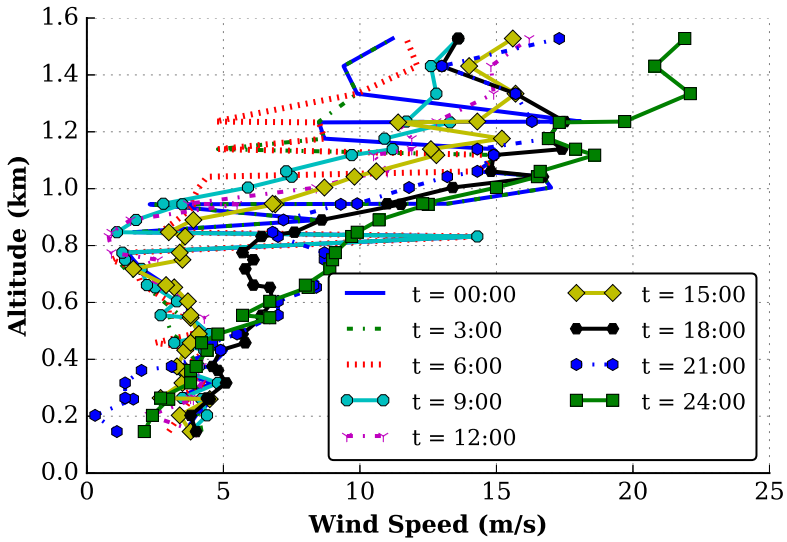


Figure 6: Sample wind shear profile based on actual data for one day from Cape Henlopen State Park, Lewes, Delaware, Date: March 1, 2014, [11]. The propagation of wind shear profile at 3 hour increments has been shown. One can conclude that (i) the optimal altitude varies over the course of the day and (ii) the wind shear profile does not fit a neat, monotonic relationship.

capture, it is imperative to employ a control scheme that is capable of coping with the uncertain nature of the wind shear profile.

3.3.3 Altitude Adjustment Model

In any AWE system, a low-level flight control system must regulate key flight variables, such as altitude, pitch angle, and roll angle, to prescribed setpoints. The block diagram of the low-level control structure used for the AWE system considered in this work (the Altaeros BAT) is shown in Fig. 7 and is described in detail in

[80]. Among other things, [80] demonstrates that the underlying flight control system can successfully control the BAT to prescribed altitude setpoints, denoted by z_{sp}. Because the efficacy of this lower-level controller has already been demonstrated in the literature and because of the relatively long time steps involved with altitude optimization, we will work with a very simple altitude adjustment model that assumes that *the altitude at the next step is equal to the altitude setpoint at the current time step*, i.e.:

$$z_{t+1} = z_t^{\text{sp}}. \tag{37}$$

Here, z_t^{sp} denotes the altitude setpoint at step t. Given the altitude tracking performance demonstrated in [80] and the long time step of 30 minutes that we will work with for the adjustment of z^{sp} in this work, this very simple model represents a reasonable approximation. In fact, a lower-level control has been designed in [80] for altitude tracking, and this lower-level controller operates on a much faster time step. Furthermore, we have evaluated the impact of different rate limitations (i.e., limitations on $z_{t+1}^{sp} - z_t^{sp}$) that ensure that any prescribed altitude adjustment can be made quickly (relative to the total 30 minute time step for altitude optimization) and the vast majority of time is spent generating energy at a prescribed altitude rather than adjusting altitude. The impact of different rate limitations on control results is studied in detail in Section 3.3.7.

3.3.4 Energy Generation Model

In this section, we formalize the objective function for altitude optimization. An appropriate objective function for altitude optimization should account for the energy

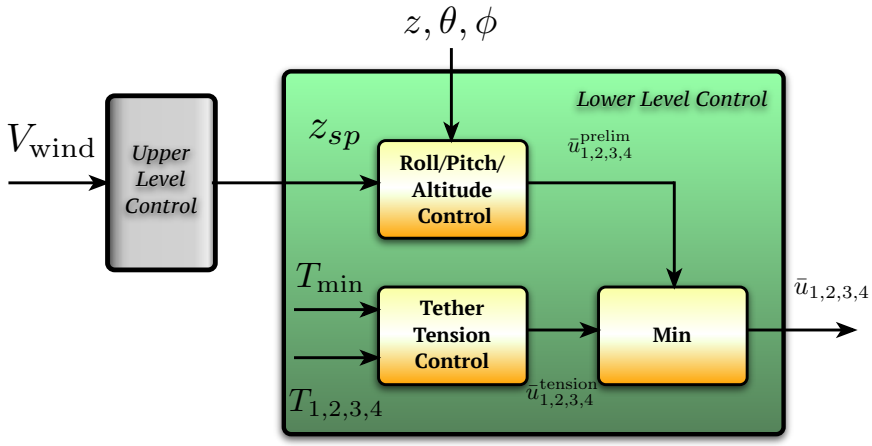


Figure 7: Lower-level controller Altaeros BAT, as detailed in [79]. $\bar{u}$ represents the vector of control inputs to the winches, whereas T_i denotes the tension in each tether.

generated by the turbine and the energy expanded adjusting and maintaining altitude (even when maintaining altitude, control energy must be expended to make small adjustments to the tethers in order to reject typical levels of turbulence). In order to account for each of these objectives, the instantaneous net power generation is expressed as [15]:

$$P_{total} = c_1 \min\left(V_{\text{wind}}, V_{\text{rated}}\right)^3 - c_2 V_{\text{wind}}^2 + P_z\left(V_{\text{wind}}, \dot{z}\right), \tag{38}$$

Generally, the power produced by a wind turbine can be modeled as:

$$P_{tur} = \frac{1}{2}\rho A {V_{\text{wind}}}^3 C_P, \tag{39}$$

where ρ is the air density, A is the rotor area, and C_P is the power coefficient. Accordingly, c_1 can be defined as:

$$c_1 = \frac{P_{\text{tur}}}{V_{\text{wind}}^3} = \frac{1}{2}\rho A C_P \tag{40}$$

To calculate c_2, it is important to consider that maintaining altitude requires the tethers to apply a total force that overcomes the lifting force on the BAT. The application of this force alone does not require mechanical energy. However, in the presence of turbulence (or any wind environment that is less than perfect), small adjustments are necessary to maintain altitude and attitude (orientation). The power required to make these adjustments is proportional to the tension in the tethers multiplied by the tether reel-in speed. The lifting force on the aerostat body is given by:

$$L = \frac{1}{2}\rho A_{\text{ref}} {V_{\text{wind}}}^2 C_L + F_b, \tag{41}$$

where L is the total lift force, A_{ref} is the reference area of the BAT, C_L is the lumped coefficient of lift, and F_b is the buoyant force. Under high wind conditions (under which the tether tensions are significant), F_b is much smaller than the aerodynamic lifting force, and the lifting force can be approximated by:

$$L = \frac{1}{2}\rho A_{\text{ref}} {V_{\text{wind}}}^2 C_L, \tag{42}$$

Since the power expenditure for maintaining altitude is proportional to this lifting force multiplied by the tether speed, that power expenditure, P_{maint}, can be expressed as:

$$P_{\text{maint}} = \frac{1}{2}\rho A_{\text{ref}} {V_{\text{wind}}}^2 V_{\text{ave}} C_L, \tag{43}$$

where V_{ave} is the average tether release speed over typical operation. Accordingly, c_2

can be defined as:

$$c_2 = \frac{P_{\text{maint}}}{{V_{\text{ave}}}^2} = \frac{1}{2}\rho A_{\text{ref}} V_{\text{ave}} C_L, \tag{44}$$

The power required to adjust altitude, P_z, is given as follows:

$$P_z\big(V_{\text{wind}}, \dot{z}\big) = \begin{cases} \frac{c_3}{\eta_m}\dot{z}V_{\text{wind}}^2 & \dot{z} \leq 0, \\ c_3\eta_g\dot{z}V_{\text{wind}}^2 & \text{otherwise.} \end{cases} \tag{45}$$

where η_m and η_g represent the efficiency of the motors that drive the winches and efficiency of the winches, respectively. The parameter c_3 is a constant that depends on the aerodynamics of the lifting body and its trim pitch angle. Because any upward motion of the AWE system must at some later time be compensated by downward movement (the AWE system must land eventually), [15] presents a simplified version of Eq. (38), which will be used in this work and is given by:

$$P = c_1 \min\left(V_{\text{wind}}(z), V_{\text{rated}}\right)^3 - c_2{V_{\text{wind}}}^2 - \bar{c}_3{V_{\text{wind}}}^2 \mid \dot{z} \mid . \tag{46}$$

Here, $\bar{c}_3$ is a lumped parameter that is dependent on overall winch efficiency and aerodynamics of the lifting body. In our discrete-time altitude optimization algorithm, we approximate $\mid \dot{z_k} \mid$ by:

$$\mid \dot{z_k} \mid = \frac{\mid z_k - z_{k-1} \mid}{\Delta t} \tag{47}$$

where Δt denotes the time step.

Ultimately, Eq. (46) serves as the reward function to be maximized. The AWE system power curve is shown in Fig. 8. This figure illustrates the power curve for a 100 kW variable-speed turbine, which is in line with the BAT design. One can

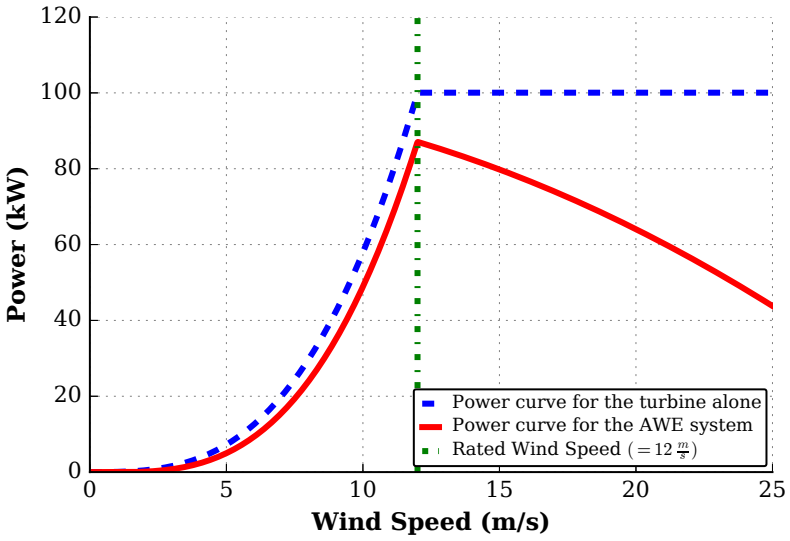


Figure 8: Comparison of power curve for the turbine alone and the full AWE system. The power production decreases for the wind velocity beyond V_{rated} due to increased energy required to control system to an altitude set-point in the presence of typical wind disturbances.

conclude from this figure that the power generated by the full AWE system is always less than the power generated by the turbine alone, due to the energy that must be used to control the system's altitude to a constant set-point, under typical turbulence levels. Moreover, unlike the conventional wind energy systems, the power generated decreases for wind speeds that exceed V_{rated}. This is the result of higher aerodynamic loads, which in turn requires more energy to control the system to an altitude set-point in the presence of typical turbulence levels. It should be noted that Fig. 8 is specific to scenarios where the AWE system is controlled to a constant setpoint. Additional energy losses exist when the altitude changes from one time step to the next.

3.3.5 Tailoring to Altitude Optimization

The ultimate goal of altitude optimization with Bayesian Optimization is to maximize the net energy generated by the BAT. Due to partially observable nature of wind shear profile, the optimal altitude in not known *a priori*. Hence, the optimization strategy must appropriately balance a trade-off between learning more about the objective function (exploration) and selecting an altitude that maximizes expected net power generation based on the collected data (exploitation). Bayesian Optimization serves as an attractive technique for achieving this tradeoff in an efficient manner, which is critically important since each evaluation of the objective function corresponds to an expensive experiment that requires physically relocating the AWE system to a new altitude.

As mentioned in Chapter 2, Bayesian Optimization involves two steps. First, at each iteration, we update a model that characterizes our "best guess" at the objective function vs. design parameter (i.e., altitude), along with a quantification of the uncertainty in that guess. This phase of Bayesian Optimization is referred to as the *learning phase*. Second, we choose an *acquisition function*, which guides the optimization by determining the next point (i.e., next altitude) to evaluate. The selection of the next altitude in an effort to maximize the acquisition function is referred to as the *optimization phase*.

3.3.5.1 Learning Phase: Using Gaussian Processes (GPs)

In the context of the altitude optimization problem, the GP framework is used to predict the generated power, $P(z)$, at candidate altitude z_c, based on a set of t

past observations, $\mathcal{D}_{1:t} = \Big\{ z_{1:t}, P(z_{1:t}) \Big\}$. In general, having $\mu(z) = 0$ significantly simplifies the training and prediction phase of a GP model [68]. To achieve this in the AWE application, we normalize the outputs of GP model using *z-score* normalization. Thus, the normalized power at step $t+1$, denoted by $\bar{P}_{t+1}(z)$, is expressed as:

$$\bar{P}_{t+1}(z) = \frac{P_{t+1}(z) - \mu^{1:t}}{\sigma^{1:t}} \tag{48}$$

where $\mu^{1:t}$ and $\sigma^{1:t}$ denote the mean and standard deviation of the population up to step t.

The normalized power at candidate altitude z_c, represented by $\bar{P}(z_c)$ in Eq. (49), for an unobserved altitude, z_c, is modeled to follow a multivariate Gaussian distribution [68]:

$$\begin{bmatrix} y_{1:t} \\ \bar{P}(z_c) \end{bmatrix} \sim \mathcal{N}\Big(0, \begin{bmatrix} K_t + \sigma_\epsilon^2 I_t & k_t^T \\ k_t & k(z_c, z_c) \end{bmatrix}\Big), \tag{49}$$

where $y_t = \Big\{ \bar{P}(z_1), \cdots, \bar{P}(z_t) \Big\}$ is the vector of observed function values. The vector $k_t(z) = [k(z_c, z_1), \cdots, k(z_c, z_t)]$ encodes the covariances between the candidate altitude, z_c, and the past data points, $z^{1:t}$. The past-data covariance matrix, with entries $[K_t]_{(i,j)} = k(z_i, z_j)$ for $i, j \in \Big\{ 1, \cdots, t \Big\}$, characterizes the covariances between pairs of past data points. The identity matrix is represented by I_t, and σ_ϵ represents the noise variance [68].

Individual elements of the covariance matrix, namely $k(z_i, z_j)$, encode the correlation between pairs of different altitudes. In order to characterize this correlation

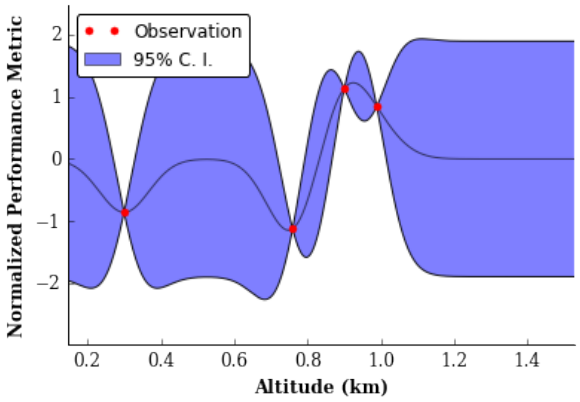


Figure 9: Gaussian Processes (GP) models can both make predictions and maintain a measure of uncertainty over those predictions. The 95% confidence of the model prediction is represented by blue area. Once an observation is made at a particular point, the uncertainty drops to zero. Recognizing that the wind shear profile is time-varying, Section 3.3.6 discusses how context-dependent Bayesian Optimization can be used to account for the fact that uncertainty increases in relation to the amount of time that has passed since an observation has been made (i.e., under context-dependent Bayesian Optimization, uncertainty at a location does not remain zero for all time just because that location was visited in the past).

in a simple, closed-form manner, a *covariance kernel* is used. This covariance kernel provides a relatively simple parametric structure for the values of $k(z_i, z_j)$. In this work, we represent the elements of the covariance matrix through a very commonly used kernel function known as the Squared Exponential (SE) covariance kernel. For two altitudes, z_i and z_j, the SE kernel is parameterized as:

$$k(z_i, z_j) = \sigma_0^2 \exp\Big(-\frac{1}{2}(z_i - z_j)^T \Lambda^{-2}(z_i - z_j)\Big) + \sigma_\epsilon \delta_{ij}, \tag{50}$$

where δ_{ij} represents the Kronecker delta operator:

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \tag{51}$$

We choose the SE kernel based on observation of key characteristics of the problem

at hand. Qualitatively, the SE kernel structure accounts for the fact that two measurements taken at close altitudes are more likely to be correlated than measurements taken at distant altitudes. Furthermore, the SE kernel accounts for the fact that the value of data decays exponentially with variations in time and altitude, which has been generally seen to be reflected in actual wind shear data. In fact, the assessment performed in [24] shows that the level of certainty in the wind speed decays exponentially with respect to both time and position.

The quantitative correlation between different data points is characterized by *hyper-parameters*, which are denoted by $\theta = \Big\{\sigma_0, \Lambda, \sigma_\epsilon\Big\}$. Kernel hyper-parameters are identified by maximizing the marginal log-likelihood of the existing observed data, $\mathcal{D}$ [68]:

$$\theta^* = \arg\max_{\theta} \log\, p(y_t \mid z^{1:t}, \theta), \tag{52}$$

where

$$\log\, p(y_t \mid z^{1:t}, \theta) = \Big(-\frac{1}{2} y_t^T K_t^{-1} y_t - \frac{1}{2}\log \mid K_t \mid -\frac{t}{2}\log 2\pi \Big) \tag{53}$$

Once the hyper-parameters are optimized, the predictive mean and variance at z^*, conditioned on these past observations, are expressed as:

$$\mu_t(z^* \mid \mathcal{D}) = k_t(z)\Big(K_t + I_t\sigma_\epsilon^2\Big)^{-1} y_t^T, \tag{54}$$

$$\sigma_t^2(z^* \mid \mathcal{D}) = k(z,z) - k_t(z)\Big(K_t + I_t\sigma_\epsilon^2\Big)^{-1} k_t^T(z), \tag{55}$$

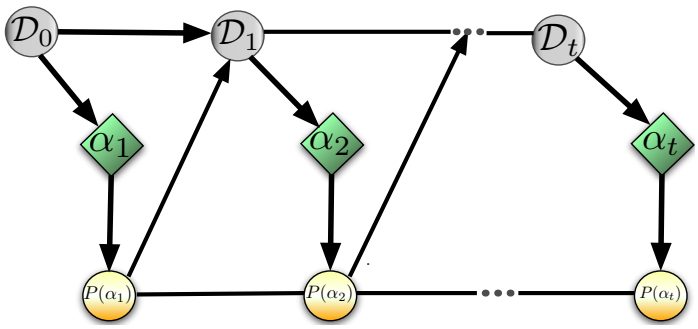


Figure 10: A graphical network (inspired from [47]) illustrating the Bayesian Optimization concept. Given some initial data, $\mathcal{D}_0$, Bayesian Optimization identifies the next best operating altitude, z_1, to evaluate. The objective function is then evaluated at this altitude, $P(z_1)$. Next, this pair is augmented to the previous data, the model is updated, and the process is repeated.

3.3.5.2 Acquisition Function

As mentioned before, Bayesian Optimization bases its choice of the next operating point on the maximization of an acquisition function, which guides the optimization by determining the next point to evaluate. Specifically, the acquisition function uses the predictive mean and variance $\big($Eqs. (54-55)$\big)$ to combine exploration (high-variance regions) and exploitation (high-mean regions) in selecting the next operating point.

3.3.6 Use of Context-Dependent Bayesian Optimization for Altitude Optimization

The time-varying nature of our objective function (resulting from variations in the wind shears profile) compels us to modify the conventional Bayesian Optimization method.

To address the challenge presented by the time-varying objective function, we leverage the context-dependent Bayesian Optimization formulation described in Section 3.2. For this purpose, we choose an addition of two kernels, one over altitude, k_z, and one

over time, k_t

$$k\Big((z_i,t_i),(z_j,t_j)\Big) = k_z(z_i,z_j) + k_t(t_i,t_j), \tag{56}$$

where:

$$k_z(z_i,z_j) = \sigma_{0z}^2 \exp\Big(-\frac{1}{2}(z_i - z_j)^T \Lambda_z^{-2}(z_i - z_j)\Big) + \sigma_{\epsilon z}\delta_{ij}, \tag{57}$$

and

$$k_t(t_i,t_j) = \sigma_{0t}^2 \exp\Big(-\frac{1}{2}(t_i - t_j)^T \Lambda_t^{-2}(t_i - t_j)\Big) + \sigma_{\epsilon t}\delta_{ij}. \tag{58}$$

In Eq. (57) and (58), $\theta_z = \Big\{\sigma_{0z}, \Lambda_z, \sigma_{\epsilon z}\Big\}$ and $\theta_t = \Big\{\sigma_{0t}, \Lambda_t, \sigma_{\epsilon t}\Big\}$ denote the hyper-parameters of k_z and k_t, respectively. These hyper-parameters describe how quickly the wind shear profile changes and how similar one altitude is to the next at a given time. The collection of hyper-parameters are automatically *tuned* (i.e., optimized) at each iteration of our algorithm, based on the data up to that time, using a maximum likelihood estimation criterion $\big($see Eq. (53)$\big)$.

3.3.7 Real-Time Altitude Optimization Results

We have evaluated the proposed altitude optimization algorithm by simulating the model in Sec. 3.3.4 based on available wind speed vs. altitude data [11]. We summarize the objective function parameters and design parameters in Tables 3 and 4, respectively. We compare the system performance under four different control scenarios:

- **Scenario 1: Off-line Optimum Fixed Altitude:** The first scenario involves flying the AWE at a *optimum fixed altitude*. We identify this optimal fixed altitude through off-line calculation of energy generation at different altitudes,

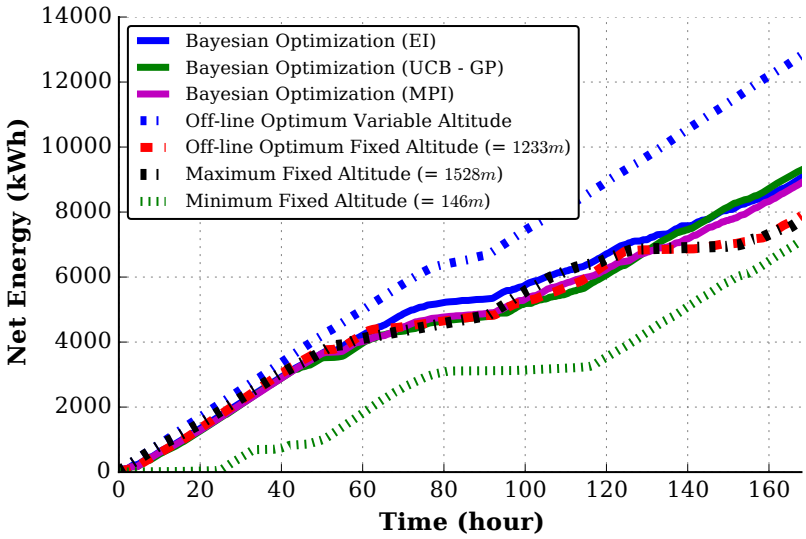
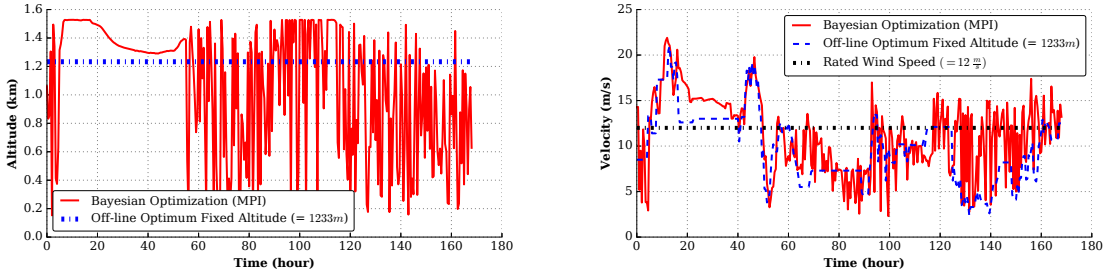


Figure 11: Comparison of net energy production over the course of one week. Bayesian Optimization (with different acquisition functions) outperforms off-line optimized fixed altitude, minimum fixed altitude, and maximum fixed altitude.



(a) Comparison of operating altitude

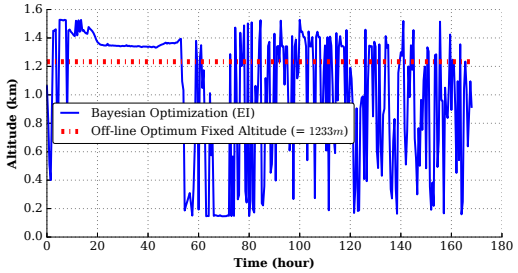
(b) Comparison of wind speeds at operating altitude

Figure 12: Comparison of operating altitude and wind velocity over the course of one week using Maximum Probability of Improvement (MPI) acquisition function

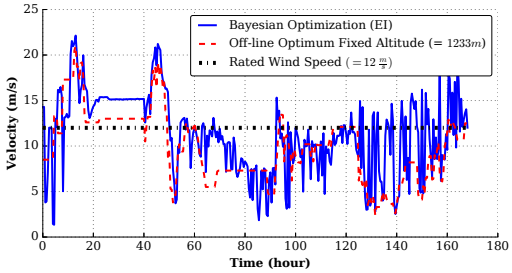
using all available wind data. Thus, the optimization takes the following form:

$$\underset{z=\text{constant}}{\text{maximize}} \quad \sum_{t=0}^{\tau} P_t(z)$$

where τ is the final time step. It is important to note that this calculation requires knowledge of the entire time window. Hence, this scenario serves an upper bound on the potential of any fixed altitude algorithm but is not imple-

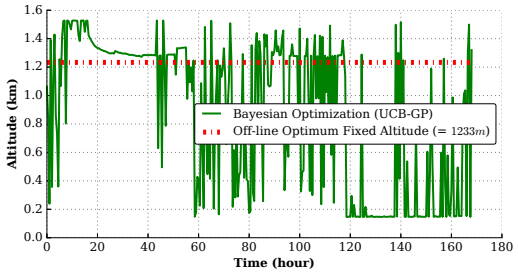


(a) Comparison of operating altitude

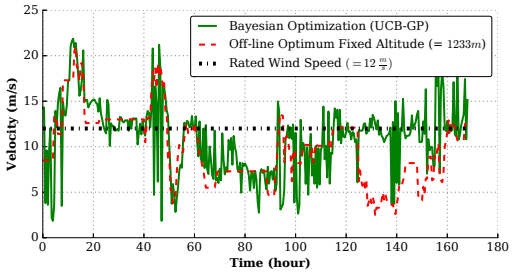


(b) Comparison of wind speeds at operating altitude

Figure 13: Comparison of operating altitude and wind velocity over the course of one week using Expected Improvement (EI) acquisition function

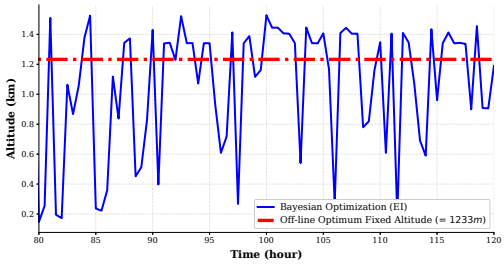


(a) Comparison of operating altitude

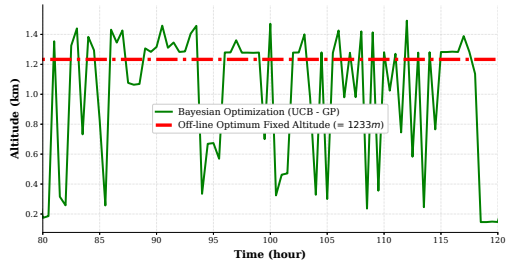


(b) Comparison of wind speeds at operating altitude

Figure 14: Comparison of operating altitude and wind velocity over the course of one week using Upper Confidence Bound (UCB-GP) acquisition function



(a) Zoomed operating altitude (EI)



(b) Zoomed operating altitude (UCB-GP)

Figure 15: Zoomed operating altitude for EI and UCB-GP acquisition function where the altitude exhibits very large fluctuations.

mentable in practice, since this would require knowledge of the future, along with measurements of the wind speed over all admissible altitudes. The off-line optimum fixed altitude was 1233m for the time period studied in this work.

- **Scenario 2: Minimum Fixed Altitude:** The second scenario is representative of conventional towered wind turbines. For this scenario, we rely on data from the lowest available altitude $146m$, which is close to the hub length of the world's tallest towered wind turbines [5].

- **Scenario 3: Maximum Fixed Altitude:** The third scenario involves flying the AWE at the maximum allowable fixed altitude, $1528m$.

- **Scenario 4: Bayesian Optimization:** The fourth scenario employs Context-Dependent Bayesian Optimization (CDBO), as outlined in Algorithm 2.

The design parameters used in the study are reported in Table 3. Bayesian Optimization is performed at each 30 minute time interval to identify the best next operating altitude.

Table 2 compares the performance of Bayesian Optimization (i.e., Scenario 4) to baseline scenarios (i.e., Scenario 1-3) for 10 randomly selected weeks in our dataset [17, 21, 22]. One can conclude form this table that our algorithm outperforms the other baseline scenarios in most instances. Furthermore, Fig. 27 illustrates the performance

Table 2: Comparison of net energy production for 10 randomly selected weeks

	Bayesian Optimization [kWh]	**Off-line optimum fixed altitude [kWh]**	**Maximum fixed altitude [kWh]**	**Minimum fixed altitude [kWh]**
Week 1	**9371**	8990	8688	6200
Week 2	**9068**	7896	7752	7111
Week 3	**8774**	7818	7809	7811
Week 4	**9441**	9374	9214	6913
Week 5	**8934**	8920	8442	7414
Week 6	**11306**	10867	7752	6786
Week 7	**8827**	8514	8278	7349
Week 8	10804	**11019**	10721	6888
Week 9	8706	**8714**	8693	7634
Week 10	9586	**9872**	9626	7060
Mean	**9481.7**	9198.4	8697.5	7116.6

Table 3: Design parameters for Bayesian Optimization algorithm

Parameter	Description	Value
P_{rated}	Rated power of turbine	100 kW
V_{rated}	Rated wind velocity	12 $\frac{m}{s}$
$z_{\min}$	Minimum allowable altitude	146 m
$z_{\max}$	Maximum allowable altitude	1528 m
Δt	Time step	30 min

of Bayesian Optimization over the course of one selected week, using each of the three candidate acquisition functions outlined in Section 2.2.2. One can conclude from this figure that all Bayesian Optimization methods result in superior energy production when compared to baseline scenarios. In this figure, we have also included a new scenario termed the off-line optimum variable altitude. This scenario serves as an upper bound on the achievable energy output that cannot be exceeded under any control strategy. It should be noted that, similar to the off-line optimum fixed altitude, this scenario cannot be implemented in real-time, as it requires the knowledge of future. The resulting operating altitudes and corresponding wind velocities are shown in Figs. 12-14. Fig. 15 provides a zoomed in version of Fig. 13 and 14 for the operating altitude where the altitude exhibits very large fluctuations.

Table 4: Cost function parameters

Parameter	Description	Value
c_1	Coefficient of power production related to the power curve	0.0579
c_2	Coefficient of power production required to maintain altitude	0.09
c_3	Coefficient of power production required to change altitude	0.15

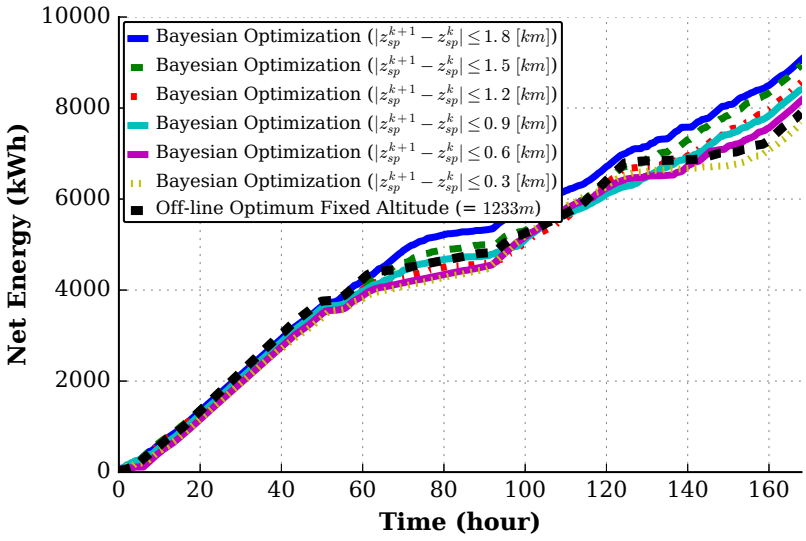


Figure 16: Comparison of net energy production where the altitude adjustment under a variety of rate limitations.

3.3.7.1 Exploration Rate Limitation

To account for different limitations on the tether release rate for the AWE system, Fig. 16 illustrates the net energy production of the AWE system under stricter rate limits than the one imposed in our initial result. One can infer from this figure that the most net energy is produced under the least severe rate limit, as expected. However, even strict rate limitations result in more energy production than constant altitude control strategies.

3.4 In-situ Layout Optimization of an OCT Array

Here, we show how Bayesian Optimization, in conjunction with a suite of low-level flight controllers, can be used to adjust the layout (staggering) of an array of OCTs. Bayesian Optimization is used to set target OCT locations, and tethers and control surfaces are used to adjust positions.

The proposed layout optimization algorithm has two key novel features. First, the location of each turbine is described through a small set of basis parameters; the number of basis parameters does not grow with increasing array size, thereby leading to an optimization that is not only computationally tractable but is also highly scalable. Secondly, context-dependent Bayesian Optimization algorithm is utilized to adjust these basis parameters as the flow environment varies.

3.4.1 Background and Literature Survey

The Gulf Stream, whose 1-2 m/s flow speeds carry over 65 million cubic meters per second of flow off the shores of Florida and North Carolina, possesses an estimated total power of 25 GW [67]. In response to this tremendous resource, several organizations (such as Florida Atlantic University, the University of North Carolina Coastal Studies Institute, Virginia Tech, and others) have investigated the design, control, and economic characterization of ocean current turbine (OCT) systems [75, 60, 76]. As with wind turbines, OCTs will ultimately need to be installed as part of large arrays (farms), such as the one in the schematic of Fig. 4, in order to create the economies of scale that render marine hydrokinetic energy economically viable. Unlike towered wind turbines and tidal turbines, however, Gulf Stream OCTs will be deployed in

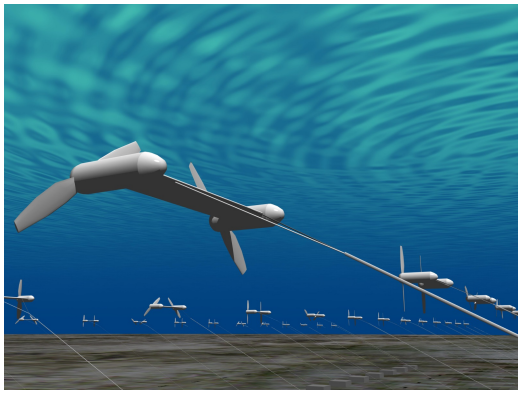


Figure 17: Conceptual design of OCT array, Image credit: [6]

deep waters that require the conventional tower to be replaced with *tethers* and a *lifting body* such as a wing. While the replacement of towers with tethers leads to dynamic modeling and control challenges (see [76]), it also provides the opportunity to adjust individual turbine positions within an array through the paying out/in of tether and manipulation of control surfaces (elevators and rudders). This ability to reconfigure the array layout *in-situ* can be very beneficial in the presence of a Gulf Stream resource profile that varies with respect to both space and time, as is shown to be the case in Fig. 18 [60].

To adjust the OCT layout online, it is necessary to employ an optimization technique that is not only effective but also enables fast convergence. A limited number of studies have focused on the layout optimization of marine turbine arrays. The existing literature generally falls into two categories:

1. Research that compares a small number of predetermined array designs but does not actually *optimize* the layout (see [62]);

2. Gradient-based optimization of the array configuration (see [44]). These algorithms generally converge quickly but easily get stuck in local optima.

A larger body of literature addresses the optimization of wind farm configurations. Unlike the aforementioned research in marine array layout optimization algorithms, many of the wind farm optimization schemes are *global* in nature. Some of the global techniques for wind farm optimization include extended pattern searches [35], particle swarm algorithms [28], particle filtering approaches [37], simulated annealing [23], evolutionary algorithms [56], and colony optimization [36]. These global optimizations

come at a computational price, however. For example, optimization of a 2D array of 47 turbines still required 61,802 simulations in [23]. While the aforementioned optimizations are effective for the offline design of wind farm layouts, they are not appropriate for the *online* optimization of tethered OCT locations.

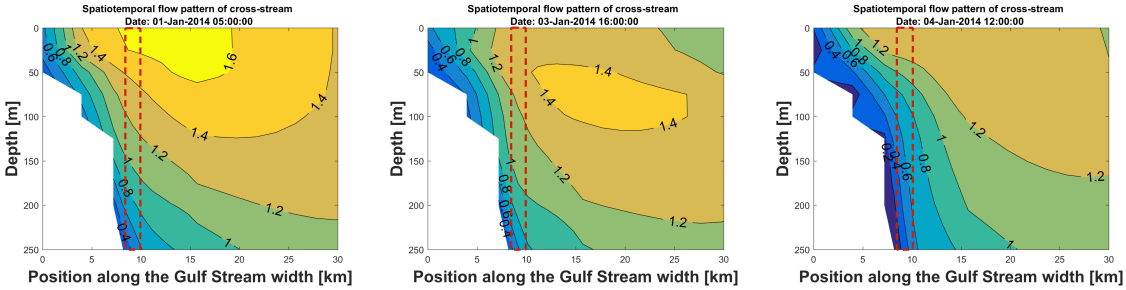
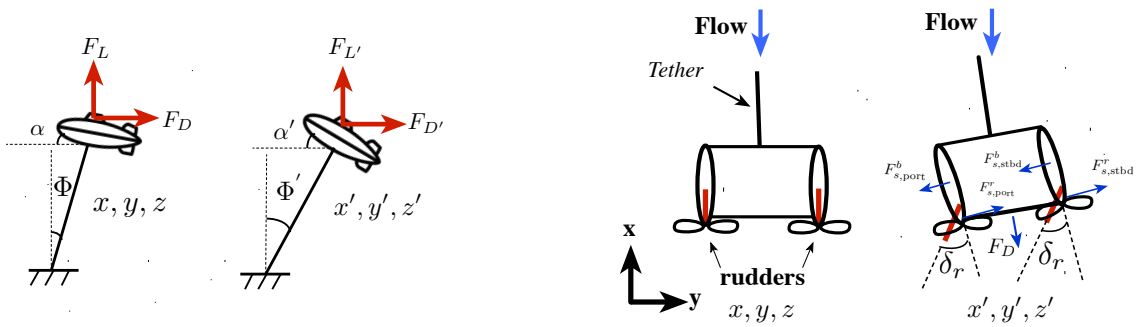


Figure 18: The Gulf Stream profile continuously varies over time. This trend motivates us to research methods by which we can reconfigure the array layout online to accommodate the most recent flow profile. It is important to note, however, that any iterative optimization process for doing so needs to happen quickly relative to the rate of change in the flow profile. In these figures, the contours represent flow velocities. Furthermore, a potential constraint volume for an OCT array is enclosed in dashed lines. As can be seen, the flow profile within this (or practically any) constraint volume varies over time.

The majority of the aforementioned studies focus on 2D modeling and optimization of array configurations. This approach makes sense for river arrays, shallow-water tidal arrays, WECs, and wind farms, where vertical spacing is either impossible or very costly. At the depths at which OCT arrays are typically deployed, however, there exists a significant vertical range over which the average flow speed varies little (due to a low shear rate near the surface). This creates a significant opportunity to stagger both the horizontal and vertical spacing of the turbines, with the design objective of maximizing power output within a constraint volume. Furthermore, while the aforementioned studies focus on fixed-based turbines, our study considers the *tethered turbines*. The use of tethered systems in deep waters will allow the OCTs



(a) (b)

Figure 19: a) Each OCT's longitudinal location can be adjusted by changing the angle of attack of the wing (i.e., α), which can in turn be altered through elevators. b) Each turbine's lateral location can be adjusted by generating side force on the wing, which can be generated through a rudder. δ_r presents the deflection caused by rudders.

to adjust their vertical positions through the adjustment of tether lengths and their horizontal (longitudinal and lateral) positions via control surfaces.

In this chapter, we propose a two-part layout optimization strategy that iteratively adjusts the 3D locations of each of the turbines within an OCT array in order to maximize output power, subject to volumetric and spacing constraints. The two unique components of our approach are:

1. A low-order parameterization of the OCT locations where a large number of turbines' locations are represented through a small number of *basis parameters*. These basis parameters are sufficient to describe a wide variety of array layouts that can be realized through adjustment of tether lengths and control surfaces. Furthermore, as we show later in this chapter, optimizing these basis parameters leads to array layouts that produce very close to the theoretically optimal power output for the array. Finally, the number of basis parameters does not grow as the array size grows, thereby leading to a highly scalable optimization.

2. Use of Bayesian Optimization to efficiently converge to an optimal set of basis parameters in a very small number of iterations.

3.4.2 Frameworks for Studying Reconfigurable OCT Arrays

We consider an array of N OCTs that each possess (i) the ability to adjust tether payout, (ii) the ability to adjust longitudinal position through the use of elevators and/or a variable bridle geometry, and (iii) the ability to adjust lateral position through the use of rudders. A conceptual drawing of the type of OCT under consideration is shown in Fig. 19a and Fig. 19b. Each OCT is assumed to have two rotors, each located at the aft of two hulls that are connected via a wing. Adjustment of tether payout can be accomplished through the use of a winch system, as is done in the airborne wind energy community (see [81]). Adjustment of longitudinal position can be accomplished through the use of elevators, which alter the angle of attack of the lifting body that houses the on-board turbines, which in turn alters the lift-to-drag ratio of the system. Finally, adjustment of the lateral position can be accomplished through the use of rudders. The deflection of the rudders through an angle δ_r alters the side slip angle of the flow incident to the rudder, which drives the OCT to a new equilibrium position. Consequently, with sufficient control authority, the OCT can navigate itself anywhere within a finite constraint volume. Fig. 20 presents two frameworks for the iterative 3D layout optimization process considered in this chapter. The Framework A assumes instantaneous OCT dynamics, while the more realistic Framework B incorporates a fused flight dynamics and wake interaction model for capturing the dynamic behavior of each OCT. In both frameworks,

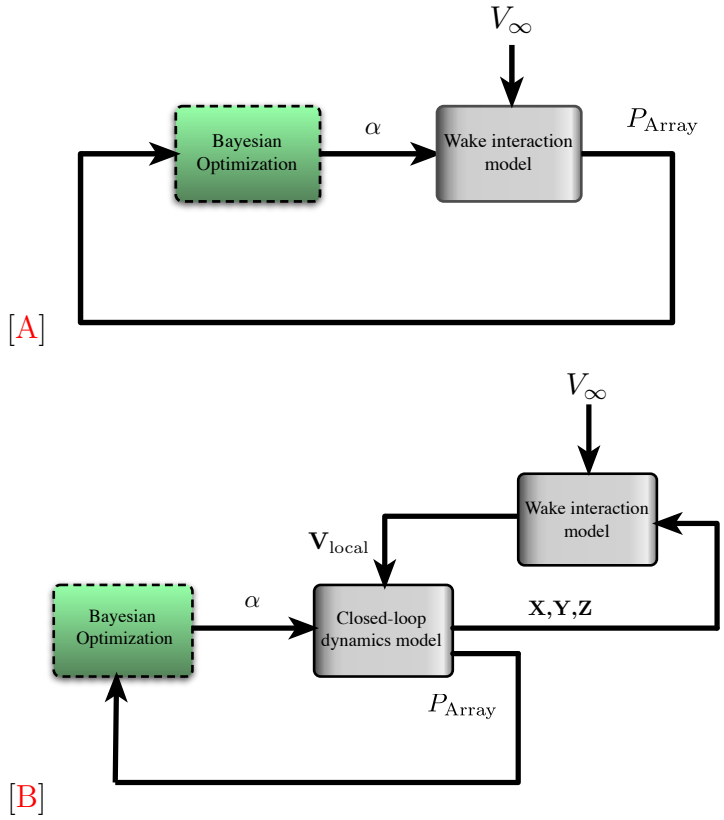


Figure 20: Illustration of the iterative layout optimization processes. While the Framework A assumes instantaneous OCT dynamics, the Framework B incorporates a fused flight dynamics and wake interaction model for capturing the dynamic behavior of each OCT. The mathematical engine for performing the optimization is Bayesian Optimization for both frameworks.

Bayesian Optimization is used to compute a new set of *basis parameters* at discrete time intervals. These basis parameters act as a low order representation of the OCT layout and are related directly to the x, y, and z positions of each turbine in the array. Once the new layout has been determined, the individual OCTs navigate themselves to the prescribed locations, and the corresponding array power output is measured.

3.4.3 Low Order Wake Interaction Model

To evaluate the efficacy of the proposed array layout optimization framework through simulation, it is essential to have a numerical model that predicts array

power output for a given layout. In this chapter, we present an extension of a low-order analytical array power production model from [28] to address very generic 3D geometries that consist of an arbitrary number of rows of turbines. The total power generated by the underwater turbine array is determined through the following four step procedure, detailed in [28], where the x (longitudinal) axis corresponds to the flow direction and the y (lateral) axis corresponds to the cross-flow direction:

- **Step 1**: An influence matrix, M, identifies if a turbine lies in the wake of another turbine. The elements of this matrix are determined as follows:

$$M_{ij} = \begin{cases} +1 & \text{if turbine i influences turbine j} \\ -1 & \text{if turbine j influences turbine i} \\ 0 & \text{if there is no mutual influence} \end{cases} \tag{59}$$

Turbine j is affected by the wake of turbine i if and only if:

$$\Delta x_{ij} < 0,$$
$$d_{ij} - \frac{D_j}{2} < \frac{D_{\text{wake},ij}}{2},$$

where:

$$\Delta x_{ij} = x_i - x_j,$$
$$\Delta y_{ij} = y_i - y_j,$$
$$\Delta z_{ij} = z_i - z_j,$$

and $d_{ij} = \sqrt{\Delta y_{ij}^2 + \Delta z_{ij}^2}$ is the separation distance between the center of turbine j and the wake from i. D_j is the rotor diameter of turbine j. $D_{\text{wake},ij}$ represents the diameter of the wake due to turbine i, reaching turbine j. The diameter of the wake from i, as it reaches j, is given by [43]:

$$D_{\text{wake},ij} = D_i + 2\beta|\Delta x_{ij}| \tag{60}$$

where β is the wake expansion constant. β is set to 0.075 based on results reported in [52, 45]. The wake expansion coefficient depends on several factors, including the upstream turbulence intensity, and turbulence and vorticity introduced by the turbines, but earlier work from a two-turbine study ([52]) suggests that the generated power appears to depend weakly on β.

The wake expansion following turbine i is accompanied by a corresponding reduction in the velocity [52] reaching turbine j:

$$U_{kj} = \Big(1 - \frac{2a}{(1 + 2\beta\Delta_{kj})^2}\Big)U_k \tag{61}$$

where a is the induction factor. Other effects such as wake dissipation due to turbulence, are not included in this model.

- **Step 2**: The turbines are ranked in increasing order of their longitudinal (x) positions. Thus, lower ranks correspond to turbines that are located closer to the "clean" flow.

- **Step 3**: The power outputs of individual turbines are calculated in ascend-

ing order of rank so that the wake effect on downstream devices is properly accounted for. Turbine j might be fully or partially in the wakes of other turbines. The wake of each preceding turbine (e.g. turbine k), for which $M_{kj} = 1$, is mapped onto turbine j as follows:

If the rotor of turbine j lies fully in the wake of turbine k, then:

$$A_{kj} = A_j$$
$$A_j = \frac{\pi D_j^2}{4}$$

Otherwise, if the rotor of turbine j lies partially in the wake of turbine k, then:

$$A_{kj} = r_k^2\, cos^{-1}(\frac{d_{kj}^2 + r_k^2 - r_j^2}{2d_{kj}r_k}) + r_j^2\, cos^{-1}(\frac{d_{kj}^2 + r_k^2 - r_j^2}{2d_{kj}r_k}) - \frac{1}{2}\sqrt{(a)(b)(c)(d)}, \quad (62)$$

where:

$$a = -d_{kj} + r_k + r_j,$$
$$b = d_{kj} - r_k + r_j,$$
$$c = d_{kj} + r_k - r_j,$$
$$d = d_{kj} + r_k + r_j.$$

Thus, Eq. (62) represents the calculation of the area of the region A_{kj}, as the area of the lenticular region between the overlapping circles of radii r_j and r_k.

The effective area of influence of the wake (from turbine k) on turbine j is represented by A_{kj}. r_k and r_j represent the radii of turbines k and j, respectively. Finally, d_{kj} represents the distance between the centers of turbines j and k. For turbine pairs that satisfy $M_{kj} = 1$, the resultant wake velocity is given by:

$$U_j = U_0 - \sqrt{\sum_k \frac{A_{kj}}{A_j}\Big(U_0 - U_{kj}\Big)^2}. \tag{63}$$

The power generated by turbine j, denoted by P_j, is then determined by:

$$P_j = C_p\Big(\frac{1}{2}\rho\pi\frac{D_j^2}{4}U_j^3\Big), \tag{64}$$

where ρ is the density of the fluid (water) and C_p represents the power coefficient.

- **Step 4**: Finally, the power generated by the array, P_{Array}, is calculated as the sum of the power outputs from the individual turbines:

$$P_{\text{Array}} = \sum_{j=1}^{N} P_j. \tag{65}$$

3.4.4 Fused Flight Dynamics and Closed-Loop Control System Model

The fused model of the tethered system array characterizes N_{sys} systems, where N_{sys} can be tailored to different array sizes. Each tethered system consists of a lower level flight controller and dynamic model of the lifting body and tethers. The general block diagram of the dynamic model is provided in Figure 21. Each of the components of this block diagram is detailed in this section. In Fig.21, the flow velocity into the tethered array is denoted as $\mathbf{V}_{flow} = \begin{bmatrix}\mathbf{v}_{flow,1} & \dots & \mathbf{v}_{flow,N_{sys}}\end{bmatrix}^T$. The control input

vector for the i^{th} dynamic model is denoted as $\mathbf{u}_{cmd,i} = \begin{bmatrix} u_{center,i} & u_{stbd,i} & u_{port,i} & u_{rudder,i} \end{bmatrix}^T$. The first three elements of $\mathbf{u}_{cmd,i}$ are tether release speeds, while $u_{rudder,i}$ is an angular deflection of the rudder.

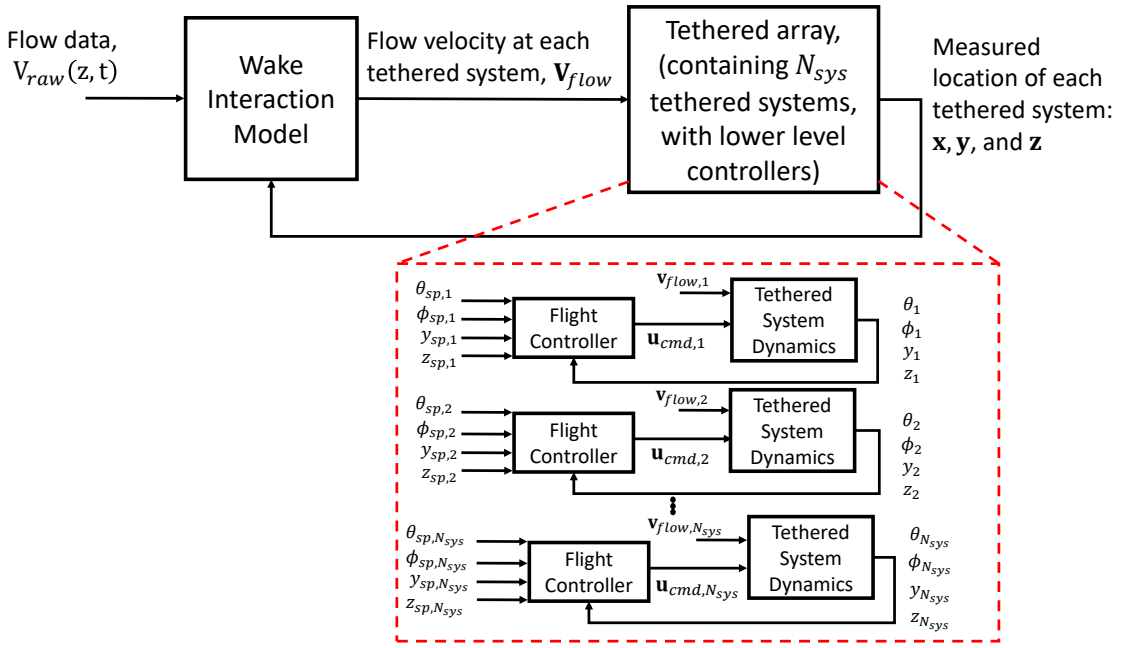


Figure 21: Block diagram of the fused flight dynamics, flight control, and wake interaction model [33]

3.4.4.1 Dynamic Model of an Individual Tethered System

The model for each individual tethered system has been developed based on an airborne wind energy system model first discussed in [81]. The model presented in this section is for a single system. Therefore, each variable in this section should be indexed with an i to correspond to i^{th} dynamic model; however, for conciseness of notation, this index is omitted in this section. This modeling framework assumes that each system contains three tethers, with lengths given by l_{center}, l_{port}, and l_{stbd}, where the average tether length is given by L_t.

Derived using an Euler-Lagrange framework, this dynamic model is described by six degrees of freedom (three uncontrolled and three controlled), which are shown in Fig. 22 and given by: Θ, Φ, Ψ, L_t, θ' and ϕ'. Θ represents the angle between the longitudinal wind axis and the tether's projection onto the ground, referred to as the azimuth angle. Φ is the zenith angle, which represents the angle between the vertical axis and tether. Ψ describes twist angle about the tether axis. The average tether length is given by L_t, whereas θ' and ϕ' represent the *induced* pitch and roll angles that are introduced by varying the lengths of the individual elements. By introducing these induced angles, we approximate a multi-tethered system as a single-tethered system with a controllable bridle joint. This leads to comparable dynamics that can be expressed entirely through differential equations, without the need to resort to differential algebraic equations (where algebraic constraint equations result from the presence of multiple tethers). The induced angles are related to the lengths of the bridle elements, denoted by l_{center}, l_{port}, and l_{stbd}, as follows:

$$\theta' = \tan^{-1}\left(\frac{l_{center} - 0.5(l_{port} + l_{stbd})}{l_{center}}\right), \tag{66}$$

$$\phi' = \tan^{-1}\left(\frac{l_{stbd} - l_{port}}{l_{center}}\right). \tag{67}$$

Each tethered system is controlled using a rudder (whose controlled angular deflection is given by u_{rudder}) and the release rates of the three tethers, which are specified by u_{center}, u_{stbd}, and u_{port}). The tether lengths are directly related to the release rates

through pure integrators:

$$\frac{d}{dt}(l_{center}) = u_{center}, \tag{68}$$

$$\frac{d}{dt}(l_{stbd}) = u_{stbd}, \tag{69}$$

$$\frac{d}{dt}(l_{port}) = u_{port}, \tag{70}$$

and u_{rudder} directly impacts the yaw moment acting on each tethered system.

Ultimately, the system dynamics are given by:

$$D(Q)\ddot{Q} + C(Q,\dot{Q})\dot{Q} + g(Q) = \tau(Q,\dot{Q},\mathbf{v}_{flow},\psi_{flow},u_{rudder}), \tag{71}$$

$$X = f(Q,\dot{Q}), \tag{72}$$

$$\Omega = g(Q,\dot{Q}), \tag{73}$$

where:

$$Q = [\Theta \;\; \Phi \;\; \Psi \;\; L_t \;\; \theta' \;\; \phi']^T, \tag{74}$$

$$X = [x \;\; y \;\; z \;\; u \;\; v \;\; w]^T, \tag{75}$$

$$\Omega = [\phi \;\; \theta \;\; \psi \;\; p \;\; q \;\; r]^T. \tag{76}$$

Here, τ is a vector of generalized forces, $\mathbf{v}_{flow}$ is the *local* flow velocity at each tethered system (in contrast to the free-stream flow velocity, the local flow velocity is an output of the wake interaction model, which accounts for the velocity deficit within the wake of each system), ψ_{flow} is the flow direction, and u_{rudder} is the rudder input. The orientation of the apparent flow velocity vector with respect to body-fixed

coordinate system is given by the angle of attack (α) and side-slip angle (β). Aerodynamic/hydrodynamic forces and moments are functions of both α and β.

The tethered system model that was used for the simulations in this chapter consists of a main body, vertical stabilizer, and two turbines. As such, the total aerodynamic/hydrodynamic coefficients have been *partitioned* between the main body and stabilizers. Partitioning the coefficients allows for rapid reconfigurability of the aerodynamic/hydrodynamic aspects of the system. Total coefficients are given by:

$$C^{total}_{D,L,S}(\alpha,\beta) = C^{body}_{D,L,S}(\alpha,\beta) + C^{V}_{D,L,S}(\alpha,\beta)\frac{A_V}{A_{body}}, \tag{77}$$

$$\begin{aligned} C^{total}_{Mx,My,Mz}(\alpha,\beta) &= C^{body}_{Mx,My,Mz}(\alpha,\beta) \\ &\quad + C^{V}_{Mx,My,Mz}(\alpha,\beta)\frac{A_V l_V}{A_{body} l_{body}}. \end{aligned} \tag{78}$$

Here, C_D, C_L, C_S, C_{Mx}, C_{My}, and C_{Mz} represent the drag, lift, side force, roll moment, pitching moment and yaw moment coefficients of a single tethered system, respectively. Superscripts and subscripts *body* and V denote the main body and vertical stabilizer; A represents a surface area, whereas l represents a chord length.

3.4.4.2 Flight Controller

The closed loop system (combined flight dynamics model and controller) for each individual tethered system is shown in Fig. 23. Similar to the dynamic modeling section, the i (denoting the particular tethered system under consideration) has been removed in this section for notational simplicity. For each individual tethered system, the goal is to to track a prescribed pitch angle setpoint (θ_{sp}), roll angle setpoint (ϕ_{sp}),

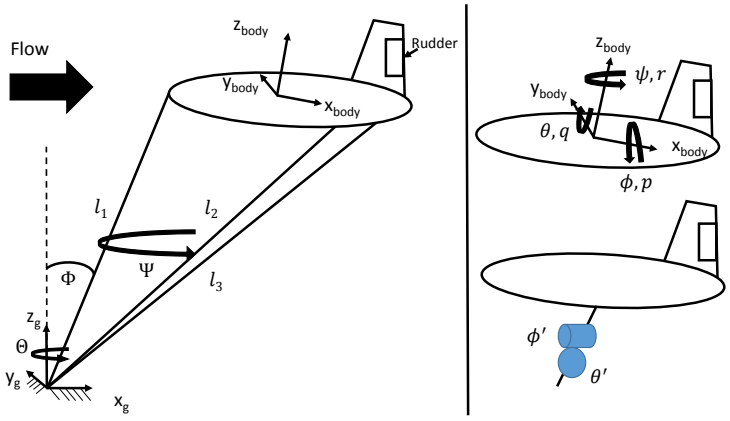


Figure 22: Axis system for tethered system dynamic model. The full system model is shown on the left in global coordinates. In the right frame, the body fixed coordinates and Euler angles from ground to body fixed coordinates can be seen. Also, the controlled degrees of freedom at the bridle joint are displayed [33].

altitude setpoint (z_{sp}), and lateral position setpoint (y_{sp}). To achieve tracking of each of the first three setpoints (θ_{sp}, ϕ_{sp}, and z_{sp}), three lead filters are used, as in [81]:

$$\bar{u}_z(s) = \frac{k_{d,z}s + k_{p,z}}{\tau_z s + 1}(z_{sp}(s) - z(s)), \tag{79}$$

$$\bar{u}_\theta(s) = \frac{k_{d,\theta}s + k_{p,\theta}}{\tau_\theta s + 1}(\theta_{sp}(s) - \theta(s)), \tag{80}$$

$$\bar{u}_\phi(s) = \frac{k_{d,\phi}s + k_{p,\phi}}{\tau_\phi s + 1}(\phi_{sp}(s) - \phi(s)). \tag{81}$$

The commanded velocities of each tether, denoted by u_{center}, u_{stbd}, and u_{port}, are related to $\bar{u}_z$, $\bar{u}_\theta$, and $\bar{u}_\phi$ through simple linear combinations:

$$u_{center} = \bar{u}_z - \bar{u}_\theta, \tag{82}$$

$$u_{stbd} = \bar{u}_z + \bar{u}_\theta + \bar{u}_\phi, \tag{83}$$

$$u_{port} = \bar{u}_z + \bar{u}_\theta - \bar{u}_\phi. \tag{84}$$

The above linear combinations correspond to the use of synchronous tether motion to control altitude (depth), asynchronous forward/aft control to regulate pitch, and asynchronous port/starboard control to regulate roll. Because of the pure integrators that relate $\bar{u}_{center}$, $\bar{u}_{stbd}$, and $\bar{u}_{port}$ to the tether lengths, integral control is not needed for tracking of θ_{sp}, ϕ_{sp}, and z_{sp}.

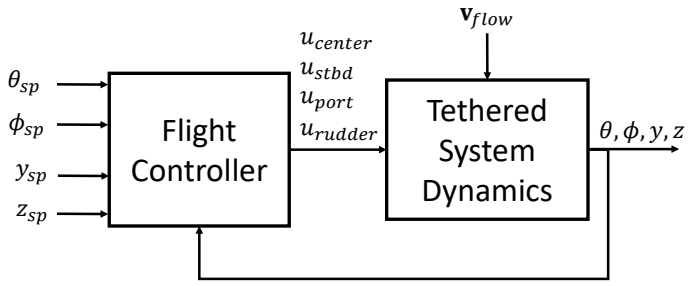


Figure 23: Closed loop block diagram for an individual tethered system [33]

Unlike the control system detailed in [81], we assume that each tethered system is equipped with a rudder that allows for lateral movement. Having the ability to use the rudder allows the system to adjust its lateral position, which can be used to move out of the wake of an upstream system. The rudder is controlled by a filtered PID controller, parameterized as:

$$u_{rudder}(s) = \frac{k_{d,y}s^2 + k_{p,y}s + k_{i,y}}{(\tau_y s + 1)s}(y_{sp}(s) - y(s)). \tag{85}$$

3.4.5 Array Layout Optimization Methodology

Our optimization process will focus on maximizing power output, subject to volumetric and spacing constraints. The design variables are the x, y, and z positions of each turbine of N OCT array. The optimization problem takes the form:

$$\text{Maximize} \quad P_{\text{Array}}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \tag{86}$$

subject to:

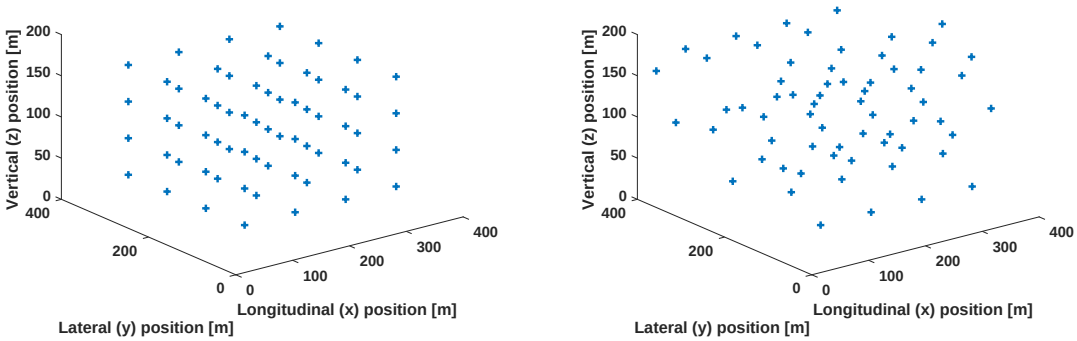
$$(x_i, y_i, z_i) \in V, \quad i = 1, \ldots, N,$$

$$d_{i,j} \geq d_{\min}, \quad i \neq j,$$

where P_{Array} represents generated power from the array, and $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ are vectors that contain the x, y, z positions of each of the N turbines in the array, i.e:

$$\mathbf{X} = [x_1, \ldots, x_N], \quad \mathbf{Y} = [y_1, \ldots, y_N], \quad \mathbf{Z} = [z_1, \ldots, z_N], \tag{87}$$

and $d_{i,j}$ represents the Euclidean distance between turbine i and j. Directly optimizing these $3N$ design variables will lead to a computationally intense and poorly scaled optimization problem (where the number of design variables grows linearly with the array size and the computational complexity of the optimization grows super-linearly in most cases). To alleviate this computational and scalability issue, we introduce a



(a) $\rho_{\text{long}} = 4$, $\rho_{\text{lat}} = 4$, $\rho_{\text{vert}} = 4$, $f_{\text{long}} = 0$, $f_{\text{lat}} = 0$, $f_{\text{vert}} = 0$

(b) $\rho_{\text{long}} = 4$, $\rho_{\text{lat}} = 4$, $\rho_{\text{vert}} = 4$, $f_{\text{long}} = 0.25$, $f_{\text{lat}} = 0.5$, $f_{\text{vert}} = 0.25$

Figure 24: Comparison of a simple 3D rectangular array (a- corresponding to $\rho_{\text{long}} = 4$, $\rho_{\text{lat}} = 4$, $\rho_{\text{vert}} = 4$, $f_{\text{long}} = 0$, $f_{\text{lat}} = 0$, $f_{\text{vert}} = 0$) with a staggered array that makes use of six basis parameters (b- corresponding to $\rho_{\text{long}} = 4$, $\rho_{\text{lat}} = 4$, $\rho_{\text{vert}} = 4$, $f_{\text{long}} = 0.25$, $f_{\text{lat}} = 0.5$, $f_{\text{vert}} = 0.25$). It is evident from this figure that a small number of basis patterns can yield highly general layouts. Blue + symbols represent the centers of the OCTs in both figures.

novel technique in this section wherein the array is parameterized as a function of a small number of *basis parameters*, leading to a more efficient and scalable optimization problem. We then use Bayesian Optimization to iteratively optimize this small set of basis parameters.

3.4.5.1 Array Layout Parameterizations

To form a computationally efficient and highly-scalable optimization framework, we describe each array configuration via a small set of *basis parameters*. Specifically, these basis parameters represent a small number of scalar array geometry variables that describe properties of the 3D array that are anticipated to be critical to power production.

In our case, we fix the number of rows, columns, and layers that comprise the array and use the basis parameters to characterize the level of *staggering* between these

rows, columns, and layers, which can be physically accomplished through the use of tether length and control surface adjustment. Specifically, we model this staggering through three basis parameters, namely:

- f_{long}: Longitudinal stagger fraction (fraction of nominal row separation by which successive columns are staggered)
- f_{lat}: Lateral stagger fraction (fraction of nominal column separation by which successive rows are staggered)
- f_{vert}: Vertical stagger fraction (fraction of nominal row separation by which successive layers are staggered)

These basis parameters will be represented in this work through a compact vector, α, where:

$$\alpha = \begin{bmatrix} f_{\text{long}} & f_{\text{lat}} & f_{\text{vert}} \end{bmatrix}^T. \tag{88}$$

Taking L, W, and D as the length, width, and depth, respectively, of the constraint domain, and taking ρ_{long}, ρ_{lat}, and ρ_{vert} as the number of rows, columns, and layers in the array, the basis parameters of α are related to the actual turbine positions through the following (rather intricate) formulas:

$$
\begin{aligned}
x_i &= \frac{L(1-\frac{1}{\rho_{\text{long}}-1})}{\rho_{\text{long}}-1}\text{floor}(\frac{i-1}{\rho_{\text{lat}}}) \\
&\quad + \frac{Lf_{\text{long}}}{\rho_{\text{long}}-1}\text{mod}(i-1,\rho_{\text{lat}}) + \frac{L}{2(\rho_{\text{long}}-1)}, \qquad (89) \\
y_i &= \frac{W(1-\frac{1}{\rho_{\text{lat}}-1})}{\rho_{\text{lat}}-1}\text{mod}(i-1,\rho_{\text{lat}}) \\
&\quad + \frac{Wf_{\text{lat}}(i-1)}{\rho_{\text{lat}}(\rho_{\text{lat}}-1)} + \frac{W}{2(\rho_{\text{lat}}-1)}, \qquad (90) \\
z_i &= \text{floor}(\frac{i-1}{\rho_{\text{long}}\rho_{\text{lat}}})\frac{D-\frac{D}{\rho_{\text{vert}}}}{\rho_{\text{vert}}-1} + \frac{D}{2(\rho_{\text{vert}}-1)} \\
&\quad + \text{floor}(\frac{\text{mod}(i-1,\rho_{\text{long}}\rho_{\text{lat}})}{\rho_{\text{lat}}})\frac{f_{\text{vert}}}{2(\frac{W}{\rho_{\text{lat}}-1})} \\
&\quad + \text{mod}(i-1,\rho_{\text{lat}})\frac{f_{\text{vert}}}{2(\frac{L}{\rho_{\text{long}}-1})}. \qquad (91)
\end{aligned}
$$

To gain an appreciation for the effect of the basis parameters, Fig. 24 compares two scenarios: The figure on the left shows a layout with $\rho_{\text{long}} = 4$, $\rho_{\text{lat}} = 4$, $\rho_{\text{vert}} = 4$, $f_{\text{long}} = 0$, $f_{\text{lat}} = 0$, and $f_{\text{vert}} = 0$. By taking staggering parameters to be equal to zero, we end up with a simple rectangular array. In the right sub-figure, we take the same values for ρ_{long}, ρ_{lat}, and ρ_{vert} but take $f_{\text{long}} = 0.25$, $f_{\text{lat}} = 0.5$, and $f_{\text{vert}} = 0.25$, respectively. Through just these three basis parameters, we arrive at an array geometry that departs significantly from a simple rectangular array. Furthermore, different values of f_{long}, f_{lat}, and f_{vert} can be used to generate widely differing array geometries. Most importantly, as we will show in Section 3.4.6, this simple parameterization is sufficient to describe array geometries that produce power outputs that are very close to the theoretical limit for the given number of turbines.

We can also quantify the feasible domain over which each turbine in the array can adjust its location through use of tether length adjustment and control surfaces. To

achieve this goal, we have conducted a simple simulation where the variation of x, y, and z position were computed through variation of the chosen basis parameters (f_{long}, f_{lat}, and f_{vert}).

Fig. 25 illustrates variation in turbine positions for three different turbines in an OCT array, as each of the aforementioned basis parameters (f_{long}, f_{lat}, and f_{vert}) is varied while the other three remain fixed at $\rho_{\text{long}} = 5, \rho_{\text{lat}} = 5, \rho_{\text{vert}} = 4$. One can conclude from this figure that it is possible to define a continuous constraint volume within which each turbine must lie, for any combination of basis parameters.

The maximum constraint volume within which each OCT is restricted to lie depends, intuitively, on the total constraint volume and number of turbines (LWD and $\rho_{\text{long}}\rho_{\text{lat}}\rho_{\text{vert}}$, respectively). Based on a least squares regression over 125 configurations, the maximum constraint volume, V_{max}, can be approximated as:

$$V_{\text{max}} = \bar{K} LWD\Big(\rho_{\text{long}}^{-1}\rho_{\text{lat}}^{-1}\rho_{\text{vert}}^{-1}\Big) \tag{92}$$

where $\bar{K} \approx 1.73$.

Identification of this value of $\bar{K}$ indicates that:

- Each OCT can indeed be guaranteed to lie within a continuous constraint volume whose size varies predictably with the packing density of OCTs within a constraint volume.

- Each OCT's constraint volume partially overlaps its neighbors' constraint volumes, which can be identified by the fact that $\bar{K} > 1$. Our optimization does not allow for turbines to occupy the same space, however, through a spacing

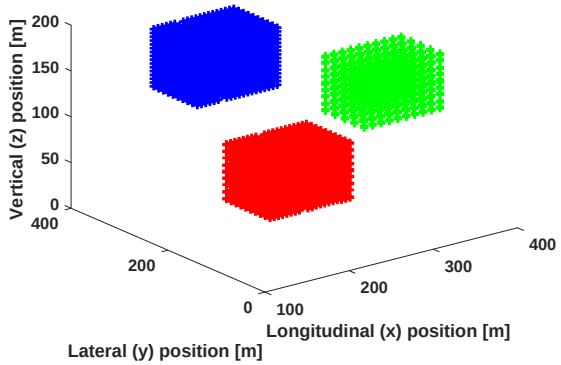


Figure 25: Domain of achievable x, y, and z positions for three different turbines that is made possible through variation of basis parameters, f_{long}, f_{lat}, and f_{vert}, holding the other basis parameters fixed.

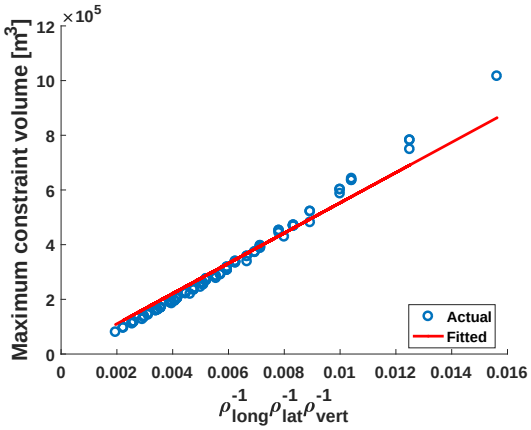


Figure 26: Maximum constraint volume based on different number of turbines

constraint.

Fig. 26 represents how the actual and estimated (i.e., fitted) maximum constraint volume (based on the least squares regression) vary for different numbers of turbines.

3.4.6 Results - Framework A

To evaluate the proposed algorithm, a simple test case was set up, consisting of a domain of $400m \times 400m \times 200m$, subjected to a uniform inlet flow of $1\frac{m}{s}$ [19, 20]. A summary of parameters used in this test case is provided in Table 5. To assess the

Table 5: The parameter values used in the proposed algorithm

Description	Symbol	Value	Units
Water density	ρ	1000	$\frac{kg}{m^3}$
Inlet flow velocity	U_0	1	$\frac{m}{s}$
Turbine diameter	D_t	5	m
Power coefficient	C_p	0.45	-
Wake expansion constant	β	0.075	-
Domain length	L	400	m
Domain width	W	400	m
Domain depth	D	200	m

effectiveness of our algorithm, we consider two different scenarios:

- **Scenario 1: Baseline Rectangular Array**

  This scenario represents the case where $f_{\text{lat}} = 0$, $f_{\text{long}} = 0$, and $f_{\text{vert}} = 0$. This baseline scenario conforms to a rectangular layout.

- **Scenario 2: Staggered Array Using Bayesian Optimization**

  Here, each of the staggering basis parameters, namely f_{lat}, f_{long}, and f_{vert}, are optimized iteratively using Bayesian Optimization.

We also compare the performance of the optimized array (Scenario 2) with the theoretical maximum array output power for the given number of turbines (N). This maximum possible output power is given by:

$$P_{\text{Array}} = \frac{1}{2}\rho C_p U_0^3 \pi (\frac{D_t}{2})^2 N \tag{93}$$

where N is the total number of turbines. This reflects the total array power output under the condition that each OCT is exposed to the free stream flow velocity, U_0.

Fig. 27 illustrates the percentage of maximum available power produced at each

iteration, where the total allowable number of turbines varies from 100 to 400. It is noteworthy here that there are several iterations over which the best power remains constant. These are iterations at which the power output is not improving (or is even getting worse); however, Bayesian Optimization is *learning* more about the nature of the array power vs. basis parameters at these iterations, leading to an ultimate improvement at later iterations. Table 7 presents the detailed results for different scenarios where the total number of turbines is restricted to a number ranging from 100 to 800. The optimized array is able to achieve nearly the maximum available power. To gain an appreciation for the final layout, Fig. 28 and Fig. 29 present the final layout for the first and second scenario, where the number of turbines is restricted to 75 and 100, respectively. In these figures, the center of each "+" represents the center of each turbine in the array. The color of each turbine indicates the amount of power that is harnessed by that turbine (see Table 6).

3.4.6.1 Parametric Trade Study

In this section, we explore how (and if) the results of the proposed layout optimization approach vary as we sweep through the values of key parameters. Specifically, we conduct a parameter sweep across the OCT axial induction factor (which is assumed to be the same for each OCT) and allowable separation distance between the

Table 6: Illustration of colored circles as center of turbines

Color	**Percentage of rated power**
Red	>= 95%
Orange	90% - 95%
Yellow	85% - 90%
Green	<85%

Table 7: Power output for a rectangular array (scenario 1) and optimized array (scenario 2)

Maximum allowable number of turbines	Optimal staggered array layout [kW]	Rectangular array out [kW]	Maximum available power [kW]
100	442	325	442
150	661	488	663
200	877	651	884
400	1750	1310	1770
800	3290	1350	3470

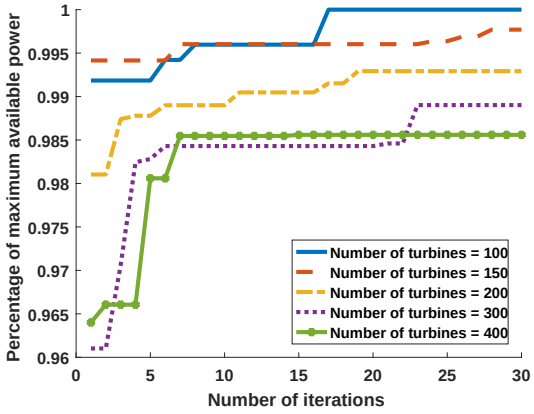
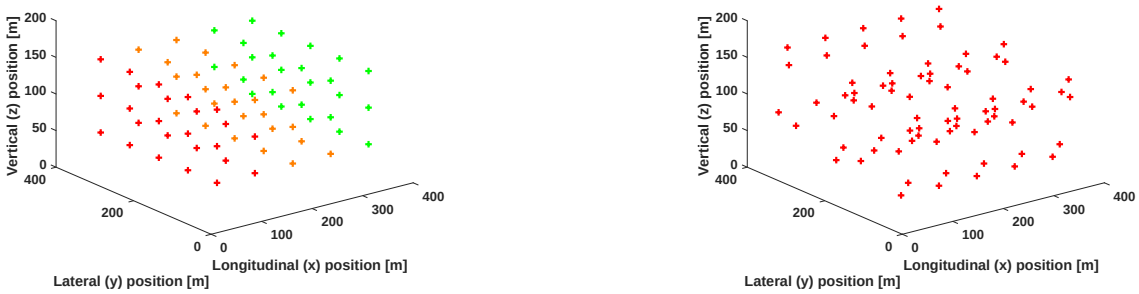


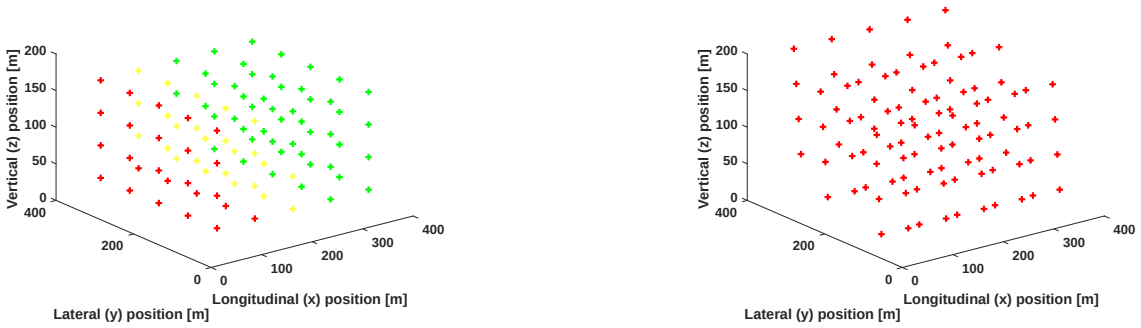
Figure 27: Evolution of the normalized best observed power output power at each iteration of Bayesian Optimization (maximum allowable number of turbines varies from 100 to 400)



(a) (b)

Figure 28: Rectangular vs. staggered Bayesian Optimization layout (maximum allowable number of turbine is 75).

turbines.



(a) (b)

Figure 29: Rectangular vs. staggered Bayesian Optimization layout (maximum allowable number of turbine is 100).

3.4.6.2 Sensitivity to Induction Factor

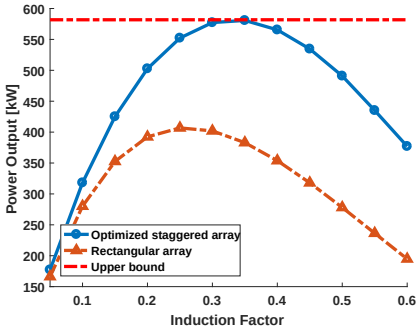
In the presence of a wake, the flow velocity reaching the rotor plane is smaller than the inlet stream velocity. The ratio of this reduction to that of the flow velocity far away from the turbine, denoted as a, is given by:

$$a = \frac{U_0 - U_1}{U_0} \tag{94}$$

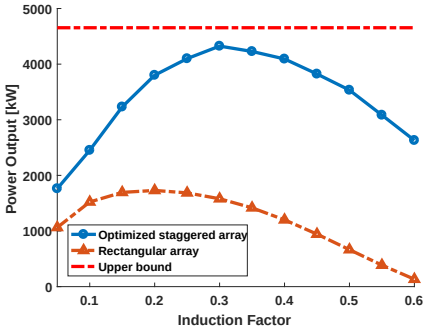
where U_0 is the inlet flow velocity and U_1 is the flow velocity at the rotor. The relationship between the power coefficient, C_p, and the induction factor, a, is given by:

$$C_p = 4a(1-a)^2 \tag{95}$$

The maximum value of C_p (called Betz limit) is equal to $\frac{16}{27}$. Therefore, even with the best rotor design, it is not possible to extract more than about 60% of the kinetic energy from the flow. Furthermore, one can show that this maximum occurs at $a = \frac{1}{3}$.



(a) Number of turbines is fixed to 100



(b) Number of turbines is fixed to 800

Figure 30: Evolution of the power output over a rage of induction factors for a fixed number of turbines

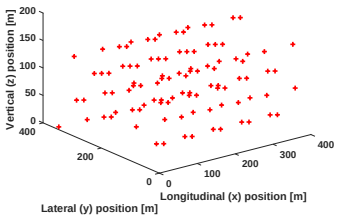


Figure 31: Induction factor = 0.3, Number of turbines fixed to 100

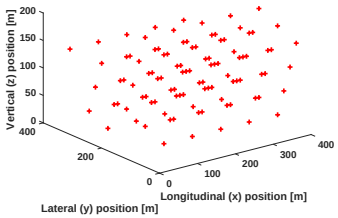


Figure 32: Induction factor = 0.4, Number of turbines fixed to 100

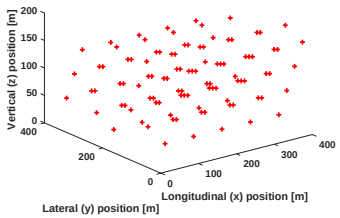


Figure 33: Induction factor = 0.5, Number of turbines fixed to 100

Fig. 30 examines how the array power output varies for a range of induction factors, where the number of turbines is fixed. One can infer from this figure that the array power output has a peak around the optimum induction factor, as expected. Furthermore, one can conclude from this figure that the staggered optimized array results in significantly more power output than a rectangular array farm, regardless of the induction factor. Furthermore, it should be noted that in these figures, the layouts have been individually optimized for each induction factor considered. For instance, Figs 31-33 represent the optimal layouts corresponding to induction factor of 0.3, 0.4, and 0.5, respectively.

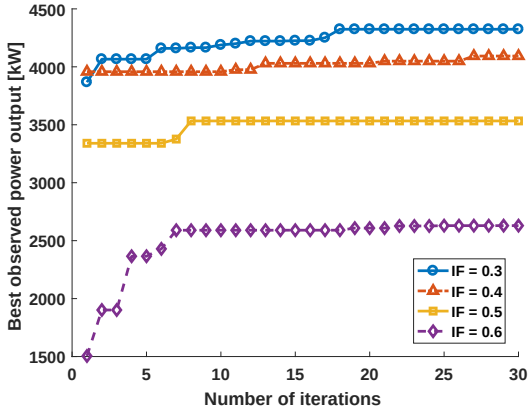


Figure 34: Evolution of the normalized best observed power output at each iteration of Bayesian Optimization for a number of induction factors

Fig. 34 illustrates the evolution of the normalized best observed power output at each iteration of Bayesian Optimization for a number of feasible induction factors.

3.4.6.3 Sensitivity to Spacing Constraint

Practical limitations will restrict the safe proximity of OCTs with respect to one another. The allowable separation of turbines is taken into account in optimization formulation as a "soft" constraint. In general, soft constraints act as approximations of the hard constraints and are penalized in the objective function. In this case, the modified optimization problem takes the following form:

$$\text{Maximize} \quad P_{\text{Array}} \quad (\mathbf{X}, \mathbf{Y}, \mathbf{Z}) - k \sum_{i=1}^{N} \Big(\max\big(D_{\text{thresh}} - d_{\min}(i), 0\big) \Big)^2 \tag{96}$$

where D_{thresh} represents the maximum allowable separation of turbines. $d_{\min}(i)$ denotes the minimum Euclidean distance between turbine i and all other turbines, and k represents a scalar weighting parameter. Fig. 35 shows the evolution of the normalized best observed power output at each iteration of Bayesian Optimization, where D_{thresh} varies from 10m to 40m. One can infer from this figure that the most power

output occurs under the least severe maximum allowable separation constraint, as expected.

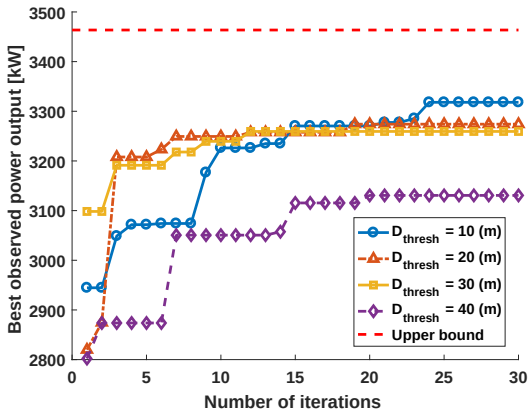
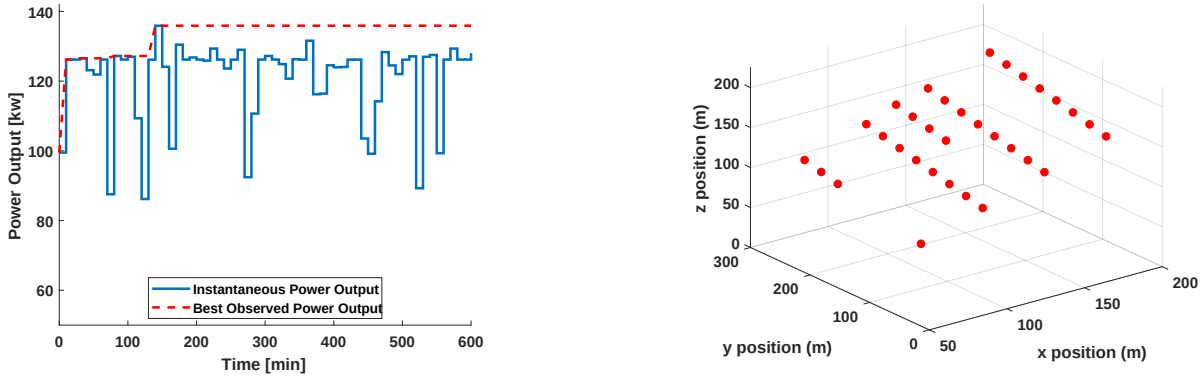


Figure 35: Evolution of the normalized best observed power output power at each iteration of Bayesian Optimization for various D_{thresh} (the number of turbines is fixed)

3.4.7 Results - Framework B

In summary, the framework B presented in Section 3.4.2 integrates the lower-level OCT layout adjustment controllers with the higher-level context-dependent Bayesian Optimization algorithm to generate realistic simulations of a reconfigurable OCT array. This model (i) is driven with an inlet velocity profile from actual ADCP current measurements, (ii) uses the low-order wake interaction model to predict local flow velocities at each OCT as a function of each OCT's location, and (iii) utilizes context-dependent Bayesian Optimization to iteratively optimize array locations.

We evaluate the performance of context-dependent Bayesian Optimization using real Gulf Stream data from the UNC-Coastal Studies Institute to drive the dynamic model. An ADCP provides flow speed measurements in 4 m intervals in space, down to a depth of $230m$, and 10 minute intervals in time. We assess the performance of our layout optimization algorithm over the 6 hour simulation, where each iteration of



(a) (b)

Figure 36: (a) Instantaneous power output vs. best observed power output (b) Corresponding optimal layout (maximum allowable number of turbine is 32).

Bayesian Optimization corresponds to a 10 minute simulation. Fig. 36-a illustrates the evolution of instantaneous power output and the best observed power output over time. Furthermore, Fig. 36-b provides a snapshot of the corresponding optimal configuration at the conclusion of the 6 hour simulation. Given the lengthy time constants associated with Gulf Stream flow variations (commonly 3-8 days), performing computationally tractable simulations over longer durations has been identified as a topic of future work.

CHAPTER 4: FUSED PLANT / CONTROLLER OPTIMIZATION USING BAYESIAN OPTIMIZATION

4.1 Introduction

So far, this thesis has examined ways in which Bayesian Optimization, a tool that has been used in many applications for design optimization in past work [26, 53, 16], can also be used for real-time control (for AWE and OCT applications). A number of engineering applications, including those considered in this research, involve both a complex design optimization *and* control system optimization, where the optimal controller depends on the design and vice versa.

This chapter shows how Bayesian Optimization can be employed to solve the problem of coupling between plant and controller optimization. Specifically, we utilize a nested framework wherein a batch of plant designs are chosen at each iteration using Bayesian Optimization, then Bayesian Optimization is used in continuous time to optimize control parameters for each given plant.

4.2 Background and Literature Survey

Combined plant and controller problems consist of those scenarios in which the optimal controller depends on the physical system design (i.e., the plant) and vice versa. Combined plant and controller optimization (often termed *co-design*) has been employed for a wide variety of systems, including automotive suspension systems (see [40], and [9]), elevator systems [39], and the AWE application described earlier (with

initial studies reported in [64], and [32]). Broadly speaking, combined plant and controller strategies fall into four different categories:

- *Sequential*: A sequential strategy completes the plant and controller optimization problems in successive order. For instance, authors in [66] use control proxy functions to augment the objective function of the plant, thereby separating the plant and and controller optimization problem into two sub-problems.

- *Iterative*: The iterative approach fully optimizes the plant design for a given controller, then optimizes the controller design for a fixed plant, then repeats the cycle (see [84], [69]).

- *Nested*: A nested optimization approach contains two loops: an inner loop that completes a full optimization of the controller and an outer loop that completes an iteration of the plant optimization (see [64], and [40], [39], and [41]).

- *Simultaneous*: In a simultaneous optimization strategy, both the plant and controller optimization problems are carried out at the same time [12], [31]. An efficient decomposition-based variant of simultaneous optimization is proposed in [10].

Among these techniques for solving the co-design problem, the nested co-design approach is unique in its ability to leverage critical differences in control parameters (which can be modified *during* experiments) and plant parameters (which, when experimental work is required, can generally only be modified *between* experiments). This sort of approach can therefore be extremely beneficial in complex systems where

experiments will ultimately be required. However, existing literature on nested co-design makes numerous simplifying assumptions (linearity, full state measurement, etc.). The tools used for the plant adjustment are typically local in nature and often unsuitable for complex systems. Furthermore, most existing literature on nested co-design uses continuous-time optimal control design techniques (such as LQR) for the controller design ([42]), without taking advantage of online adaptation capabilities.

Recently, authors in [32] proposed a nested co-design approach in which the control parameter is adjusted during a simulation/experiment and plant parameters are optimized *between* simulations/experiments. The benefit here is that when the design optimization process involves lengthy simulations or experiments, the ability to adjust controller parameters during the simulations/experiments can significantly reduce the time and cost of the optimization process. The framework in [32] used G-optimal design of Experiments (DoE) to select a batch of candidate plant parameters that cover the design space while maximizing a statistical information metric. Furthermore, extremum seeking (ES) control was utilized to adjust the control parameter(s) in real-time over the course of experiments/simulations. The authors recently extended their work in [34] to consider online adaptation mechanisms that are based on a global statistical information metric. While resolving some challenges, these approaches suffer from two main drawbacks:

1. Populating the design space with candidate points in order to merely gain the most information about the design space may lead to the evaluation of candidate designs that have very poor associated performance. This will lead to significant

Table 8: Description of key variables involved in the combined plant and controller optimization

Variable	Description
$\mathbf{p}_p$	Plant parameter(s)
$\mathbf{p}_c$	Control parameter(s)
$\mathbf{p}_c^*(\mathbf{p}_p)$	Optimal control parameter(s) for a candidate plant design
$\mathcal{B}_j$	Elements (candidate plant designs) in a batch at iteration j
$J(\mathbf{p}_c^*(\mathbf{p}_p), \mathbf{p}_p)$	Integral performance value while operating at the optimal control parameter(s) and candidate plant design

effort expanded in characterizing regions of the design space that are unlikely to yield optimal design parameters.

2. ES only achieves local, rather than global optimality.

To address these challenges, we need a framework that (i) effectively explores the global design space in a small number of iterations and (ii) dose not focus on explanation of designs with very poor expected performance. Once again, Bayesian Optimization is a fitting approach for addressing these challenges.

4.3 Fused Plant and Controller Methodology

4.3.1 Problem Formulation

The ultimate goal of this study is to solve the following optimization problem:

$$\underset{\mathbf{p}_p, \mathbf{p}_c}{\text{minimize}} \quad J(\mathbf{p}_p, \mathbf{p}_c) = \int_0^{T_{\text{final}}} g\big(\mathbf{x}(t); \mathbf{p}_p, \mathbf{p}_c\big) dt \tag{97}$$

subject to:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{d}; \mathbf{p}_p, \mathbf{p}_c) \tag{98}$$

$$\mathbf{p}_p \in P, \quad \mathbf{p}_c \in C \tag{99}$$

where (1) describes an integral cost function, (2) represents a general dynamic model that governs the system dynamics, and (3) presents hard constraints on plant and control parameters, which are denoted by $\mathbf{p}_p$ and $\mathbf{p}_c$, respectively. $\mathbf{d}$ is the state disturbance vector. This work will focus on co-design processes that are carried out in controlled environments, where the environmental perturbation (manifested by $\mathbf{d}$) is consistent between simulations/experiments but nonetheless important.

The nested optimization framework consists of two main loops:

- **Outer loop**: Plant parameters ($\mathbf{p}_p$) are adjusted in a direction that is chosen by one iteration of our chosen optimization tool (i.e., Bayesian Optimization).
- **Inner loop**: For the candidate set of plant parameters generated by the outer loop, the inner loop completes a full optimization of the control parameter vector.

Fig. 37 illustrates the proposed nested co-design framework. A description of each variable within the general process is provided in Table 8. The process continues until convergence to optimal parameters is obtained. Algorithm 3 also summarizes the process proposed in this work.

Bayesian Optimization traditionally seeks to maximize the value of some performance index. To keep our exposition aligned with this traditional implementation, we define:

$$R \triangleq -J(\mathbf{p}) \tag{100}$$

as the reward to be maximized.

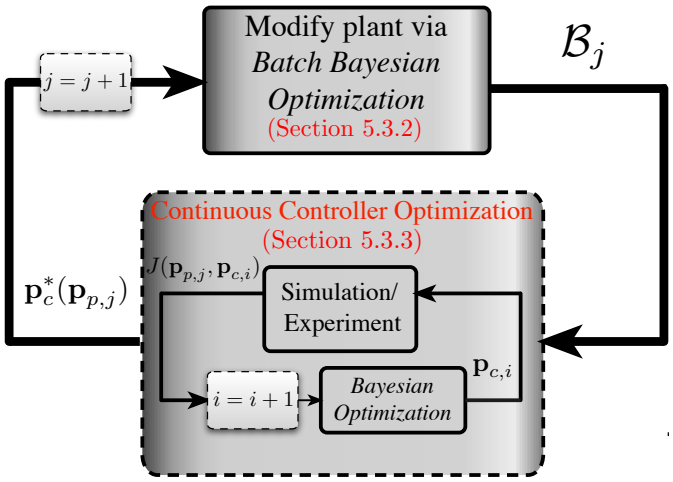


Figure 37: Machine learning variant of nested plant and controller co-design using Batch Bayesian Optimization

Algorithm 3

```
procedure FUSED PLANT AND CONTROLLER WITH BO
    while plant parameters not converged do
        Run one iteration of Bayesian Optimization in outer loop
        for plant candidate parameters do
            while control parameter not converged do
                Run full Bayesian Optimization in inner loop
                Update control parameter
            end while
        end for
    end while
end procedure
```

4.3.2 Bayesian Optimization-Based Co-Design: Batch Plant Design

In this section, we examine the use of Batch Bayesian Optimization at the plant optimization level to generate a *set* of plant designs at each iteration of the overall optimization process, recognizing that there will exist economies of scale in running multiple experiments at each outer loop iteration. To mathematically introduce the idea of Batch Bayesian Optimization, we slightly modify the notation of acquisition function presented so far. Specifically, we take $\alpha(\mathbf{p}, \mathcal{I}_{t,k})$ as the acquisition function,

where $\mathcal{I}$ represents the available data set, $\mathcal{D}$, *plus* the GP structure when n data points are available. Consequently, subscripts t and k represent the kth element of tth batch, respectfully.

The Local Penalization (LP) algorithm originally presented in [46] is a heuristic approach for Batch Bayesian Optimization that works by iteratively penalizing the current peak in the acquisition function to find the next peak. Consequently, each successive element of the batch is chosen to maximize the modified acquisition function, which has been modified based on each previous element of the batch. According to the LP algorithm, every element in t^{th} batch is given by:

$$\mathbf{p}_{t,k} = \arg\max_{\mathbf{p}\in\mathbf{P}} \left\{ g\big(\alpha(\mathbf{p}, \mathcal{I}_{t,0})\big) \prod_{j=1}^{k-1} \varphi(\mathbf{p}; \mathbf{p}_{t,j}) \right\} \tag{101}$$

where g is a differentiable transformation of α that keeps the acquisition function positive. $\varphi(\mathbf{p}; \mathbf{p}_{t,j})$ is the core component of LP algorithm and called the *local penalizer* centered at $\mathbf{p}_j$ (in the tth batch). If selected properly, the LP estimates the distance from $\mathbf{p}_j$ to the true optimum of the cost function. If we believe that the distance from $\mathbf{p}_j$ is far from the true optimum, then a large penalizer can discard a large portion of the design parameter domain that should not be considered in selecting one of the batch elements. On the other hand, if we believe that $\mathbf{p}_j$ is close to the true optimum, a small penalizer keeps collecting elements in a close neighborhood of true optimum. The main challenge lies in selecting an appropriate local penalizer.

4.3.2.1 Selecting a local penalizer

As mentioned earlier, the local penalizer characterizes a belief about the distance from the batch locations to true optimum. Let $R_M = \max_{\mathbf{p}\in\mathbf{P}} R(\mathbf{p})$. Consider the ball:

$$B_{r_j}(\mathbf{p}_j) = \left\{ \mathbf{p} \in \mathbf{P} : \left\| \mathbf{p}_j - \mathbf{p} \right\| \leq r_j \right\} \tag{102}$$

where

$$r_j = \frac{R_M - R(\mathbf{p}_j)}{L} \tag{103}$$

and L is a valid Lipschitz constant of R.

If $R(\mathbf{p}) \sim \mathcal{GP}\big(\mu(\mathbf{p}), k(\mathbf{p}, \mathbf{p}')\big)$, then we can choose a local penalizer, $\varphi(\mathbf{p}; \mathbf{p}_j)$, as the probability that $\mathbf{p}$, any point in $\mathbf{P}$ that is a potential candidate for batch elements, does not belong to $B_{r_j}(\mathbf{p}_j)$:

$$\varphi(\mathbf{p}; \mathbf{p}_j) = 1 - p\Big(\mathbf{p} \in B_{r_j}(\mathbf{p}_j)\Big) \tag{104}$$

However, we need an analytical expression for $\varphi(\mathbf{p}; \mathbf{p}_j)$ to compute each batch element. Proposition 1 provides an analytical form for the local penalizer [46]:

Proposition 1. *If $R(\boldsymbol{p}) \sim \mathcal{GP}\big(\mu(\boldsymbol{p}), k(\boldsymbol{p}, \boldsymbol{p}')\big)$, then $\varphi(\boldsymbol{p}; \boldsymbol{p}_j)$ as defined in (18), is a local penalizer at $\boldsymbol{p}_j$ such that:*

$$\varphi(\boldsymbol{p}, \boldsymbol{p}_j) = \frac{1}{2}\mathrm{erfc}(-z) \tag{105}$$

where

$$z = \frac{1}{\sqrt{2\sigma_n^2(\boldsymbol{p})}}\Big(L\left\|\boldsymbol{p} - \boldsymbol{p}_j\right\| - R_M + \mu_n(\boldsymbol{p}_j)\Big). \tag{106}$$

and erfc is the complementary error function.

Proposition 1 implies that if $\mu_n(\mathbf{p}_j)$ is close to R_M, then $\varphi(\mathbf{p}, \mathbf{p}_j)$ will have a more localized and small impact on $\alpha(\mathbf{p})$. On the other hand, if $\mu_n(\mathbf{p}_j)$ is far from R_M, then $\varphi(\mathbf{p}; \mathbf{p}_j)$ will have a big impact on $\alpha(\mathbf{p})$.

4.3.2.2 Selecting the parameters L and R_M

An appropriate local penalizer relies on valid choices for R_M and L. However, the value of R_M and L are unknown in general. To approximate R_M, one can take

$$\hat{R}_M = \max_i \left\{ R_i \right\} \tag{107}$$

Regarding the parameter L we take

$$\hat{L} = \max_{\mathbf{p} \in \mathbf{P}} \parallel \nabla \mu(\mathbf{p}) \parallel \tag{108}$$

as a valid Lipschitz constant [46].

Algorithm 4

procedure Batch Bayesian Optimization with LP
  Inputs: batch size: n_b
  for $t = 1$ until convergence **do**
    Fit a GP to $\mathcal{D}_t$
    Build the acquisition $\alpha(\mathbf{p}, \mathcal{I}_{t,0})$ using the current GP
    $\tilde{\alpha}_{t,0}(\mathbf{p}) \leftarrow g\big(\alpha(\mathbf{p}, \mathcal{I}_{t,0})\big)$
    $\hat{L} \leftarrow \max_{\mathbf{p} \in \mathbf{P}} \parallel \nabla \mu(\mathbf{p}) \parallel$
    for $j = 1$ to n_b **do**
      $\mathbf{p}_{t,j} \leftarrow \arg\max \tilde{\alpha}_{t,j-1}(\mathbf{p})$
      $\tilde{\alpha}_{t,j}(\mathbf{p}) \leftarrow \tilde{\alpha}_{t,0}(\mathbf{p}) \prod_{j=1}^{k-1} \varphi(\mathbf{p}; \mathbf{p}_{t,j})$
    end for
    $\mathcal{B}_t^{n_b} \leftarrow \left\{ \mathbf{p}_{t,1}, \ldots, \mathbf{p}_{t,n_b} \right\}$
    $R_{t,1}, \ldots, R_{t,n_b} \leftarrow$ evaluation of R at $\mathcal{B}_t^{n_b}$
    $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \left\{ \mathbf{p}_{t,j}, R_{t,j} \right\}_{j=1}^{n_b}$
  end for
end procedure

Finally, algorithm 4 summarizes the procedure presented for Batch Bayesian Op-

timization.

4.3.3 Bayesian Optimization-Based Co-Design: Continuous Controller Optimization

Bayesian Optimization is an iteration-based optimization algorithm. Thus, in order to use Bayesian Optimization as a tool for adaptation of controller parameters in a continuous-time setting, it is crucial to make a clear connection between the concept of *discrete iterations* (used in Bayesian Optimization), and *continuous time* over the course of a simulation/experiment. Each iteration in the Algorithm 3 corresponds to one window of time within the total simulation/experiment duration. We divide each of these windows into three different phases (See Fig. 38). Phase one represents the settling period. To avoid unfairly introducing system transients from one iteration to another in our cost function calculation, we allow the system to settle during the first period, then only compute the cost function value based on the second phase, which we refer to as the performance period. Finally, the calculation of the subsequent decision variable(s) occurs in the third phase, using Bayesian Optimization.

4.3.4 Convergence Detection

In any optimization problem, the stopping criteria is determined based on either a fixed iteration budget or a convergence criterion. Detecting convergence, rather than relying on a fixed iteration budget, is critical particularly where experiments come into play, since every experiment requires time and money. In this work, we set the following stopping criterion for convergence:

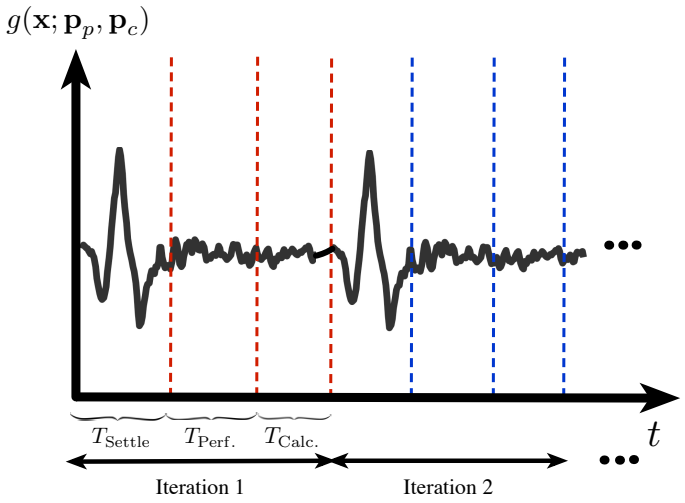


Figure 38: Clarification of iteration vs. time in the proposed framework

$$|R(\mathbf{p}_i) - R(\mathbf{p}_{i-j})| < \epsilon \quad j = 1, \ldots, n \tag{109}$$

We set n equal to 2 in this work.

4.4 Plant and Controller Design Optimization of Altaeros BAT

4.4.1 Plant Model of Altaeros BAT

The Altaeros BAT ([1]) features a horizontal axis turbine that is suspended within a buoyant shell. Unlike many AWE concepts, the BAT is designed to remain substantially stationary and passively align with the wind. By accomplishing this, the BAT can achieve secondary objectives, such as telecommunications, which Altaeros has publicly indicated interest in. To that end, it is of great interest to design the combined plant and control system to achieve the steadiest possible flight under atmospheric disturbances, which is the focus of our case study.

The dynamic model of Altaeros BAT system was originally introduced in [78]. We briefly introduce some of the important features of this model here. Fig. 39 shows

the variables used in the dynamic model.

The model, which was derived using an Euler-Lagrange framework, describes the position and orientation of the BAT through six generalized coordinates: Θ, ϕ, ψ, L_t, $\theta^{'}$, and $\phi^{'}$. Three of these generalized coordinates represent uncontrolled angles. Specifically, Θ is the azimuth angle (angle of the tether projection on the horizontal plane), Φ is the zenith angle (angle of tether with respect to vertical), and Ψ is the twist angle (about the tether axis). Table 9 represents the full state variables of AWE system.

The three tethers are modeled as a single tether with length L_t with a bridle joint at the top that provided three attachment points on the BAT. The bridle joint is modeled through two angles, $\phi^{'}$ and $\theta^{'}$, which are referred to as induced roll and induced pitch, respectively. The single tether approximation removes algebraic constraint equations, allowing the system to be described by ordinary differential equations (ODEs) rather than differential algebraic equations (DAEs). The center of mass location is modeled as a function of Φ (zenith angle), Θ (azimuth angle), and L_t (average tether length). The induced angles are related to the tether lengths (l_1, l_2, and l_3) through the following expressions:

$$\phi' = tan^{-1}\Big(\frac{l_3 - l_2}{l_{\text{sep}}^{\text{lat}}}\Big) \tag{110}$$

$$\theta' = tan^{-1}\Big(\frac{l_1 - 0.5(l_2 + l_3)}{l_{\text{sep}}^{\text{long}}}\Big) \tag{111}$$

where $l_{\text{sep}}^{\text{long}}$ and $l_{\text{sep}}^{\text{lat}}$ are longitudinal and lateral tether attachment separation distances, respectively. l_1, l_2 and l_3 are the distances between the bridal joint and the three tether attachment points. The control inputs are the tether release speeds, $\bar{u}_i$, which are

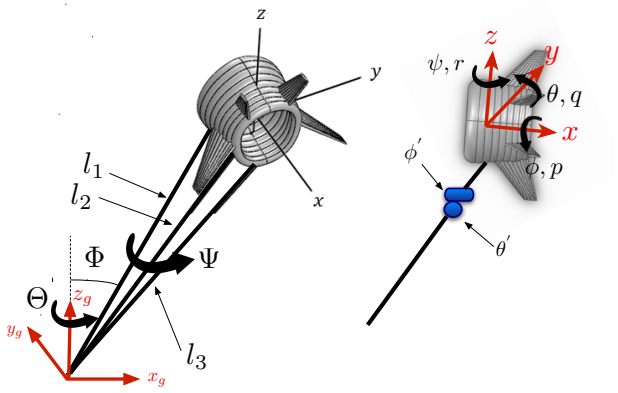


Figure 39: Ground-fixed and body-fixed coordinates plus the key variables used in deriving Euler-Lagrangian dynamics.

given by:

$$\bar{u}_i = \frac{d}{dt} l_i \tag{112}$$

Ultimately, the governing system equations given below are derived using an Euler-Lagrange formulation and are expressed by:

$$D(Q)\ddot{Q} + C(Q,\dot{Q})\dot{Q} + g(Q) = \tau\big(Q, \dot{Q}, V_{\text{wind}}, \psi_{\text{wind}}\big), \tag{113}$$

$$X = f(Q,\dot{Q}), \tag{114}$$

$$\Omega = g(Q,\dot{Q}), \tag{115}$$

where:

$$Q = [\Phi \;\; \Theta \;\; \Psi \;\; L_t \;\; \theta' \;\; \phi'] \tag{116}$$

$$X = [x \;\; y \;\; z \;\; u \;\; v \;\; w] \tag{117}$$

$$\Omega = [\phi \;\; \theta \;\; \psi \;\; p \;\; q \;\; r] \tag{118}$$

Here, V_{wind} is the wind speed, and ψ_{wind} and τ represent the wind direction and vector of generalized aerodynamic forces and moments, respectively. Aerodynamic forces and moments are functions of α (angle of attack) and β (side slip angle), which describe the orientation of the apparent wind vector with respect to the body-fixed coordinates of the BAT. X and Ω represent the translational and rotational dynamics, respectively. Since we treat the horizontal stabilizer area, A_H, as a design parameter to be optimized, the aerodynamic coefficients are modeled as explicit functions of the stabilizer areas as follows:

$$\begin{aligned} C_{D,L,S}(\alpha,\beta) &= C^F_{D,L,S}(\alpha,\beta) + C^H_{D,L,S}(\alpha,\beta)\frac{A^H}{A_{ref}} \\ &\quad + C^V_{D,L,S}(\alpha,\beta)\frac{A^V}{A_{ref}} \qquad (119) \\ C_{M_x,M_y,M_z}(\alpha,\beta) &= C^F_{M_x,M_y,M_z}(\alpha,\beta) \\ &\quad + C^H_{M_x,M_y,M_z}(\alpha,\beta)\frac{A^H l^H}{A_{ref} l_{ref}} \\ &\quad + C^V_{M_x,M_y,M_z}(\alpha,\beta)\frac{A^V l^V}{A_{ref} l_{ref}} \qquad (120) \end{aligned}$$

Here, C_D, C_L, and C_S represent the drag, lift, and side force coefficients, whereas C_{M_x}, C_{M_y}, and C_{M_z} represent the roll, pitch, and yaw moment coefficients.

4.4.2 Closed-Loop Controller

The controller is designed to track three different set points, namely altitude (z_{sp}), pitch (θ_{sp}) and roll (ϕ_{sp}). Due to the symmetrical configuration, ϕ_{sp} is always set to zero. The block diagram of the controller is shown in Fig. 40. Bayesian Optimization is used to update θ_{sp}, which is programmed/adjusted internally on the flight computer

Table 9: Full state variables of an AWE system

State variable	Notation
Zenith angle	Φ
Azimuth angle	Θ
Twist angle	Ψ
Zenith angle rate	$\dot{\Phi}$
Azimuth angle rate	$\dot{\Theta}$
Twist angle rate	$\dot{\Psi}$
Unstreched tether length	l
Induced roll	ϕ'
Induced pitch	θ'

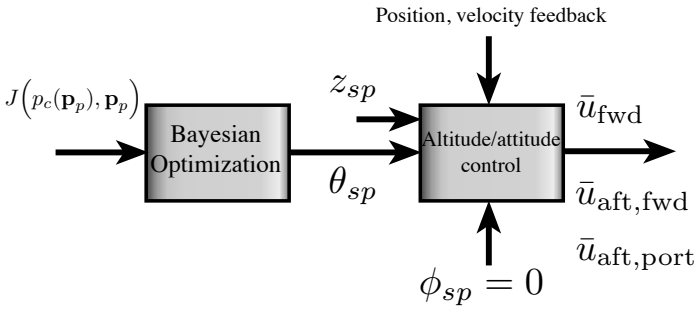


Figure 40: Block diagram of closed loop flight controller for the BAT. z_{sp} denotes a constant altitude set-point. We choose $p_c = \theta_{sp}$ and $\mathbf{p}_p = [x_{cm} - x_{cb}\ A_H]^T$ in our case study results.

and is not a user-specified value (hence, θ_{sp} represents a control parameter). Tracking is achieved through the use of three lead filtered PD controllers, which individually control altitude, pitch angle, and roll angle. The outputs of these three lead filters are denoted, respectively, by $\bar{v}_z$, $\bar{v}_\theta$, and $\bar{v}_\phi$. These outputs represent the average tether release speed (used to regulate altitude), forward/aft tether speed difference (used to regulate pitch), and port/starboard tether speed difference (used to regulate roll), respectively. These controller outputs are related to the tracking errors in altitude

(z_e), pitch (θ_e) and roll (ϕ_e) through:

$$\bar{v}_z(s) = \frac{k_d^z s + k_p^z}{\tau_z s + 1} z_e(s), \tag{121}$$

$$\bar{v}_\theta(s) = \frac{k_d^\theta s + k_p^\theta}{\tau_\theta s + 1} \theta_e(s), \tag{122}$$

$$\bar{v}_\phi(s) = \frac{k_d^\phi s + k_p^\phi}{\tau_\phi s + 1} \phi_e(s). \tag{123}$$

The tether release speeds $\bar{u}_{\text{center}}$, $\bar{u}_{\text{stbd}}$, and $\bar{u}_{\text{port}}$ serve as control inputs to three motors and are related to $\bar{v}_z$, $\bar{v}_\theta$, and $\bar{v}_\phi$ through:

$$\begin{bmatrix} \bar{u}_{\text{center}} \\ \bar{u}_{\text{stbd}} \\ \bar{u}_{\text{port}} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \bar{v}_z \\ \bar{v}_\theta \\ \bar{v}_\phi \end{bmatrix}. \tag{124}$$

For our case study results, we focus on the following plant parameters to be optimized:

$$\mathbf{p}_p = \left\{ x_{cm} - x_{cb}, \quad A_H \right\} \tag{125}$$

where $x_{cm} - x_{cb}$ describes the longitudinal center of mass position relative to the center of buoyancy and A_H represents the horizontal stabilizer area.

The controller parameter used in this work is given by:

$$p_c = \theta_{sp} \tag{126}$$

where θ_{sp} presents the trim pitch angle.

4.4.3 Performance Index

The performance index, to be minimized, takes into account two main system properties:

1. *Ground footprint*: It is of interest to minimize the land usage requirements for multiple systems. For this purpose, the horizontal projected area of land that BAT covers is used as a criterion for quantifying the ground footprint. This area is represented by $A = \pi l^2 \sin^2 \Phi$. Thus, as Φ decreases, the projected area decreases.

2. *Quality of flight*: A low value of heading angle typically corresponds to few oscillations in the system and desirable direct-downwind operation. Furthermore, since we are focused on steady, level flight, we desire to have the BAT as stationary as possible. To characterize the degree to which we accomplish this goal, we penalize heading and roll angle tracking error ($\psi - \psi_{\text{flow}}$ and $\phi - \phi_{\text{sp}}$, respectively) in our performance metric.

Ultimately, the performance index is denoted by:

$$J\Big(p_c(\mathbf{p}_p), \mathbf{p}_p\Big) = \int_{T_i}^{T_f} \Big(k_1 \Phi^2 + k_2(\psi - \psi_{\text{flow}})^2 + k_3(\phi - \phi_{\text{sp}})^2)\Big) dt, \tag{127}$$

4.4.4 Simulation Setup

To excite the system with a wind environment that is consistent across simulations, we implement a frequency approximation of vortex shedding of flow over a cylinder. The model perturbation is based on spectral analysis of flow over a cylinder in [70].

Each component of the velocity is approximated as a sinusoidal perturbation about a mean velocity found in [70]. Each of the velocity components is given by:

$$v_x = v_x^{\text{base}} + v_{x0}\sin(\omega_{\text{dist}}t) \quad (128)$$

$$v_y = v_{y0}\sin(\omega_{\text{dist}}t) \quad (129)$$

$$v_z = v_{z0}\sin(\omega_{\text{dist}}t) \quad (130)$$

where $v_{\text{base}} = 0.606\frac{m}{s}$, $v_{x0} = 0.0866\frac{m}{s}$, $v_{y0} = 0.065\frac{m}{s}$, $v_{z0} = 0.0087\frac{m}{s}$, and $\omega_{\text{dist}} = 1HZ$.

This mechanism for perturbing the system is attractive because it (i) excites key lateral dynamics and (ii) can be implemented in later lab-scale experimental co-design using the team's water channel setup and pictured in Fig. 46.

4.5 Results

We evaluated the proposed algorithm on the BAT numerical model. We assessed the effectiveness of Batch Bayesian Optimization algorithm for 3 and 4 elements (plant designs) in each batch. We also compare results from Batch Bayesian Optimization against results for a batch size of 1 (i.e., generic Bayesian Optimization) [18].

Fig. 41 shows the convergence of plant parameters (horizontal stabilizer area and relative center of mass), control parameter (trim pitch angle), and integral of cost function for different batch sizes. Each trajectory in this figure represents the evolution of a different batch element over the course of optimization.

One can immediately see that fewer total iterations are required to converge with larger batch sizes. Furthermore, as can be seen from this figure, the plant parameters

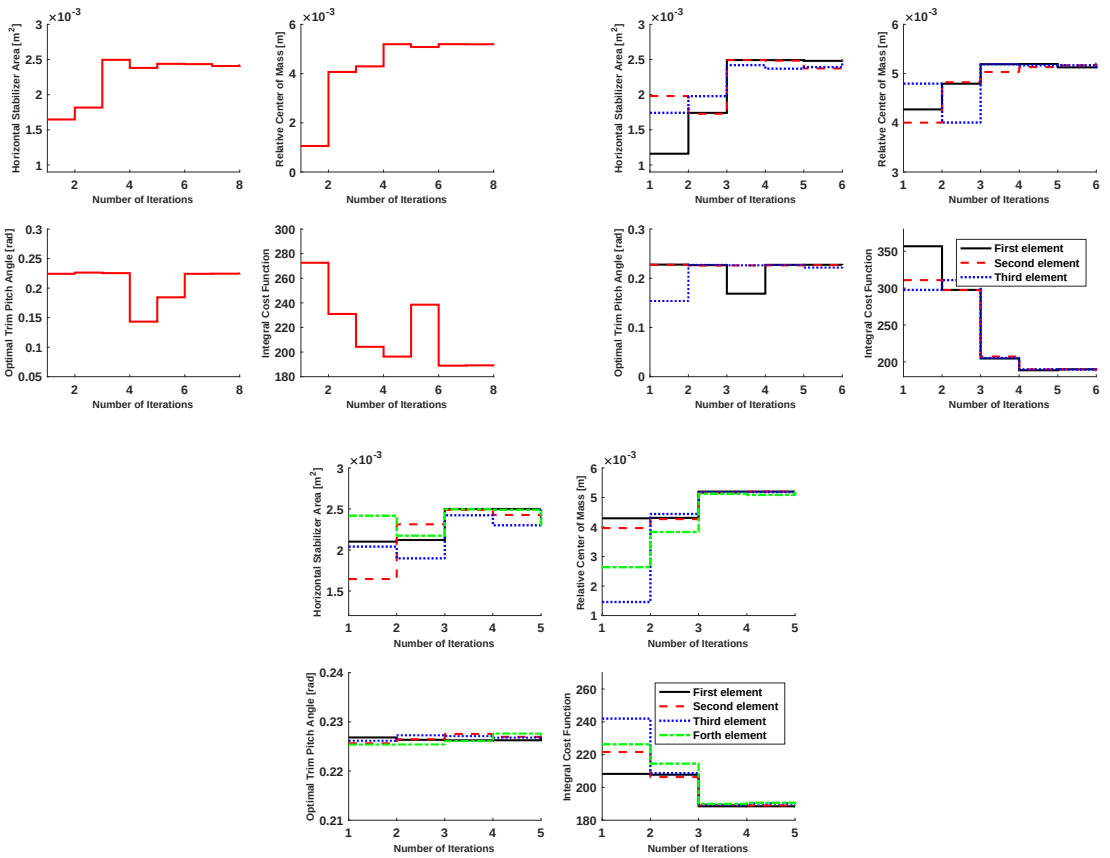


Figure 41: Convergence of plant parameters, control parameter, and integral cost function for 1, 3, and 4 batch sizes (from upper left to lower right)

converge after only a small number of iterations (each iteration corresponds to one round of system performance evaluation). With the same simulation setup, comparing these results with those reported in [32] reveals that Bayesian Optimization leads to a faster convergence than the optimal DoE proposed in that work. It should be emphasized that plant design changes are much more expensive compared to adjusting control parameters in instances when simulations are replaced with experiments (which is a long-term goal of the present work). Therefore, reducing the number of required plant reconfigurations in the design optimization process is a very important goal.

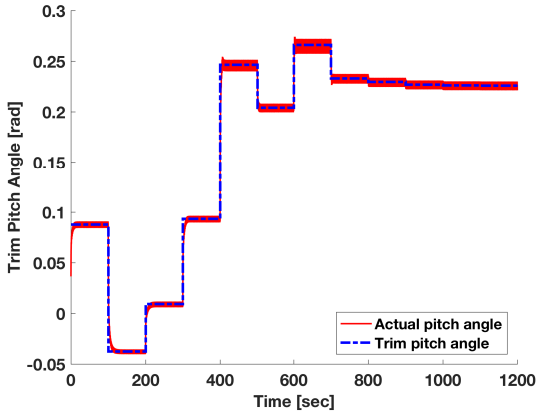


Figure 42: Sample evolution of control parameter (i.e., trim pitch angle) and actual pitch angle in inner loop over the time

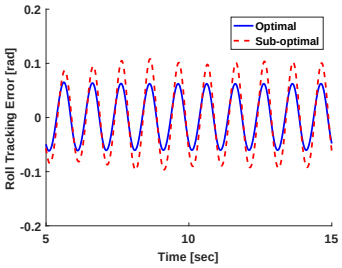


Figure 43: (Zoomed) roll tracking error before and after the optimization

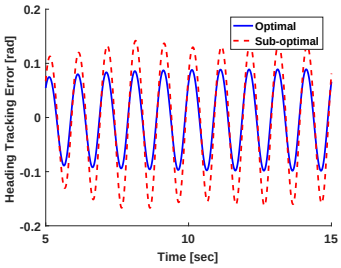


Figure 44: (Zoomed) heading tracking error before and after the optimization

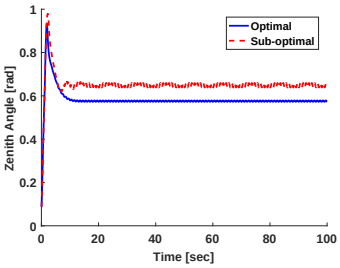


Figure 45: Zenith angle before and after the optimization

Fig. 41 also illustrates the optimal control parameter in the inner loop for each plant design generated by the outer loop. Furthermore, Fig. 42 represents a sample evolution of the control parameter (trim pitch angle) and actual pitch angle (which is regulated by a lower-level flight controllers) in the inner loop over a simulation.

Figs. 43, 44, and 45 illustrate the evolution of the individual instantaneous cost function components (i.e., roll tracking error, heading tracking error, and zenith angle) before and after the iterative optimization process to demonstrate the effectiveness of the proposed approach.

4.6 Economies of Scale

So far, we have explored the idea of Batch Bayesian Optimization at the plant optimization level to generate a set of plant designs at each iteration of the overall optimization process. In this section, we will assess, for the AWE system, whether this will result in economies of scale.

To assess the economies of scale that are introduced through a Batch Bayesian Optimization process, we focus our attention on a lab-scale experimental platform (depicted in Fig. 46) for closed-loop flight characterization of AWE systems. With this system, 3D printed models (depicted in Fig. 47) of AWE system lifting bodies are tethered and flown in the UNC-Charlotte water channel. Micro DC motors are used to regulate tether lengths, high-speed cameras are used for real-time image capture, and a high-performance target computer is used for real-time image processing and control. Because this experimental platform represents the ultimate "end game" for the Batch Bayesian Optimization approach, we use it as the basis for the economies of scale analysis presented herein.

In order to run an experimental batch (and non-batch) Bayesian Optimization in water channel, the following steps need to happen:

- Prior to running any experiments: 3D print the main body.

  The cost associated with this step is modeled as:

$$C_{3D} = \Big(m \times c_{\text{eng}} + c_{\text{recharge}}\Big) N T_{\text{print}} \tag{131}$$

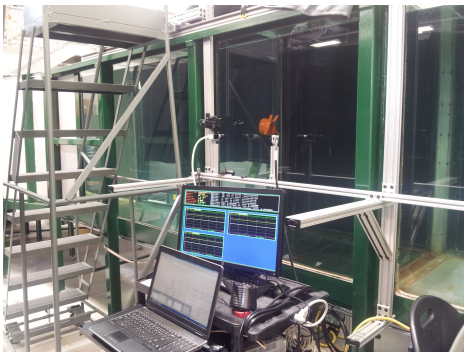


Figure 46: Water channel experimental setup at UNC Charlotte

Figure 47: $\frac{1}{100}$-scale BAT model

Table 10: Design parameters for economic assessment

Parameter	**Description**	**Value**
c_{eng}	cost required to hire employee(s) to conduct experiments	$\frac{\$30}{\text{hours}}$
c_{recharge}	cost required for equipment recharge	$\frac{\$240}{\text{day}}$
$c_{\text{Wrecharge}}$	cost required for water channel recharge	$\frac{\$2400}{\text{day}}$
c_{lostTime}	opportunity cost of lost time	$\frac{\$1200}{\text{day}}$
T_{print}	time required to print the main body	12 hours
T_{3DFins}	time required to print the fins	4 hours
T_{lead}	lead time	5 days
T_{setup}	time required to setup experiment	30 min
T_{reconfig}	time required to reconfigure the model	5 min
T_{exp}	time required to conduct one experiment in inner loop	1200 sec
N	number of elements per batch	1, 3, 4
m	number of employee(s) required to print the model and fins	1
m'	number of employee(s) required to conduct experiments	2

Table 11: Cost in $ associated with running experiments with different batch sizes

# of elements per batch	**Cost ($)**
1	54293
3	49200
4	44533

where c_{eng} and c_{recharge} represent the cost required for engineer(s) to conduct experiments and the facility charge, respectively. T_{print} denotes the time required to print the main body. m is the number of employees required to print the model (and fins).

- After each batch is chosen, the following needs to happen once per batch:

  (i) 3D print N sets of fins. This cost is given by:

$$C_{3D\text{Fins}} = \Big(m \times c_{\text{eng}} + c_{\text{recharge}}\Big) N T_{3D\text{Fins}} \tag{132}$$

  where $T_{3D\text{Fins}}$ represents the time required to print the fins.

  (ii) Schedule future time in the water channel. While the act of scheduling testing time does not result in a direct cost, the lead time involved in waiting for a testing slot results an opportunity cost, which is modeled as:

$$C_{\text{lead}} = c_{\text{lostTime}} T_{\text{lead}} \tag{133}$$

  where c_{lostTime} and T_{lead} represent the opportunity cost of lost time and the lead time, respectively.

  (iii) Initialize the water channel:

$$\begin{aligned} C_{\text{Wchannel}} = \Big(m' \times c_{\text{eng}} + c_{\text{Wrecharge}}\Big)\Big(T_{\text{setup}} \\ + N T_{\text{exp}} + N T_{\text{reconfig}}\Big) \end{aligned} \tag{134}$$

  where T_{setup} and T_{reconfig} denote the time required to set up the experiment in

the water channel and the time required to reconfigure the model, respectively. Furthermore, $c_{\text{Wrecharge}}$ and T_{exp} represent the cost of running the water channel and time required to conduct an experiment in the inner loop, respectively. m' is the number of employees required to conduct experiments.

- Between each experiment within a batch, the following needs to happen N times per batch:

  (i) Swap out fins; (ii) Adjust center of mass; (iii) Initialize image processing for a given configuration; (iv) Run the experiment.

Table 10 summarizes the design parameters for economic assessment. The total cost can be ultimately expressed as:

$$C_{\text{total}} = N_{\text{convergence}} \times \Big(C_{3D} + C_{3D\text{Fins}} + C_{\text{lead}} + C_{\text{Wchannel}} \Big) \tag{135}$$

where $N_{\text{convergence}}$ denotes the number of iterations required for convergence based on the number elements per batch. As can be seen from C_{total}, some terms depend on N, while some terms do not. This indicates that there exist economies of scale in running Batch Bayesian Optimization. Table 11 shows the cost associated with different batch sizes, according to our economic assessment. One can conclude from this table that as the number of elements per batch increases, the associated cost decreases.

REFERENCES

[1] Altaeros Energies, inc., http://www.altaerosenergies.com.

[2] https://daidalos-capital.com.

[3] https://ecotricity.co.uk.

[4] http://windenergyfoundation.org/.

[5] http://www.windpowerengineering.com/.

[6] Kawasaki, http://global.kawasaki.com.

[7] U.s. Energy Information Administration, https://www.eia.gov.

[8] U. Ahrens, M. Diehl, and R. Schmehl. *Airborne wind energy*. Springer Science & Business Media, 2013.

[9] J. T. Allison, T. Guo, and Z. Han. Co-design of an active suspension using simultaneous dynamic optimization. *Journal of Mechanical Design*, 136(8):081003, 2014.

[10] J. T. Allison and S. Nazari. Combined plant and controller design using decomposition-based design optimization and the minimum principle. In *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 765–774. American Society of Mechanical Engineers, 2010.

[11] C. Archer. Wind profiler at Cape Henlopen. Lewes, Delaware, http://www.ceoe.udel.edu/our-people/profiles/carcher/fsmw. Technical report, University of Delaware, 2014.

[12] T. W. Athan and P. Y. Papalambros. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27(2):155–176, 1996.

[13] A. Azadeh, M. Sheikhalishahi, and S. Asadzadeh. A flexible neural network-fuzzy data envelopment analysis approach for location optimization of solar plants with uncertainty and complexity. *Renewable energy*, 36(12):3394–3401, 2011.

[14] A. Bafandeh, S. Bin-Karim, A. Baheri, and C. Vermillion. A comparative assessment of hierarchical control structures for spatiotemporally-varying systems, with application to airborne wind energy. *Control Engineering Practice*, 74:71–83, 2018.

[15] A. Bafandeh and C. Vermillion. Real-time altitude optimization of airborne wind energy systems using Lyapunov-based switched extremum seeking control. In *American Control Conference*, Boston, US, 2016.

[16] A. Baheri. Parameters estimation for state-of-power (SOP) prediction of lithium-ion batteries: A Bayesian Optimization approach. In *Thecnical report*, 2017.

[17] A. Baheri, S. Bin-Karim, A. Bafandeh, and C. Vermillion. Real-time control using Bayesian Optimization: A case study in airborne wind energy systems. *Control Engineering Practice*, 69:131–140, 2017.

[18] A. Baheri, J. Deese, and C. Vermillion. Combined plant and controller design using Bayesian Optimization: A case study in airborne wind energy systems. In *ASME 2017 Dynamic Systems and Control Conference*, pages V003T40A003–V003T40A003. American Society of Mechanical Engineers, 2017.

[19] A. Baheri, P. Ramaprabhu, and C. Vermillion. Iterative in-situ 3D layout optimization of a reconfigurable ocean current turbine array using bayesian optimization. In *ASME 2017 Dynamic Systems and Control Conference*, pages V003T40A002–V003T40A002. American Society of Mechanical Engineers, 2017.

[20] A. Baheri, P. Ramaprabhu, and C. Vermillion. Iterative 3D layout optimization and parametric trade study for a reconfigurable ocean current turbine array using bayesian optimization. *Renewable Energy*, 2018.

[21] A. Baheri and C. Vermillion. Altitude optimization of airborne wind energy systems: A Bayesian Optimization approach. In *American Control Conference*, Seattle, US, 2017.

[22] A. Baheri and C. Vermillion. Context-dependent Bayesian Optimization in real-time optimal control: A case study in airborne wind energy systems. In *NIPS Workshop on Bayesian Optimization*, Long Beach, CA, 2017.

[23] M. Bilbao and E. Alba. Simulated annealing for optimization of wind farm annual profit. In *2009 2nd International Symposium on Logistics and Industrial Informatics*, Linz, AU, 2009.

[24] S. Bin-Karim, A. Bafandeh, A. Baheri, and C. Vermillion. Spatiotemporal optimization through gaussian process-based model predictive control: A case study in airborne wind energy. *IEEE Transactions on Control Systems Technology*, 2017.

[25] E. Brochu, V. M. Cora, and N. De Freitas. A tutorial on Bayesian Optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[26] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. P. Deisenroth. Bayesian gait optimization for bipedal locomotion. In *International Conference on Learning and Intelligent Optimization*, Gainesville, US, 2014.

[27] M. Canale, L. Fagiano, and M. Milanese. High altitude wind energy generation using controlled power kites. *IEEE Transactions on Control Systems Technology*, 18(2):279–293, 2010.

[28] S. Chowdhury, J. Zhang, A. Messac, and L. Castillo. Unrestricted wind farm layout optimization (UWFLO): Investigating key factors influencing the maximum power generation. *Renewable Energy*, 38(1):16–30, 2012.

[29] D. D. Cox and S. John. Sdo: A statistical method for global optimization. In *in Multidisciplinary Design Optimization: State-of-the-Art*, pages 315–329, 1997.

[30] I. G. Damousis, M. C. Alexiadis, J. B. Theocharis, and P. S. Dokopoulos. A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation. *IEEE Transactions on Energy Conversion*, 19(2):352–361, 2004.

[31] I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997.

[32] J. Deese, N. Deodhar, and C. Vermillion. Nested plant / controller co-design using G-optimal design and extremum seeking: Theoretical framework and application to an airborne wind energy system. In *International Federation of Automatic Control World Congress*, Toulouse, FR, 2017.

[33] J. Deese, P. Razi, M. Muglia, P. Ramaprabhuu, and C. Vermillion. Fused closed-loop flight dynamics and wake interaction model. In *ASME 2018 Dynamic Systems and Control Conference*, 2018.

[34] J. Deese and C. Vermillion. Nested plant/controller co-design using G-optimal design and continuous time adaptation laws: Theoretical framework and application to an airborne wind energy system. *Submitted to ASME Journal on Dynamic Systems, Measurement, and Control*.

[35] B. L. Du Pont and J. Cagan. An extended pattern search approach to wind farm layout optimization. *Journal of Mechanical Design*, 134(8):081002, 2012.

[36] Y. Eroğlu and S. U. Seçkiner. Design of wind farm layout using ant colony algorithm. *Renewable Energy*, 44:53–62, 2012.

[37] Y. Eroğlu and S. U. Seçkiner. Wind farm layout optimization using particle filtering approach. *Renewable Energy*, 58:95–107, 2013.

[38] L. Fagiano, M. Milanese, and D. Piga. Optimization of airborne wind energy generators. *International Journal of robust and nonlinear control*, 22(18):2055–2083, 2012.

[39] H. K. Fathy, S. A. Bortoff, G. S. Copeland, P. Y. Papalambros, and A. G. Ulsoy. Nested optimization of an elevator and its gain-scheduled LQG controller. In *ASME 2002 International Mechanical Engineering Congress and Exposition*, pages 119–126. American Society of Mechanical Engineers, 2002.

[40] H. K. Fathy, P. Y. Papalambros, and A. G. Ulsoy. Integrated plant, observer, and controller optimization with application to combined passive/active automotive suspensions. In *ASME 2003 International Mechanical Engineering Congress and Exposition*, pages 225–232. American Society of Mechanical Engineers, 2003.

[41] H. K. Fathy, P. Y. Papalambros, A. G. Ulsoy, and D. Hrovat. Nested plant/controller optimization with application to combined passive/active automotive suspensions. In *American Control Conference, 2003. Proceedings of the 2003*, volume 4, pages 3375–3380. IEEE, 2003.

[42] H. K. Fathy, J. A. Reyer, P. Y. Papalambros, and A. Ulsov. On the coupling between the plant and controller optimization problems. In *American Control Conference, 2001. Proceedings of the 2001*, volume 3, pages 1864–1869. IEEE, 2001.

[43] S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann, S. Larsen, J. Højstrup, and M. Thøgersen. Analytical modelling of wind speed deficit in large offshore wind farms. *Wind energy*, 9(1-2):39–53, 2006.

[44] S. W. Funke, P. E. Farrell, and M. Piggott. Tidal turbine array optimisation using the adjoint approach. *Renewable Energy*, 63:658–673, 2014.

[45] M. G. Gebreslassie, M. R. Belmont, and G. R. Tabor. Comparison of analytical and CFD modelling of the wake interactions of tidal turbines. In *10th European Wave and Tidal Energy Conference (EWTEC2013)*, 2013.

[46] J. González, Z. Dai, P. Hennig, and N. Lawrence. Batch Bayesian Optimization via local penalization. In *Artificial Intelligence and Statistics*, pages 648–657, 2016.

[47] J. González, M. Osborne, and N. D. Lawrence. Glasses: Relieving the myopia of Bayesian optimisation. *arXiv preprint arXiv:1510.06299*, 2015.

[48] W. Guoyang, X. Yang, and W. Shasha. Discussion about short-term forecast of wind speed on wind farm. *Jilin Electric Power*, 181(5):21–24, 2005.

[49] M. W. Isaacs, J. B. Hoagg, I. I. Hussein, and D. Olinger. Retrospective cost adaptive control for a ground tethered energy system. In *50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, US, 2011.

[50] A. Z. J. Mockus, V. Tiesis. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.

[51] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[52] I. Katic, J. Højstrup, and N. O. Jensen. A simple model for cluster efficiency. In *European wind energy association conference and exhibition*, pages 407–410, 1986.

[53] M. M. Khajah, B. D. Roads, R. V. Lindsey, Y.-E. Liu, and M. C. Mozer. Designing engaging games using Bayesian Optimization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5571–5582. ACM, 2016.

[54] A. Krause and C. S. Ong. Contextual Gaussian Process bandit optimization. In *Advances in Neural Information Processing Systems*, pages 2447–2455, 2011.

[55] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

[56] A. Kusiak and Z. Song. Design of wind farm layout for maximum wind energy capture. *Renewable Energy*, 35(3):685–694, 2010.

[57] S. Li. Wind power prediction using recurrent multilayer perceptron neural networks. In *Power Engineering Society General Meeting, 2003, IEEE*, volume 4. IEEE, 2003.

[58] A. Marvuglia and A. Messineo. Monitoring of wind farms' power curves using machine learning techniques. *Applied Energy*, 98:574–583, 2012.

[59] M. Mohandes, T. Halawani, S. Rehman, and A. A. Hussain. Support vector machines for wind speed prediction. *Renewable Energy*, 29(6):939–947, 2004.

[60] M. Muglia, R. He, C. Lowcher, J. Bane, Y. Gong, and P. Taylor. Gulf Stream marine hydrokinetic energy resource characterization off Cape Hatteras, North Carolina. In *AGU Fall Meeting Abstracts*, 2015.

[61] A. Munoz. Machine learning and optimization. *URL: https://www. cims. nyu. edu/˜ munoz/files/ml_optimization. pdf [accessed 2016-03-02][WebCite Cache ID 6fiLfZvnG]*, 2014.

[62] L. Myers, A. Bahaj, C. Retzler, P. Ricci, and J.-F. Dhedin. Inter-device spacing issues within wave and tidal energy converter arrays. In *Third International Conference and Exhibition on Ocean Energy*, Bilbao, ES, 2010.

[63] V. Nguyen, S. Gupta, S. Rana, C. Li, and S. Venkatesh. Regret for expected improvement over the best-observed value and stopping condition. In *Asian Conference on Machine Learning*, pages 279–294, 2017.

[64] P. NikpoorParizi, N. Deodhar, and C. Vermillion. Combined plant and controller performance analysis and optimization for an energy-harvesting tethered wing. In *American Control Conference (ACC), 2016*, pages 4089–4094. American Automatic Control Council (AACC), 2016.

[65] K. S. Perera, Z. Aung, and W. L. Woon. Machine learning techniques for supporting renewable energy generation and integration: a survey. In *International Workshop on Data Analytics for Renewable Energy Integration*, pages 81–96. Springer, 2014.

[66] D. L. Peters, P. Papalambros, and A. Ulsoy. Control proxy functions for sequential design and control optimization. *Journal of Mechanical Design*, 133(9):091007, 2011.

[67] N. Rahman, S. Badshah, A. Rafai, and M. Badshah. Literature review of ocean current turbine. *International Journal of Scientific & Engineering Research*, 5(11):11, 2014.

[68] C. E. Rasmussen. *Gaussian processes for machine learning*. Number ISBN 0-262-18253-X. The MIT Press, 2006.

[69] J. A. Reyer and P. Y. Papalambros. Optimal design and control of an electric DC motor. In *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, pages 85–96. Citeseer, 1999.

[70] M. Sarioğlu and T. Yavuz. Vortex shedding from circular and rectangular cylinders placed horizontally in a turbulent flow. *Turkish Journal of Engineering and Environmental Sciences*, 24(4):217–228, 2000.

[71] T. Senjyu, D. Hayashi, A. Yona, N. Urasaki, and T. Funabashi. Optimal configuration of power generating systems in isolated island with renewable energy. *Renewable Energy*, 32(11):1917–1933, 2007.

[72] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian Optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[73] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[74] A. Uihlein and D. Magagna. Wave and tidal current energy–a review of the current state of research beyond technology. *Renewable and Sustainable Energy Reviews*, 58:1070–1081, 2016.

[75] J. H. VanZwieten, N. Vanrietvelde, and B. L. Hacker. Numerical simulation of an experimental ocean current turbine. *IEEE Journal of Oceanic Engineering*, 38(1):131–143, 2013.

[76] Z. VanZwieten, P. Pyakurel, T. Ngo, and C. Sultan. An assessment of using variable blade pitch for moored ocean current turbine flight control. *International Journal of Marine Energy*, 13:16–26, 2016.

[77] C. Vermillion. Altitude and crosswind motion control for optimal power-point tracking in tethered wind energy systems with airborne power generation. Palo Alto, US, 2013. ASME 2015 Dynamic Systems and Control Conference.

[78] C. Vermillion, B. Glass, and S. Greenwood. Evaluation of a water channel-based platform for characterizing aerostat flight dynamics: A case study on a lighter-than-air wind energy system. In *Lighter-Than-Air Systems Conference, AIAA (Part of Aviation 2014)*, 2014.

[79] C. Vermillion, T. Grunnagle, and I. Kolmanovsky. Modeling and control design for a prototype lighter-than-air wind energy system. In *2012 American Control Conference*, Monteral, Canada, 2012. IEEE.

[80] C. Vermillion, T. Grunnagle, R. Lim, and I. Kolmanovsky. Model-based plant design and hierarchical control of a prototype lighter-than-air wind energy system, with experimental flight test results. *IEEE Transactions on Control Systems Technology*, 22(2):531–542, 2014.

[81] C. Vermillion, T. Grunnagle, R. Lim, and I. Kolmanovsky. Model-based plant design and hierarchical control of a prototype lighter-than-air wind energy system, with experimental flight test results. *IEEE Transactions on Control Systems Technology*, 22:531–542, 2014.

[82] Z. Wang and N. de Freitas. Theoretical analysis of Bayesian Optimisation with unknown Gaussian Process hyper-parameters. *arXiv preprint arXiv:1406.7758*, 2014.

[83] P. Williams, B. Lansdorp, and W. Ockels. Modeling and control of a kite on a variable length flexible inelastic tether. In *AIAA Guidance, Navigation and Control Conference*, Hilton Head, US, 2007.

[84] K. Youcef-Toumi. Modeling, design, and control integration: a necessary step in mechatronics. *IEEE/ASME Transactions on Mechatronics*, 1(1):29–38, 1996.