NESTED OPTIMIZATION FRAMEWORK FOR FUSING ITERATIVE PLANT OPTIMIZATION WITH CONTINUOUS CONTROLLER OPTIMIZATION

by

Joseph Deese

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

2020

Approved by:

____

Dr. Christopher Vermillion

____

Dr. Peter Tkacik

____

Dr. Scott Kelly

____

Dr. Sukumar Kamalasadan

____

Dr. Tao Hong

ABSTRACT

JOSEPH DEESE. NESTED OPTIMIZATION FRAMEWORK FOR FUSING ITERATIVE PLANT OPTIMIZATION WITH CONTINUOUS CONTROLLER OPTIMIZATION. (Under the direction of DR. CHRISTOPHER VERMILLION)

This research establishes a novel combined plant and controller optimization (termed *co-design*) framework, aimed at complex systems, that speeds up the optimization process by adjusting controller parameters *during* an experiment while plant parameters are adjusted *between* batches of experiments. Most legacy co-design approaches have been restricted to analytical or numerical approaches that require full knowledge of the system dynamics and do not leverage the unique ability to optimize control parameters in real time. To address these challenges, the proposed nested co-design framework relies on an iterative outer loop to adjust the plant design and an inner loop that optimizes the control parameters during an experiment. Following each round of experiments, performance of the dynamical system is characterized across the design space, along with a statistical characterization of uncertainty. Using these characterizations, the design space is reduced prior to proceeding to subsequent iterations. The process is repeated until the design space has been sufficiently reduced. This dissertation evaluates a variety of candidate methodologies for both the outer-loop plant iteration and inner-loop control adaptation, including statistical design of experiments, Gaussian Process (GP) modeling, and more traditional adaptive control techniques. The dissertation focuses heavily on the fusion of a GP-based technique for plant iteration and a novel recursive GP (RGP)-based adaptive control technique for control parameter adaptation. Not only does this result in a formulation where the inner-loop control parameter adaptation mirrors the iteration-based plant parameter adaptation;

the RGP-based adaptation technique also represents a standalone contribution to the adaptive control literature. All variations of this framework have been validated in simulations and/or lab-scale experiments for an airborne wind energy system. The results in this dissertation demonstrate the efficacy of the nested co-design framework in efficiently converging to optimal design parameters.

ACKNOWLEDGMENTS

Over the past several years, I have had the great privilege to study under my advisor, Dr. Chris Vermillion. I am grateful for his guidance and support throughout my studies as a PhD student. My development as a researcher over the past several years has been in large part due to Dr. Vermillion. Thank you for giving me the opportunity as an inexperienced undergraduate to conduct meaningful research. I would also like to thank my committee members, Dr. Peter Tkacik, Dr. Scott Kelly, Dr. Sukumar Kamalasadan, and Dr. Tao Hong for their guidance throughout this process.

All of my lab mates in the Control and Optimization for Renewables and Energy Efficiency (CORE) Lab have been a pleasure to work with. I would like to thank Nihar Deodhar, Alireza Bafandeh, Ali Baheri, Mitchell Cobb, Ben Groelke, Christian Earnhardt, John Borek, Ayaz Siddiqui, James Reed, Josh Daniels, Ben Haydon, and Blake Shepherd for their advice, tough questions, readiness to help with any code/bugs/experiments, and thoughtful feedback during group presentations.

None of my achievements would have been possible without the unwavering support of my family and friends. I would like to thank my wife for her countless encouraging words and unwavering belief in me. Without her, I don’t know how I would have finished this dissertation. I am extremely grateful to my parents who have always been by my side and have shown me the value of dedication and discipline. Finally, I would like to thank God for providing me with the strength needed to finish such an exciting and laborious experience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

For many active system designs, the desire of the designer is to achieve robust and efficient system operation. The robustness and efficiency of the system is directly related to the design decisions made for the physical design and control schemes of the system. However, in many applications, the best decisions are not intuitive, necessitate immense knowledge of the system at hand, and require expertise in many different disciplines. To develop such knowledge of a complex system requires either extensive simulations or an experimental platform, both of which can require large financial and time commitments. This research has developed optimization strategies that leverage combined plant and controller optimization tools that are applicable to plant and controller design of complex, active systems that require either extensive simulations or experiments.

1.1 Background - Combined plant and controller optimization

Traditionally, plant (the set of parameters describing the physical characteristics of the system) and control (the set of parameters used to actively make decisions in real time) design optimizations have been considered in isolation, but in many real-world systems, *coupling* exists between the plant and controller designs. When the plant and controller designs are coupled, the optimal plant design depends on the control design and vice versa. Systems that have been shown to exhibit such coupling are automotive suspension systems ([1], [2], and [3]), elevator systems in [4], advanced powertrain systems in [5], and airborne wind energy systems in [6] and [7]. To address the challenges of coupled systems, combined plant and controller optimization, coined *co-design*, was introduced in [8]. Co-design techniques can be divided into four categories: sequential, iterative, nested, and simultaneous. Sequential co-design techniques consecutively complete a single plant and controller optimization. For a sequential strategy, the plant design is optimized for a given controller design, followed by controller optimization for the optimized plant design. An iterative strategy optimizes the plant design for a single controller design and then optimizes the controller design for a fixed plant design (see [9] and [10]). Alternating between plant and controller optimization is carried out until a prescribed convergence criterion is met. Nested strategies are composed of two loops: an inner loop that completes an entire controller optimization (for a specific plant design) and an outer loop that carries out an iteration of the plant optimization (see [1], [2], and [4]). Lastly, a simultaneous approach combines the plant and controller optimization into a single optimization problem and is solved iteratively until convergence to a solution has been achieved.

In the presence of coupling, achieving a combined optimal solution can be challenging. In [8], it has been shown that convergence to an optimal solution can only be guaranteed for simultaneous and nested approaches (under limiting assumptions). However, variations of sequential and iterative optimization strategies have been developed that leverage the computational advantages of sequential and iterative approaches. The authors in [11] propose the use of control proxy functions to address the issue of unidirectional coupling by decoupling the plant and controller design problems into two separate, but linked (via a control proxy function) optimization problems. A decomposition-based optimization, described in [12], partitions the combined plant and controller optimization problem by using Augmented Lagrangian Coordination (ALC). After partitioning, each subproblem is solved simultaneously, where the plant design optimization is carried out using traditional optimization techniques (e.g., sequential quadratic programming) and the controller optimization is done using optimal control theory. In [1], a nested optimization strategy is used to design an active suspension system, where the inner loop is solved by linear-quadratic regulator and the plant is adjusted in the outer loop using an interior-point method. The previously developed co-design techniques exhibit the following key gaps:

1. The complexity of systems that can be considered by existing co-design tools is limited — often substantially — by restrictions on the underlying optimization tools. For example, LQR-based optimal control design techniques assume a linear system. Plant optimization techniques that use sequential quadratic programming (SQP) require an accurate estimate of the gradient and Hessian of the objective function or Lagrangian for constrained problems. Practical implementation of Pontryagin's minimum principle (PMP)-based techniques requires a relatively simple, closed-form expression for the system dynamics.

2. The vast majority of prior co-design research focuses exclusively on numerical and analytical techniques. Furthermore, the more limited experimental co-design techniques do not leverage the ability to adjust control parameters in real time. In a nested co-design framework, in particular, the ability to adjust controller parameters *during* the simulations/experiments can dramatically reduce the time and cost of the optimization process.

My research focuses specifically on the development of co-design strategies that leverage the unique ability for control parameters to be adapted in real time. This is naturally appealing for a nested framework, wherein plant parameters can only be adjusted between experiments, whereas control parameters can be fully optimized for a given plant design, during an experiment. The framework presented here is classified as a nested co-design strategy, but the techniques used at the outer and inner loop are specifically tailored to meet the gaps discussed above. Table 1 compares the techniques that have been used at each level of the nested strategy in the co-design literature. The work presented in [1], [2], and [4], which performs offline optimization of both the controller and plant, is reliant on an accurate model of the system that must be *linear*. This severely restricts the applicable systems that can employ this technique. More recent work in [13], [14], and [15] has attempted to broaden the types of systems that can utilize nested co-design techniques by realizing the unique capability of control parameters to be optimized online, in *real time*. In [14], Bayesian optimization is used to optimize the controller parameters in real time by breaking the simulation into distinct intervals for periods of settling, evaluation, and computation. By evaluating the performance of control design decisions over a large time step, consideration of the temporal correlation of measurements due to system dynamics is avoided. Although the controller parameters are optimized online in [14], the continuous-time simulation is essentially discretized into a sequence of miniature experiments. My work in [13] and [15] has established nested co-design techniques that achieve controller parameter optimization in *continuous time*.

As noted earlier, a notable deficiency in the majority of co-design literature is its reliance entirely on analytical and numerical models, rather than experiments. Recently, in [16], the authors proposed a simultaneous optimization strategy that blends the results from cheap,

Table 1: Comparison of nested co-design techniques.

Author	Outer loop optimization: single design or batch	Outer loop technique	Inner loop optimization: offline or real-time	Inner loop technique
Fathy (see [1], [2], and [4])	Single	Interior-point method	Offline	LQR/LQG
Deese (see [13])	Batch	Optimal DoE	Real-time, continuous	Extremum seeking
Baheri (see [14])	Single	Bayesian optimization	Real-time, discontinuous	Bayesian optimization
Deese (see [15])	Batch	Optimal DoE	Real-time, continuous	Continuous-time DoE
Deese (recent work, awaiting press)	Batch	Batch Bayesian optimization	Real-time, continuous	RGP-based adaptive control

but less accurate simulations with expensive, but accurate experiments. Within this simultaneous optimization strategy, a batch of candidate system designs are generated by optimal design of experiments (DoE). In general, optimal DoE techniques populate a candidate design space with a set of candidate designs that maximize a statistical information metric. The performance of each candidate system design is evaluated either in a simulation or experiment. Fusion of the simulation and experimental results is used in an attempt to correct simulations to more accurately reflect the true performance of the system at subsequent iterations. However, since the approach in [16] is a simultaneous approach, there is no distinction made between the control and plant parameters. Plant and controller parameters are fundamentally different in the sense that control parameters can be optimized in *real time*, which is accounted for in the proposed framework.

In contrast to the prior literature, this dissertation contributes a nested co-design framework that (i) utilizes statistical techniques that are highly relevant to experimental work, (ii) readily enables real-time adaptation of control parameters during the course of experiments, and (iii) has actually been demonstrated experimentally. A large number of complex systems will benefit from this approach. Designing complex, active systems in practice requires immense multidisciplinary knowledge base, financial expenses, and large time commitments. Complicating matters further, the design parameter tuning rules can be counterintuitive. One such application is the design optimization of active exoskeletons. In [17], the authors detail various challenges associated with designing the controller parameters for optimal performance of active exoskeletons used in gait rehabilitation. Namely, since the physiological needs of each patient varies, proper tuning of the patient-specific parameters requires hours of trial-and-error tuning per patient. Additionally, the design of the physical system (e.g., torque requirements at joints) also depends on the physical characteristics of each patient [18]. Applying nested co-design with real-time adaptation of control parameters would reduce the time required to tune the active exoskeleton for each patient's unique needs. Another system that could benefit from the application of co-design techniques is the design optimization of diesel engine timing. The authors in [19] and [20] present a number of critical parameters that contribute to the overall performance of the engine, such as the mass air flow, rail pressure, drain orifice diameter, inlet orifice diameter, and solenoid valve seat angle. Optimization of these parameters can be challenging because of the increased regulation on emissions and a trade-off between fuel consumption and emissions. By applying nested co-design, the time required to optimize the diesel engine for ever-changing regulations would be significantly reduced.

1.2 Proposed application: Airborne wind energy (AWE) systems

This research will focus on airborne wind energy (AWE) systems for the validation of nested co-design tools with real-time controller adaptation. AWE systems replace the tower of traditional wind turbines with tethers and a lifting body that enable the turbine to reach high altitudes, where winds are typically stronger and more consistent. The material costs associated with the foundation and tower of traditional wind turbines are significantly reduced by introducing tethers. The tethers enable the operating height of the turbine to be adjusted during operation [21, 22, 23]. Additionally, the tethers can be used to induce crosswind motion of the system in order to augment power generation [24, 25, 26, 27, 28, 29]. Currently, a number of companies, specifically Makani [30], Altaeros [31], KiteGen [32], Ampyx Power [33], and KiteMill [34], are exploring the development of deployable AWE systems. Figure 1 displays the designs of each of the aforementioned companies. Energy generation of AWE systems can occur onboard the lifting body or at the ground station. Two examples of onboard generation systems are the Makani and Altaeros systems. Altaeros' Buoyant Airborne Turbine (BAT), a three tether stationary flight system, houses an onboard turbine in an annular shroud and transmits power to the ground station via a single conductive tether. Makani's system accommodates onboard turbines on a rigid wing structure equipped with control surfaces; however this system moves in crosswind motion to augment power generation. Through tether tension induced by crosswind motion, Ampyx Power's rigid wing system generates power at the ground station. Lastly, KiteGen uses a kite-like lifting body moving in a crosswind motion to generate power at the ground station via tether tensions.

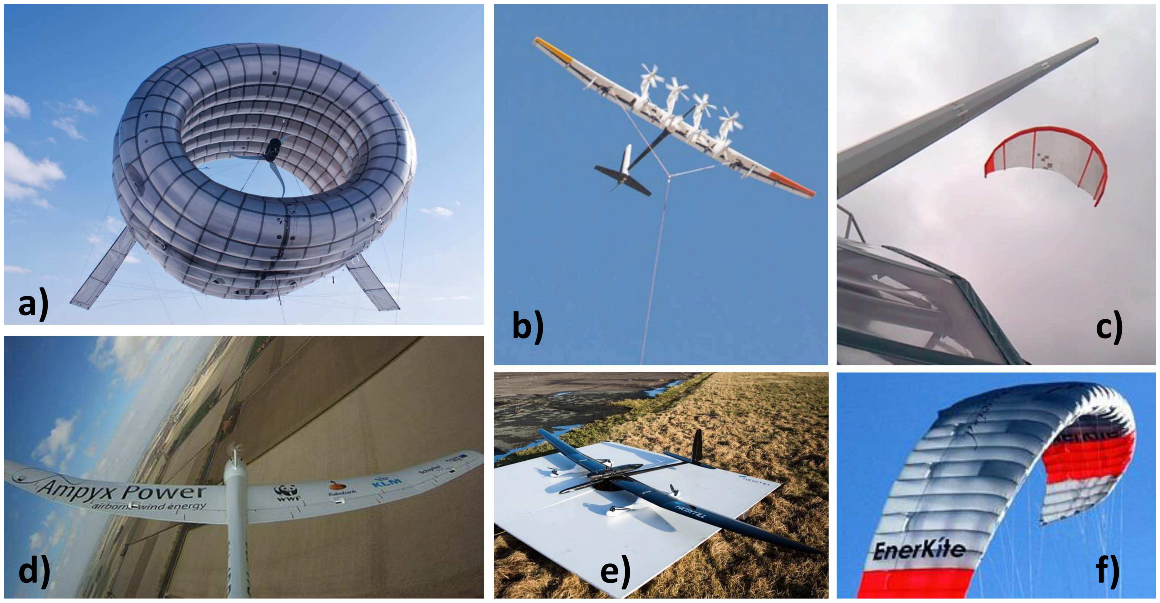

Figure 1: Collection of AWE system designs from a) Altaeros Energies [31], b) Makani [30], c) Kitegen [32], d) Ampyx Power [33], e) KiteMill [34], and f) EnerKite. [35]

The AWE community has placed extensive emphasis on control system design, with much less attention paid to plant design. In particular, significant emphasis has been placed on altitude optimization and crosswind flight trajectory optimization. Altitude optimization, discussed in [21] and [22], attempts to optimize the operating altitude of the turbine in the presence of a spatiotemporally varying environment. In [24], [25], [26], [27], [28], and [29] power augmentation of AWE systems, realized by moving the system in controlled crosswind flight patterns, is studied. Much of the AWE community has not considered the coupling between plant and controller designs, even though coupling has been shown to exist in [36, 37].

AWE systems are ideal candidate systems for the development of advanced co-design techniques for five main reasons. First, the AWE system dynamics are complex and highly nonlinear. Secondly, AWE system dynamics have not been fully characterized, and many control-oriented dynamic models neglect key higher-order phenomena, such as unsteady flow effects, fluid/structure interaction, and effects of the spatially varying flow field in the vicinity of the lifting body. Thirdly, the plant and controller design are *coupled*, which means that the optimal plant design depends on the controller design and vice versa. Fourth, complex system dynamics result in objective functions that cannot (for all practical purposes) be expressed in an analytical form in terms of design parameters. Most importantly from the perspective of optimization, gathering first or second order information from the objective function (or performance index) can be inaccurate in simulation and time-consuming for experimental studies. Lastly, at the water channel in the Motorsports Research Lab at UNC-Charlotte, a unique lab-scale testing platform was established in [6], which provides a mechanism to evaluate AWE system designs and control systems experimentally. Therefore, the co-design strategies developed in this proposal can be validated in simulation and experimentally. Recently, [38] presented a experimentally-infused, simultaneous optimization strategy that addresses the coupling between the plant and controller designs. Nevertheless, the application of co-design to AWE systems remains in a nascent state.

1.3 Problem statement and intended contributions

The goal of this co-design framework is to determine the plant and controller parameter vectors, denoted by $\mathbf{p}_p$ and $\mathbf{p}_c$, respectively, that minimize an integral performance index, subject to a dynamic model and bounds on the parameters. Mathematically, the optimization problem can be expressed as follows:

$$\begin{aligned}
&\underset{\mathbf{p}_p,\mathbf{p}_c}{\text{minimize}} && J(\mathbf{p}_c,\mathbf{p}_p)=\int_0^{t_f} J_{inst}(\mathbf{x}(t),\mathbf{u}(t);\mathbf{p}_p,\mathbf{p}_c)dt \\
&\text{subject to:} && \mathbf{p}_p\in P,\ \mathbf{p}_c\in C \\
&\text{given:} && \dot{\mathbf{x}}=f(\mathbf{x},\mathbf{u};\mathbf{p}_p,\mathbf{p}_c) \\
& && \mathbf{u}=g(\mathbf{x};\mathbf{p}_c)
\end{aligned} \tag{1}$$

Here, J_{inst} is a instantaneous performance index that captures the performance characteristics of the system, $\mathbf{x}(t)$ are the states that describe the system's dynamics, and $\mathbf{u}(t)$ are the control inputs to the system. The control inputs to the system are a function of the states because the adaptation used to optimize the control parameters is a function of the system's states.

In legacy co-design literature, severe restrictions have been placed on the system dynamics. The overarching objective of this research is to extend the legacy co-design methodologies to be applicable to real-world systems that require either extensive simulations or experiments. In light of this overarching goal, the following research questions can be posed:

1. How can plant and control design points be selected in order to provide the most information in each respective design space?

2. Can continuous-time adjustment of control parameters be leveraged within a nested optimization strategy that does not require knowledge of the plant or first or second order information of the objective function?

3. Can traditional DoE techniques (which generally are used to generate a discrete set

of design points at discrete iterations) be continualized in a way that enables those techniques to take the form of a real-time adaptive control law within the inner loop of a nested strategy?

A general process diagram for the proposed nested framework can be seen in Figure 2. To begin the process, a *batch* of candidate plant designs is generated using iterative experimental design techniques. Each of the candidate plant designs is then tested in a simulation or experiment, where an adaptive control law is used to optimize the control parameter(s) in *real time*. After converging to the optimal control design, an integral objective function value is calculated. A response surface characterization is carried out, which yields two important sources of information used for design space reduction. The first is the approximate performance value at off design points. The second is an uncertainty characterization that approximates the uncertainty in the approximate performance values across the design space. Using these two mathematical quantities, the plant design space is reduced. Within the reduced design space, another batch of candidate plant designs is generated. This process is carried out iteratively until the plant design space has been sufficiently reduced.

Due to the limitations of the techniques in the adaptive control literature for performing non-parametric adaptive control in real time, using statistical, information-based techniques, significant focus has been placed on the inner loop controller optimization. Overall, three adaptive control laws have been utilized in this work; two of which are standalone contributions of this research. The first, extremum seeking (ES) ([39]), is a popular non-model based adaptive control technique that relies on an oscillatory perturbation to move through the design space in order to search for a optimizer. The second adaptive control

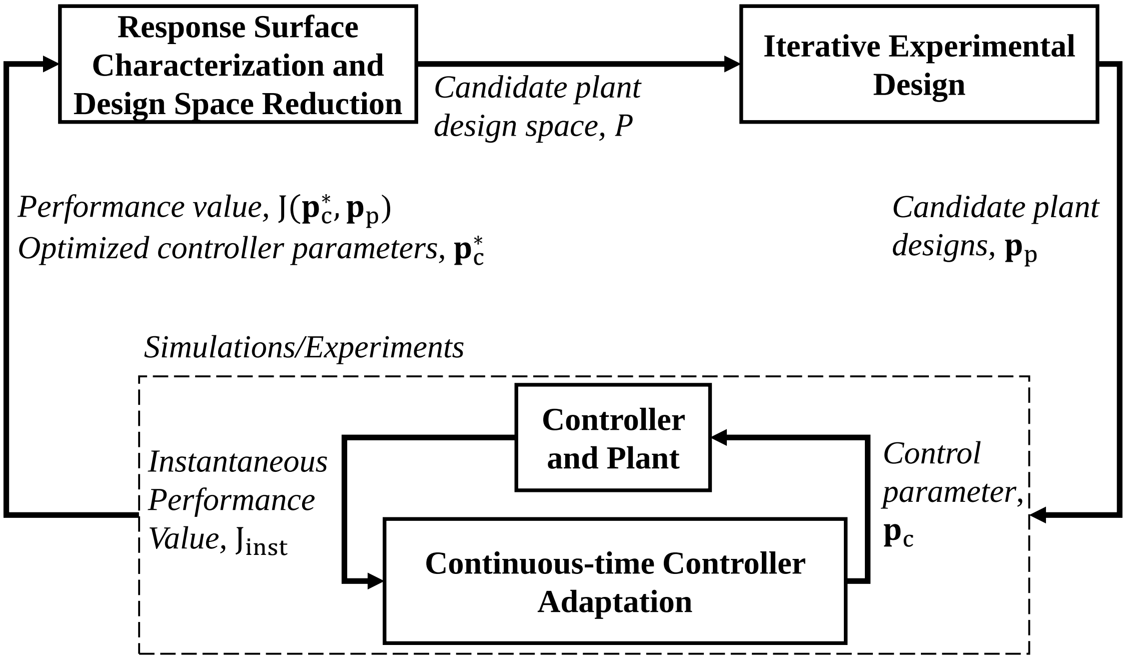
Response Surface Characterization and Design Space Reduction
Candidate plant design space, P
Iterative Experimental Design
Performance value, $J(\mathbf{p}_c^*, \mathbf{p}_p)$
Optimized controller parameters, $\mathbf{p}_c^*$
Candidate plant designs, $\mathbf{p}_p$
Simulations/Experiments
Controller and Plant
Instantaneous Performance Value, J_{inst}
Control parameter, $\mathbf{p}_c$
Continuous-time Controller Adaptation

Figure 2: General process diagram of nested optimization strategy developed in this research.

technique is an information-based algorithm inspired by traditional design of experiments approaches, which we have coined *entropy-based* DoE. Continuous-time design space exploration and reduction is achieved by utilizing an information metric, based on the concept of information entropy [40]. Specifically, design space exploration is accomplished by maximizing a normalized information metric, while design space reduction is executed by using the normalized information metric as an approximation of the response surface uncertainty. The third, coined recursive Gaussian process (RGP)-based adaptive control, formalizes the concepts from the entropy-based DoE approach by grounding them in a statistical framework.

The overarching focus of this research was to develop a nested co-design framework that is applicable to complex system design and leverages the ability to optimize controller parameters in real-time. Specific contributions are as follows:

1. Establishment of a *real-time* controller adaptation law that leverages RGP modeling to recursively update predictive mean and variance in the presence of temporal correlation that results from system dynamics, which is subsequently used to guide global exploration of the design space;

2. Experimental validation of the RGP-based adaptation from 1;

3. Development of a nested co-design strategy that leverages GP modeling at the inner and outer loops;

4. Experimental validation of the GP-driven nested optimization framework described in 3.

CHAPTER 2: AIRBORNE WIND ENERGY SYSTEM: DYNAMIC MODEL, PERFORMANCE CONSIDERATIONS, AND EXPERIMENTAL SETUP

The optimization strategies developed in this work are verified using the Buoyant Airborne Turbine (BAT) of Altaeros as a case study. The BAT replaces the tower of traditional wind turbines with a Helium-filled annular shroud and three tethers. Each tether is attached to winches, installed on a ground station, that enable the adjustment of position and orientation of the system. Housed within the annular shroud, a horizontal axis turbine is used to extract energy from the wind. Energy is transmitted to the ground station by a single conductive tether. In Figure 3, the BAT system can be seen during full-scale flight testing at Loring Air Force Base (decommissioned).

2.1 Dynamic model of the BAT

The simplified BAT dynamic model discussed in [41] is used for the simulation studies in this work. Figure 4 illustrates the ground-fixed, body-fixed, and generalized coordinate systems of the BAT dynamic model. To allow the dynamics to be modeled entirely through ordinary differential equations (ODEs) (avoiding the need for a differential algebraic equation model), the three tethers are approximated as a single kinematic link with length, L_t. At the shroud, the tether has a bridle joint with two controlled degrees of freedom. From the resulting set of ODEs, the BAT position and orientation can be described by six generalized coordinates: Θ, Φ, Ψ, L_t, θ' and ϕ'. The orientation of the BAT tether can be described by the first three of the generalized coordinates. Zenith angle (Φ), the angle between the tether

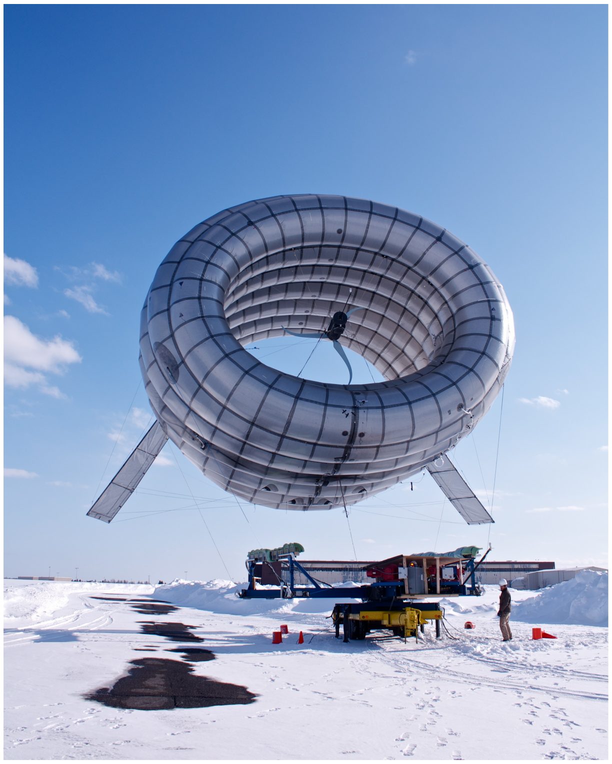

Figure 3: Full-scale prototype of Buoyant Airborne Turbine (BAT) during 2012 flight testing at Loring Air Force Base. [31]

and ground-fixed vertical axis, and azimuth angle (Θ), the angle of the tether projection on the horizontal plane, describe the orientation of the tether with respect to the ground-fixed coordinate system. Twist angle, Ψ, represents the orientation of the BAT shroud with respect to the tether axis. The length of the single tether is described by L_t. The induced roll (ϕ') and pitch (θ') angles are controlled angles induced from tether length differences between the starboard and port tethers and forward and aft tethers, respectively. These controlled coordinates can be described in terms of the individual lengths of each tether in the three tether model by the following equations:

$$\phi' = tan^{-1}(\frac{l_3 - l_2}{l_{sep}^{lat}}), \quad (2)$$

$$\theta' = tan^{-1}(\frac{l_1 - 0.5(l_2 + l_3)}{l_{sep}^{long}}), \quad (3)$$

$$L_t = \frac{1}{3}(l_1 + l_2 + l_3), \quad (4)$$

where l_{sep}^{long} and l_{sep}^{lat} are longitudinal and lateral tether attachment separation distances, respectively. The control inputs are the tether release speeds, $\bar{u}_i$, given by:

$$\bar{u}_i = \frac{d}{dt} l_i, \ \ i = 1,2,3. \quad (5)$$

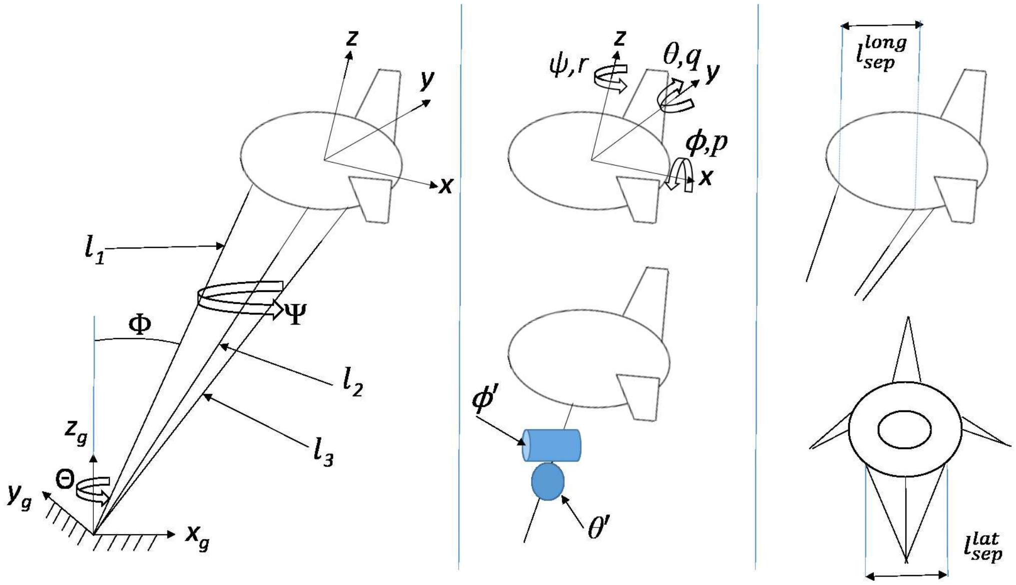

Figure 4: Axis system for the BAT dynamic model. This diagram illustrates generalized coordinates (left and bottom middle), Euler angles (top middle), and tether attachment points (right).

Derived using an Euler-Lagrange formulation, the system dynamics are given by:

$$\begin{aligned} D(Q)\ddot{Q}+C(Q,\dot{Q})\dot{Q}+g(Q) &= \tau(Q,\dot{Q},V,\psi_{wind}), && (6)\\ X &= f(Q,\dot{Q}), && (7)\\ \Omega &= g(Q,\dot{Q}), && (8) \end{aligned}$$

where:

$$\begin{aligned} Q &= [\Theta \;\; \Phi \;\; \Psi \;\; L_t \;\; \theta' \;\; \phi'], && (9)\\ X &= [x \;\; y \;\; z \;\; u \;\; v \;\; w], && (10)\\ \Omega &= [\phi \;\; \theta \;\; \psi \;\; p \;\; q \;\; r]. && (11) \end{aligned}$$

Here, τ represents a vector of generalized forces, V is the wind speed, and ψ_{wind} is the wind direction. The angle of attack (α) and sideslip angle (β) describe the orientation of the apparent wind vector with respect to the body-fixed coordinates of the BAT. Aerodynamic forces and moments are functions of both α and β.

The surface areas of the BAT's horizontal and vertical stabilizers represent important design parameters. However, prior to this work, the numerical model was a lumped aerodynamic model, meaning that aerodynamic coefficients were characterized for the combined shroud (fuselage) and stabilizers. In order to consider the impact the stabilizer geometry has on the performance of the system, the aerodynamics of the system were *partitioned* between the main body, horizontal stabilizers, and vertical stabilizers. Total aerodynamic coefficients are given by:

$$
\begin{aligned}
C^{\text{total}}_{D,L,S}(\alpha,\beta) &= C^{\text{body}}_{D,L,S}(\alpha,\beta)+C^{V}_{D,L,S}(\alpha,\beta)\frac{A_V}{A_{\text{body}}} \\
&\quad +C^{H}_{D,L,S}(\alpha,\beta)\frac{A_H}{A_{\text{body}}},
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
C^{\text{total}}_{Mx,My,Mz}(\alpha,\beta) &= C^{\text{body}}_{Mx,My,Mz}(\alpha,\beta) \\
&\quad +C^{V}_{Mx,My,Mz}(\alpha,\beta)\frac{A_V l_V}{A_{\text{body}} l_{\text{body}}} \\
&\quad +C^{H}_{Mx,My,Mz}(\alpha,\beta)\frac{A_H l_H}{A_{\text{body}} l_{\text{body}}}.
\end{aligned}
\tag{13}
$$

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 the BAT, respectively. The subscripts *body*, V, and H correspond to the main body, vertical stabilizer, and horizontal stabilizer, respectively.

2.2 Experimental platform for AWE systems

At UNC-Charlotte a unique experimental platform, first detailed in [6], provides a rapidly reconfigurable platform for assessing the characteristics of AWE systems while operating under closed-loop control with various environmental perturbations. The lab-scale experimental platform facilitates the ability to rapidly iterate through various design configurations at a fraction of the cost of full-scale flight testing. This work utilizes the UNC-Charlotte water channel, shown in Figure 5, which is the 5^{th} largest in the United States. The UNC-Charlotte water channel has a cross section of $1.0\ m \times 1.0\ m$ and is capable of flow speeds up to $1.0\frac{m}{s}$.

Scaled models of AWE systems are 3D printed and “flown” in the water channel en-

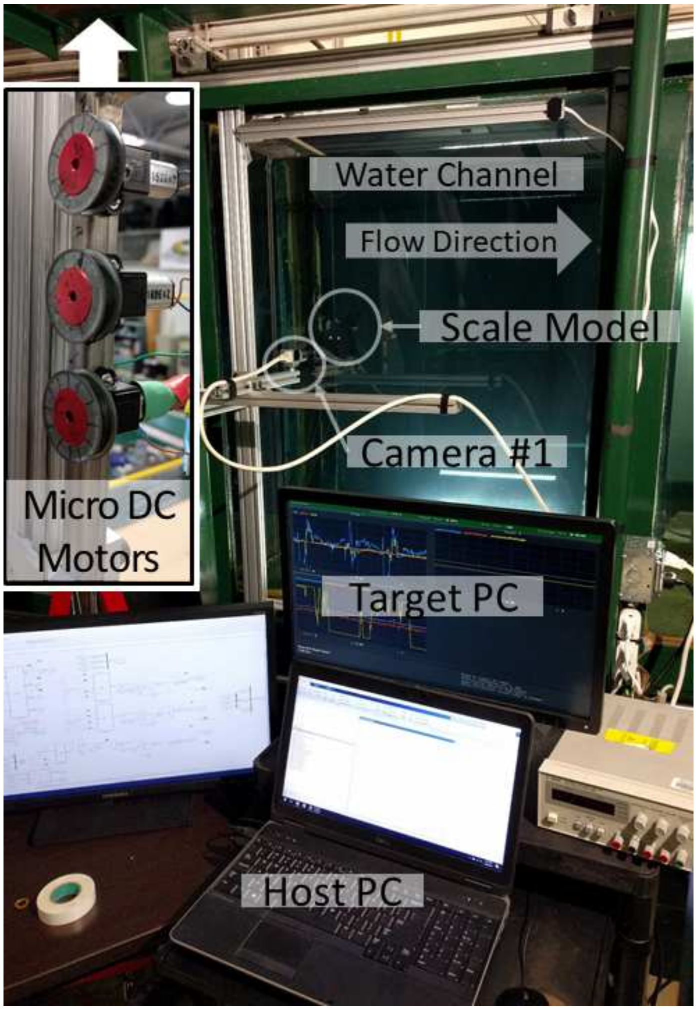

Figure 5: Image of UNC-Charlotte water channel experimental platform highlighting the components.

vironment. Tracking of the scaled model is achieved by three high speed cameras used for image acquisition and an image processing algorithm detailed in [29]. One camera is mounted on the side of the water channel, perpendicular to the side of the water channel. Two cameras are placed on the bottom of the water channel, one of which is perpendicular to the bottom of the water channel, the other of which is placed at a 45 degree angle with respect to the bottom of the water channel. A schematic of the camera layout can be seen in Figure 6. The experimental platform utilizes three DC motors, mounted on the testing frame at the top of the water channel, to control the lengths of each of the three tethers on the scaled system. A target computer is used to handle the computational burden of the image processing, geometric transformations that are used to calculate the position and orientation of the BAT, and execution of the control algorithm. The target computer, equipped with 6 cores and 64 GB of memory, carries out all calculations at a sample time of 0.025 seconds. The host computer is used for code development and provides a graphical user interface (GUI) to execute real-time adjustments to the controller during each experiment.

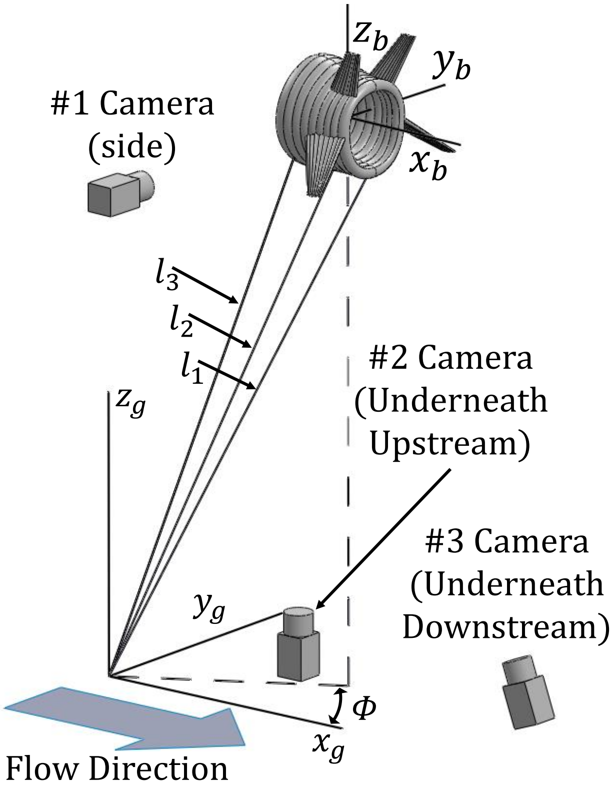

Figure 6: Schematic of water channel configuration.

In general, there are a number of candidate plant design parameters that could be considered in this co-design framework, such as tether attachment locations, stabilizer reference areas, longitudinal location of center of mass, stabilizer angle of incidence, and stabilizer dihedral angle. Figure 7 illustrates the current features of the scaled models that can be rapidly readjusted. The ballast holes allow for rapid variation of the center of mass location of the BAT, while the multiple tether attachment locations allow the model to be tethered in several configurations. Stabilizer slots provide a mechanism for rapidly adjusting the features associated with the stabilizers, such as the stabilizer reference areas, stabilizer angle of incidence, etc. The plant design parameters focused on in this dissertation are the stabilizer reference area scale factor (K_A), the longitudinal location of the center of mass (x_{cm}), and net buoyancy (ρ_β). The reference area scale factor uniformly scales the horizontal (A_H) and vertical (A_V) stabilizer reference areas.

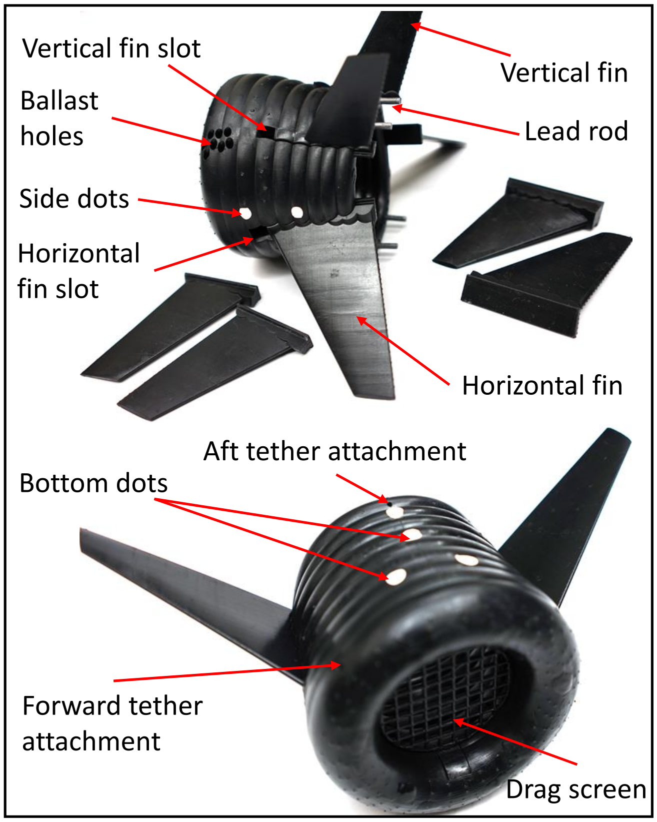

Figure 7: Image of 3D printed scaled model of BAT illustrating the rapid reconfigurable features.

2.3 Control design structure for the BAT

To achieve autonomous flight, AWE systems are equipped with lower-level flight controllers that adjust control inputs to the system in order to track desired setpoints. For the BAT, the tether release speeds are actuated to track altitude, roll angle, and pitch angle setpoints. The operation of the BAT during stationary and crosswind flight were used as case studies throughout this research. Because desired flight objectives for each operational mode are different, the controller varies to meet the desired flight characteristics of each operational mode. During stationary flight, secondary tasks, such as telecommunications, surveillance, etc., are executed while generating power. These secondary tasks during flight require the BAT to remain sufficiently stationary. Crosswind flight, on the other hand, is used to maximize the power production by increasing the apparent wind velocity presented to the on-board turbine(s) through crosswind motions [25, 26, 27, 28, 29].

2.3.1 Attitude/Altitude Controller

Regardless of the operational mode of the BAT, the attitude/altitude controller remains the same. The attitude/altitude controller structure for BAT can be seen in Figure 8. For stationary flight θ_{sp}, ϕ_{sp}, and z_{sp} are all considered to be constant setpoints.

Based on the tracking error, the flight controller uses three lead filters to compute preliminary control inputs that describe the average tether release speed, the forward/aft tether speed difference, and the port/starboard speed difference, denoted by $\bar{u}_z$, $\bar{u}_\theta$, and $\bar{u}_\phi$, respectively:

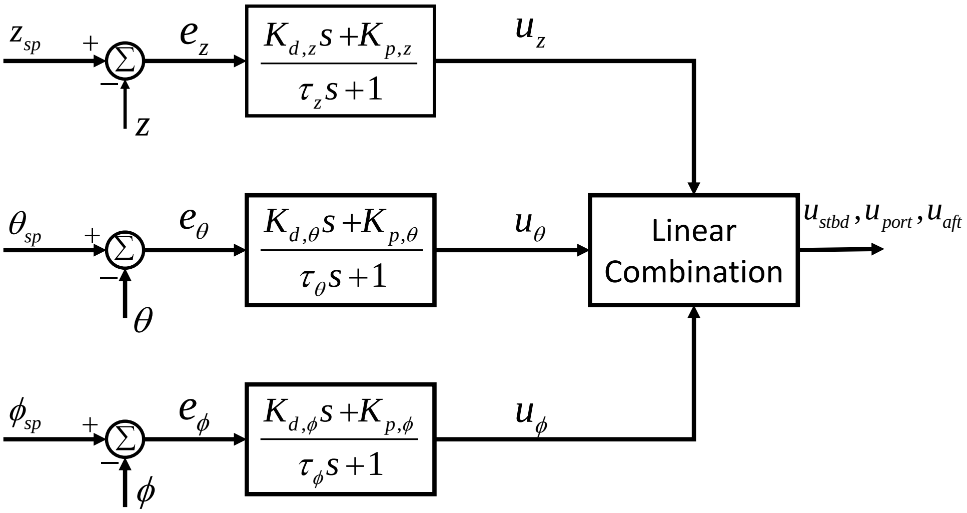
z_{sp}
+
−
z
e_z
$\frac{K_{d,z}s+K_{p,z}}{\tau_z s+1}$
u_z
θ_{sp}
+
−
θ
e_θ
$\frac{K_{d,\theta}s+K_{p,\theta}}{\tau_\theta s+1}$
u_θ
Linear Combination
$u_{stbd}, u_{port}, u_{aft}$
ϕ_{sp}
+
−
ϕ
e_ϕ
$\frac{K_{d,\phi}s+K_{p,\phi}}{\tau_\phi s+1}$
u_ϕ

Figure 8: Block diagram of attitude/altitude controller for the BAT.

$$\begin{aligned}
\bar{u}_z(s) &= \frac{k_{d,z}s+k_{p,z}}{\tau_z s+1}(z_{sp}(s)-z(s)), && (14)\\
\bar{u}_\theta(s) &= \frac{k_{d,\theta}s+k_{p,\theta}}{\tau_\theta s+1}(\theta_{sp}(s)-\theta(s)), && (15)\\
\bar{u}_\phi(s) &= \frac{k_{d,\phi}s+k_{p,\phi}}{\tau_\phi s+1}(\phi_{sp}(s)-\phi(s)). && (16)
\end{aligned}$$

The commanded tether release speeds ($\bar{u}_{center}$, $\bar{u}_{stbd}$, and $\bar{u}_{port}$) are calculated as linear combinations of the preliminary control inputs shown in (14), (15), and (16). These linear combinations are given by:

$$\begin{aligned}
\bar{u}_{center} &= \bar{u}_z-\bar{u}_\theta, && (17)\\
\bar{u}_{stbd} &= \bar{u}_z+\bar{u}_\theta+\bar{u}_\phi, && (18)\\
\bar{u}_{port} &= \bar{u}_z+\bar{u}_\theta-\bar{u}_\phi. && (19)
\end{aligned}$$

2.3.2 Crosswind controller augmentation

Crosswind motion is achieved in both simulation and experimentally by commanding a nonzero roll angle to induce sideslip. The selection process for the roll angle setpoint varies between simulation and experiment. Additionally, a rudder is used to independently adjust the sideslip angle in simulation.

2.3.2.1 Design considerations for crosswind flight controller in simulation

The controller structure used to induce crosswind flight for the dynamic model discussed in Section 2.1 is shown in Figure 9. This controller structure is detailed in [21]. By engaging the motion setpoint and crosswind controllers within the hierarchical structure in Figure 9, a nonzero roll angle setpoint is commanded to the lower-level roll controller. Lateral translation of the AWE system is caused by the introduction of sideslip. The motion setpoint block is used to determine the target crosswind velocity v_{sp}^{cw} and periodic timing of the system's tack. Periodic reversal of the system's tack is used to maintain the desired crosswind motion. The target crosswind velocity is limited by the rated power of the onboard turbines because operating above the rated power of the wind turbine is not beneficial. To increase the efficiency of the crosswind motion, a rudder is added to the vertical stabilizer to induce yaw, in order to maintain a sideslip angle that keeps apparent wind on the correct side of the BAT.

2.3.2.2 Experimental crosswind flight controller

Experimental crosswind flight at the UNC-Charlotte water channel was first implemented in [29]. Just as in the attitude/altitude controller structure for stationary flight, shown in

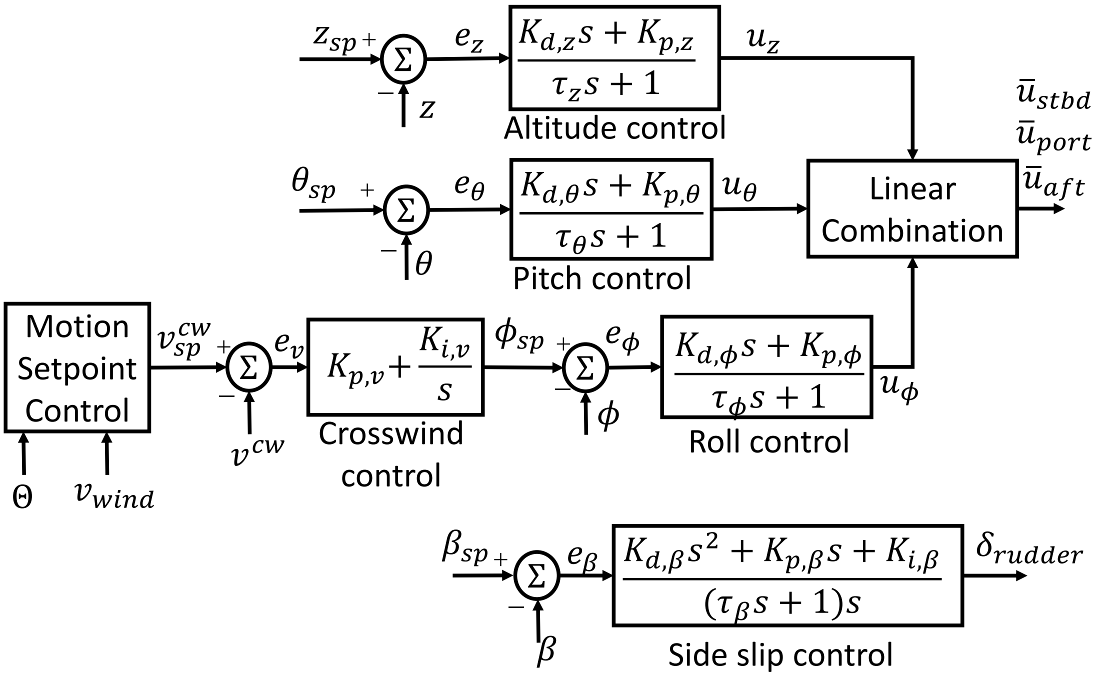

Figure 9: Crosswind flight controller structure for numerical simulation of BAT. Setting β_{sp} and ϕ_{sp} to zero and turning off the crosswind motion controller results in stationary flight.

Figure 8, three lead filters are used to track roll, pitch, and altitude. The pitch angle (θ_{sp}) and altitude (z_{sp}) setpoints are constants. However, to induce lateral motion, the roll angle setpoint is a square wave, which periodically redirects components of the lift vector in the lateral direction in order to induce the desired motion. The roll angle setpoint is given by:

$$\phi_{sp} = \begin{cases} a_{sq}(t), & \int_{t_i}^{t} \frac{1}{T_{sq}(\tau)} d\tau \leq 0.5 \\ -a_{sq}(t), & 0.5 < \int_{t_i}^{t} \frac{1}{T_{sq}(\tau)} d\tau \leq 1 \end{cases} \tag{20}$$

Here, a_{sq} denotes the amplitude of the square wave, T_{sq} denotes the period of the square wave, and t_i denotes the time at which the present square wave started. Once the integral the reaches a value of 1, it is reset to zero.

2.4 Dynamic Similarity Between Lab-Scale and Full-Scale

One hallmark of the lab-scale platform is its ability to replicate the full-scale flight dynamics of AWE systems at a fraction of the cost. Dimensional analysis, using the Buckingham Pi Theorem, was performed on the plant design of an AWE system in [36] to determine dimensionless groups that must be conserved across multiple scales to ensure dynamic similarity. These dimensionless groups give rise to scaling laws that if followed result in lab-scale flight that replicates full-scale with the exception of uniformly scaled time constants. The scaling rules are as follows:

1. To achieve geometric similarity between full-scale and lab-scale, the individual tether lengths must be scaled by the same scale factor as the model length.

2. Net buoyancy must be preserved between full-scale and lab-scale.

3. To conserve Froude number (the fifth dimensionless group in [36]), the lab-scale flow velocity v_{lab} must be scaled by the square root of the length-scale factor. For example, if L is the length scale factor and v_{full} is the full-scale target wind velocity, then $v_{lab} = \sqrt{L} v_{full}$

The aforementioned scaling laws are only valid if two assumptions are met. The first assumption states that the mass distribution between lab-scale and full-scale are identical, which is fulfilled by properly designing the mass properties of the 3D printed model. The second assumption requires that fluid coefficients are preserved, in spite of Reynolds number differences. This has been validated for the BAT via force and moment testing in [36].

In order to extrapolate closed-loop performance to full-scale, not only must the plant parameters be properly scaled, but additionally the controller parameters must be properly scaled. In [29], the Buckingham-Pi analysis from [36] was extended to the controller parameters to determine the dimensionless groups for the controller that must be conserved to achieve dynamic similarity. The controller parameters must be scaled according to the following scaling laws to ensure closed-loop dynamic similarity between full-scale and lab-scale flight:

1. The derivative gains of the roll and pitch controller at lab-scale are proportional to the length scale factor, L, i.e. $k_{d,\theta}, k_{d,\phi} \propto L$.

2. The lab-scale proportional gain of the roll, proportional gain of the pitch control, and tether release speed saturation (v_l^{sat}) are proportional to the square root of the length scale factor, i.e. $k_{p,\theta}, k_{p,\phi} \propto \sqrt{L}$ and $v_l^{sat} \propto \sqrt{L}$.

3. The lab-scale proportional gain of the altitude controller is proportional to $\frac{1}{L}$, i.e. $k_{p,z} \propto \frac{1}{L}$.

4. The lab-scale filter time constants and roll setpoint period must be scaled by L, i.e. $\tau_{\phi,\theta,z} \propto L$ and $T_{sq} \propto L$.

If the plant and controller scaling laws are satisfied, the lab-scale, closed-loop dynamics will replicate the closed-loop dynamics during full-scale flight with uniformly scaled time constants. In particular, the time constants will scale according to the square root of the length scale factor, i.e. $\tau \propto \sqrt{L}$. It is important to note that the work of this dissertation is conducted at lab-scale because the overall goal is to implement the co-design methodologies developed here experimentally.

2.5 Performance indices

One of the main ingredients to any optimization problem is a performance index (or objective function), which reflects features that are important to overall system performance. The Altaeros BAT is a unique AWE system that can achieve both stationary and crosswind flight. The optimization strategies developed in this work explore both. The performance index associated with each mode of operation differ since the goal of each mode is different.

2.5.1 Stationary flight performance index

One economic use case for the BAT is the dual-use case of simultaneous energy harvesting and telecommunications. This dual use case requires substantially stationary application. These qualitative requirements necessitate a control system whereby the BAT remains sufficiently motionless in variable wind conditions. Furthermore, if multiple BATs are implemented in a farm, it is beneficial that the "ground footprint" of each system (defined as the projected area over which the turbines can float) be as small as possible, so as to maximize the number of BATs in a tight farm. The ground footprint can be considered by incorporating the zenith angle (shown as Φ in Figure 4) as a term within the performance index. The quality of flight (i.e., the extent to which the BAT remains stationary) is characterized by penalizing roll angle tracking error ($\phi_e = \phi_{des} - \phi$) and yaw angle offset ($\psi_e = \psi_{flow} - \psi$). In this specific application, ϕ_{des} and ψ_{flow} are both equal to zero. The instantaneous performance index, used to drive the continuous-time control parameter adaptation, is given by:

$$J_{inst}(\mathbf{p}_c, \mathbf{p}_p) = k_1 \Phi^2 + k_2 \psi_e^2 + k_3 \phi_e^2. \tag{21}$$

To evaluate the performance of each candidate plant design, while operating at a fixed controller, an integral performance index value is calculated. The integral performance index value is calculated by:

$$J(\mathbf{p}_c^*(\mathbf{p}_p), \mathbf{p}_p) = \int_{t_c}^{t_c+\Delta t} (k_1 \Phi^2 + k_2 \psi_e^2 + k_3 \phi_e^2) dt, \tag{22}$$

where t_c is the time at which the inner loop optimization converged to the optimal controller design and Δt represents the time interval over which the integral performance index is evaluated over.

2.5.2 Crosswind flight performance index

The objective of an AWE system during crosswind operation is to maximize lap-averaged power production. This is realized mathematically by selecting the instantaneous performance index as the ratio between the power that could be generated through crosswind flight and the power that would be generated under stationary flight, assuming the same onboard turbine and corresponding power coefficient. This ratio, termed the *power factor*, is given by

$$J_{inst}(\mathbf{p}_c, \mathbf{p}_p) = \left(\frac{|v_{app}|}{|v_w|} \right)^3. \tag{23}$$

Here, v_{app} is the apparent wind velocity and v_w is the free-stream wind velocity. After the controller optimization has converged, an average power factor, while operating at a fixed controller, is calculated. The average power factor is calculated by

$$J(\mathbf{p}_c^*(\mathbf{p}_p),\mathbf{p}_p) = \frac{1}{\Delta t}\int_{t_c}^{t_c+\Delta t}\left(\frac{|v_{app}|}{|v_w|}\right)^3 dt \tag{24}$$

Here, t_c is the time at which the inner loop optimization converged to the optimal controller design and Δt represents the time interval over which the integral performance index is evaluated over.

2.6 Environmental perturbation during stationary flight

To evaluate the dynamic performance of the BAT during stationary flight, the system must be perturbed. This is because the water channel, unperturbed, only supplies unidirectional flow. Thus, a symmetric BAT starting in a direct downwind configuration will not experience any excitement of its lateral modes in the absence of a perturbation. Perturbations can be achieved by initializing the system at a point that is not an equilibrium or by applying an external excitation. In selecting an appropriate perturbation for the AWE application, there were two criteria considered. First, since online adaptive control techniques will be used to optimize the controller parameters in real time, the perturbation must be *consistent* throughout the simulation. A consistent perturbation ensures that the instantaneous performance value is only dependent on the controller parameters, instead of characteristics of the perturbation at a particular time within the simulation. Second, considering that experimental implementation of the proposed design optimization strategies is the focus of future work, a perturbation that can be implemented in a lab-scale experimental setup is desirable. In [6], a lab-scale experimental platform was developed to perform flight performance evaluations of AWE systems (discussed in Chapter 2.2). To provide a consistent excitation using the lab-scale platform in [6], a frequency approximation of vortex shedding off of a cylinder was selected. The Strouhal number, a dimensionless parameter used to characterize the frequency of vortex shedding in the wake of a cylinder, was used to generate velocity profile presented to the BAT. The Strouhal number is given by:

$$St = \frac{fL}{U} = 0.198(1 - \frac{19.7}{Re_D}), \tag{25}$$

where f is the vortex shedding frequency, L is the characteristic length of the cylinder, U is the flow velocity, and Re_D is the Reynolds number associated with L and U. The empirical formula shown in equation (25) can be used to solve for the vortex shedding frequency, f, at a given Reynolds number. It is important to note that this empirical formula is only valid for Reynolds numbers in the range $250 < Re_D < 2 \cdot 10^5$, which is applicable to the lab-scale platform to be used in future experiments. In simulation, the velocity components in each direction are functions of the oscillation frequency f, solved for using equation (25). Each of the velocity components is given by:

$$\begin{aligned} v_x &= v_x^{base} + v_{x0} \sin(\omega_{dist} t + \frac{\pi}{2}), & (26) \\ v_y &= v_{y0} \sin(\omega_{dist} t), & (27) \\ v_z &= v_{z0} \sin(\omega_{dist} t + \frac{\pi}{2}), & (28) \end{aligned}$$

where $v_x^{base} = 0.606\frac{m}{s}$, $v_{x0} = 0.0866\frac{m}{s}$, $v_{y0} = 0.0650\frac{m}{s}$, $v_{z0} = 0.00866\frac{m}{s}$, and $\omega_{dist} = 2\pi\frac{rad}{s}$. These velocities correspond to flow velocities in the *lab-scale* platform described in [36]. Through the dimensional analysis discussed in [36], these lab-scale flow velocities are equivalent to wind speeds of $v_x^{base} = 6.06\frac{m}{s}$, $v_{x0} = 0.866\frac{m}{s}$, $v_{y0} = 0.650\frac{m}{s}$, $v_{z0} = 0.0866\frac{m}{s}$ (i.e., a scale factor of 10) and an oscillation frequency of $\omega_{dist} = 0.2\pi\frac{rad}{s}$ (i.e., a scale factor of 1/10) on the full-scale BAT, which has 100 times the characteristic length of the models used in the lab-scale setup.

CHAPTER 3: PLANT-LEVEL ITERATION TOOLS

In the outer loop of this optimization strategy, the goal is to generate a batch of candidate plant design points that populates a candidate design space in a fashion that yields the maximum amount of information. While the details vary, the structure of the outer loop remains the same throughout this work. That general structure takes the following form:

- A batch of candidate plant designs is generated using an information-seeking strategy.

- Following the inner-loop controller optimization, which results in an optimal control parameter vector and performance value for each plant design, a response surface is used to characterize performance as a function of the design variables across the design space.

- Using an uncertainty characterization and the response surface characterization, the design space reduction rejects portions of the design space that are determined to be suboptimal with a statistical level of confidence.

This structure is repeated until the design space has been significantly reduced. The remainder of this chapter provides a detailed discussion of the methodologies that have been used at the outer loop of the nested co-design framework.

3.1 Optimal design of experiments approach

Optimal design of experiments (DoE) seeks to choose a set of experiments within a given design space that maximizes a statistical measure of information [42]. Here, G-optimal DoE is used to populate the plant design space with candidate design points at each iteration of the co-design process, while a distance weighted regression approximates the response at off-design points. Since optimal DoE populates experiments based solely on information, statistical design space reduction must be used to exploit the perceived optimum. Figure 10 displays the process diagram for the outer loop of the nested co-design strategy that utilizes optimal design of experiments. The remainder of this section will discuss each of these blocks in greater detail.

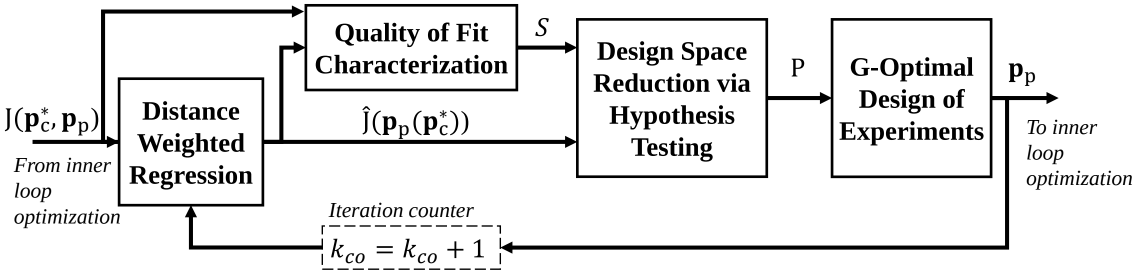
Quality of Fit Characterization
S
Design Space Reduction via Hypothesis Testing
P
G-Optimal Design of Experiments
$\mathbf{p}_\text{p}$
To inner loop optimization
$J(\mathbf{p}_\text{c}^*, \mathbf{p}_\text{p})$
From inner loop optimization
Distance Weighted Regression
$\hat{J}(\mathbf{p}_\text{p}(\mathbf{p}_\text{c}^*))$
Iteration counter
$k_{co} = k_{co} + 1$

Figure 10: Optimal DoE-based outer loop of the nested co-design framework.

3.1.1 G-optimal design of experiments

Efficiently populating design spaces with candidate points that supply large amounts of information can be found throughout traditional optimization and design of experiments literature [42]. The technique selected for a portion of this work falls into the category of *optimal design of experiments* (DoE). In this class of DoE techniques, there are a number of different methods for generating candidate design points, such as D-optimal, G-optimal, A-optimal, T-optimal, and I-optimal. In this work, the G-optimal DoE criterion is used, which has been shown to efficiently populate design spaces with informative points that do not repeat (see [43]). G-optimal DoE relies upon an assumed structure of the performance index as a function of the design variables. The performance index of the candidate plant design, operating at the optimal control parameters, is the focus at the outer loop, and is given by:

$$J(\mathbf{p}_c^*(\mathbf{p}_p),\mathbf{p}_p) = \int_0^{t_f} g(\mathbf{x}(t),\mathbf{u}(t);\mathbf{p}_c^*(\mathbf{p}_p),\mathbf{p}_p)dt. \tag{29}$$

To begin the process of G-optimal DoE, the performance index given in equation (29) is approximated as the inner product of an M element regressor vector, $z(\mathbf{p}_p)$, and M element coefficient vector, β, as follows:

$$\hat{J}(\mathbf{p}_c^*(\mathbf{p}_p),\mathbf{p}_p) = z^T(\mathbf{p}_p)\beta. \tag{30}$$

The regressor vectors corresponding to each batch of N design points can be represented compactly through a regressor matrix, Z:

$$Z = [\begin{array}{ccc} z_1 & \dots & z_N \end{array}]^T, \tag{31}$$

where each column of Z corresponds to a single design point. For example, the column z_i will correspond to the i^{th} candidate design point. The structure of the regressor vector is chosen to reflect the expected dependencies on the plant parameters; e.g., if it is anticipated that the dependency of J on $\mathbf{p}_p$ will be quadratic, then quadratic terms will be included in the regressor vector. G-optimal DoE populates the design space by minimizing an information metric called the prediction variance, which is equal to the maximum entry in the diagonal of $z_i^T(Z^TZ)^{-1}z_i$ (see [44]).

After generating the batch of candidate plant designs, the batch is separated into training and validation points. The performance associated with each candidate plant design is evaluated in a simulation or experiment. The response surface characterization is carried out using the responses associated with the training points, while quality of fit is calculated using the responses of the validation points.

3.1.2 Plant response surface characterization and design space reduction

After completing a control parameter optimization for each candidate plant design, the plant design space must be reduced prior to the next full iteration of the co-design process. To do this, the training points are used to characterize a response surface. This response surface characterization is performed with respect to the plant parameters, having just optimized the controller parameters. The response surface characterization is performed based on the integral performance index value of each training plant design point. To fit a response surface over the entire plant design space with this data, the estimated performance index at any plant design, $\mathbf{p}_p$, is computed through the following distance-weighted average:

$$\hat{J}(\mathbf{p}_c^*(\mathbf{p}_p),\mathbf{p}_p) = \frac{\sum\limits_{i=1}^{N_t} w_i J(\mathbf{p}_c^{*i}(\mathbf{p}_p^i),\mathbf{p}_p^i)}{\sum\limits_{i=1}^{N_t} w_i}. \tag{32}$$

The weight that point $\mathbf{p}_{p,i}$ has on $\mathbf{p}_p$ is given by

$$w_i = \exp\left(-K_r d(\mathbf{p}_p,\mathbf{p}_p^i)^2\right), \tag{33}$$

where K_r is a tunable gain and $d(\mathbf{p}_p, \mathbf{p}_p^i)$ is the Euclidean distance between $\mathbf{p}_p$ and $\mathbf{p}_p^i$. Therefore, as the distance between $\mathbf{p}_p$ and $\mathbf{p}_p^i$ increases, the weight of $J(\mathbf{p}_c^{*i}(\mathbf{p}_p^i), \mathbf{p}_p^i)$ on $\hat{J}(\mathbf{p}_c^*(\mathbf{p}_p), \mathbf{p}_p)$ decreases. In this work, a golden-section search algorithm is used to select the value of K_r that minimizes the mean-squared error between the validation data and estimated response surface, denoted by S and given mathematically by:

$$S = \sqrt{\frac{1}{N_{val}} \sum_{k=1}^{N_{val}} (J(\mathbf{p}_c^*(\mathbf{p}_p^{N_t+k}), \mathbf{p}_p^{N_t+k}) - \hat{J}(\mathbf{p}_c^k(\mathbf{p}_p^{N_t+k}), \mathbf{p}_p^{N_t+k}))^2}. \tag{34}$$

Since the overall goal of the co-design framework is to converge to an optimal system design, it is important to reduce the size of the design space at each iteration so as to focus only on plant designs that could possibly be optimal. The main idea of the design space reduction is to reject, *with confidence*, portions of the design space that produce inferior performance. The response surface computed in the outer loop yields two important quantities that are used to shrink the design space. The first is the approximated performance index for a given plant design operating at the optimal controller design ($\hat{J}(\mathbf{p}_c^*(\mathbf{p}_p^*), \mathbf{p}_p^*)$). The second is the quality of fit metric, S, which characterizes how well the response surface approximates the true system performance index at validation points. To reduce the plant design space, an aggregate quality of fit metric is used because it is assumed that the variance of the integral performance index value will be constant across the plant design space.

By comparing the approximated optimal performance with the approximated performance at all other points in the design space, some plant designs can be rejected. This comparison is carried out by hypothesis testing, which computes a z-score by:

$$z = \frac{\hat{J}(\mathbf{p}_c^*(\mathbf{p}_p),\mathbf{p}_p) - \hat{J}(\mathbf{p}_c^*(\mathbf{p}_p{}^*),\mathbf{p}_p{}^*)}{S\sqrt{2}} \tag{35}$$

Design points are rejected whenever the test statistic falls outside of a designated threshold (i.e. $z > z_{thres}$). For this work, a 95% confidence interval, which corresponds to $z_{thres} = 1.96$, was chosen. The reduced plant design space, which excludes all points rejected by the z-test, is denoted by P.

Using aggregate quality of fit metrics as a surrogate for uncertainty is practical in scenarios where the variance is assumed to be consistent throughout the design space. Since the response surface characterization at the plant level is based on an integral performance index value, as opposed to an instantaneous performance index value in continuous-time, it is reasonable to assume that the variance of the response surface is relatively constant throughout the design space. Additionally, the design space at the plant level is populated with *multiple* design points at each iteration that cover the space in an informative manner, which provides a response surface that is driven by the response associated with designs that efficiently cover the design space. Therefore, the response surface characterization accurately characterizes the response at off-design points.

3.2 Gaussian process-based plant-level approach

The outer loop of the nested framework has two main goals:

1. Maintain a characterization of the expected performance index and measure of uncertainty over the entire plant design space,

2. Select a batch of candidate plant design points that will yield the most information when evaluated experimentally.

Gaussian process (GP) modeling is used to achieve the first goal. GP modeling is a probabilistic regression technique that characterizes a distribution over all possible functions [45]. This distribution is defined by a predictive mean and variance, which is precisely what is needed to achieve the first goal above. A popular machine learning technique that leverages GP modeling to select design points is Bayesian optimization. Bayesian optimization is traditionally an iterative GP-based optimization strategy, but Bayesian optimization only selects a *single* point at each iteration instead of a batch of points. Therefore, batch Bayesian optimization, a variation of Bayesian optimization presented in [46], was selected to populate the plant design space with a batch of candidate design points. Figure 11 displays the process diagram for the outer loop of the nested co-design strategy that utilizes Gaussian process modeling.

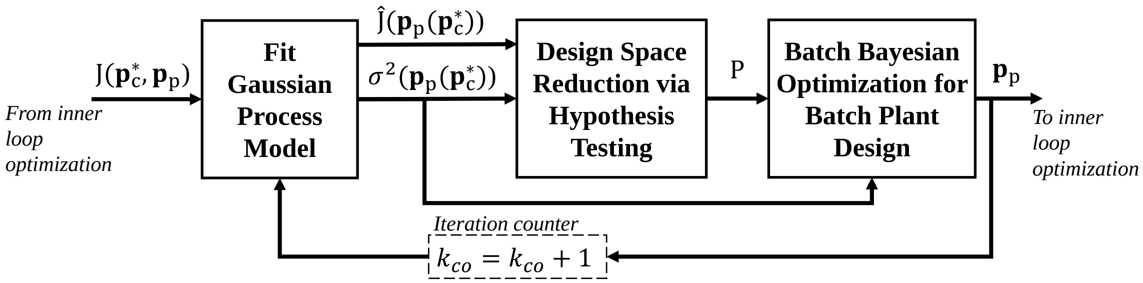
$\mathrm{J}(\mathbf{p}_c^*, \mathbf{p}_p)$
From inner loop optimization
Fit Gaussian Process Model
$\hat{\mathrm{J}}(\mathbf{p}_p(\mathbf{p}_c^*))$
$\sigma^2(\mathbf{p}_p(\mathbf{p}_c^*))$
Design Space Reduction via Hypothesis Testing
P
Batch Bayesian Optimization for Batch Plant Design
$\mathbf{p}_p$
To inner loop optimization
Iteration counter
$k_{co} = k_{co} + 1$

Figure 11: GP-based outer loop of the nested co-design framework.

3.2.1 Background: Gaussian process modeling

Gaussian process (GP) modeling is a non-parametric regression method for learning a unknown function (sometimes called a latent function) from noisy data. A GP model is a collection of Gaussian distributions over a function, conditioned upon the collected data. The mean and covariance parameters from a Gaussian distribution over vectors are analogous to mean and covariance *functions* for GP models. To fully define a GP model, one must define a mean function and covariance function, which provide an expectation of the latent function and structure for covariance of collected observations, respectively. It is important to note that the *true* function value is never observed. Instead, a noise corrupted version of the function is observed. Let the observation model take on the following form:

$$y_i = f(\mathbf{x}_i) + \varepsilon, \tag{36}$$

where i corresponds to the iteration of the observation, $\mathbf{x}$ represents the input whose relationship with f is of interest, and y_i is the observation at iteration i. The process or latent function, $f(\mathbf{x})$, from which the data is collected is completely unknown and is corrupted by ε, which is assumed to be white Gaussian noise (i.e. $\varepsilon \sim \mathscr{N}(0, \sigma^2_{noise})$). As mentioned earlier, a GP model is fully defined by a mean function, $m(\mathbf{x})$, and covariance function $k(\mathbf{x}, \mathbf{x}')$:

$$F(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{37}$$

Here, the mean function represents the expected value of the latent function and the covariance function characterizes the correlation between points within the design space. Mathematically, this is given as:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \tag{38}$$

$$k(\mathbf{x},\mathbf{x}') = \mathbb{E}[(f(\mathbf{x})-m(\mathbf{x}))(f(\mathbf{x}')-m(\mathbf{x}'))]. \tag{39}$$

There are a number of different covariance functions that are used within the GP literature. The most common covariance function is the squared exponential, which in the multivariable case is given by:

$$k(\mathbf{x},\mathbf{x}') = \alpha^2 exp(\frac{-1}{2}(\mathbf{x}-\mathbf{x}')^T\Lambda^{-1}(\mathbf{x}-\mathbf{x}')). \tag{40}$$

Here, α represents the signal noise of the latent function (which is assumed to be constant for all inputs) in $\mathbf{x}$ and Λ^{-1} is a diagonal matrix that contains the squared characteristic length scales $(l_1, l_2, \ldots l_d)$ for each input along the diagonal. The signal noise (α^2), characteristic length scales $(l_1, l_2, \ldots l_d)$, and noise variance (σ^2_{noise}) are referred to as *hyperparameters* of the GP model. Tuning of these hyperparameters is discussed in detail in [45].

A GP model is a methodology for updating prior beliefs about a stochastic process as data are made available. If a collection of observations are taken from a stochastic process and stored in a database, i.e. $\mathbb{D} = \{\mathbf{X}_{\mathbb{D}} = \mathbf{x}_{1:k-1}, \mathbf{Y}_{\mathbb{D}} = y_{1:k-1}\}$, a posterior belief about the the function at the next iteration k can be generated. Consider an arbitrary point within the design space, $\mathbf{x}_k$, whose function value is described by a Gaussian distribution. We arrive at a joint Gaussian distribution for $y_{1:k-1}$ and y, given by:

$$\begin{pmatrix} y_{1:k-1} \\ y_k \end{pmatrix} \sim \mathscr{N}\left(\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{pmatrix} k(\mathbf{X}_{\mathbb{D}}, \mathbf{X}_{\mathbb{D}}) & k(\mathbf{X}_{\mathbb{D}}, \mathbf{x}_k) \\ k(\mathbf{x}_k, \mathbf{X}_{\mathbb{D}}) & k(\mathbf{x}_k, \mathbf{x}_k) \end{pmatrix}\right). \tag{41}$$

Using the joint distribution of Gaussian variables provided in the Appendix, the conditional probability distribution of $p(y_k|y_{1:k-1})$ can be calculated. This conditional probability distribution is characterized by a mean function and prediction variance. For any candidate point $\mathbf{x}_k$, the mean and prediction variance are calculated by:

$$\mu_f(\mathbf{x}_k) = E\{f(\mathbf{x}_k)\} = k(\mathbf{x}_k, \mathbf{X}_{\mathbb{D}})\mathbf{K}_x^{-1}\mathbf{Y}_{\mathbb{D}}, \tag{42}$$

$$\sigma_f^2(\mathbf{x}_k) = var\{f(\mathbf{x}_k)\} = k(\mathbf{x}_k, \mathbf{x}_k) - k(\mathbf{x}_k, \mathbf{X}_{\mathbb{D}})\mathbf{K}_x^{-1}k(\mathbf{X}_{\mathbb{D}}, \mathbf{x}_k), \tag{43}$$

respectively. Here, $\mathbf{K}_x$, is the covariance between all inputs of the collected data plus measurement noise (which is assumed to be white noise) (i.e. $\mathbf{K}_x = k(\mathbf{X}_{\mathbb{D}}, \mathbf{X}_{\mathbb{D}}) + \sigma_{noise}^2 I$).

3.2.2 Batch Bayesian optimization

The ultimate objective of the outer loop plant iteration is to generate a *batch* of candidate design points. In this work, that is accomplished through batch Bayesian optimization via local penalization, as initially described in [46]. Traditionally, Bayesian optimization *sequentially* selects a *single* design point at each time step, but the approach in [46] generates a batch of design points at each iteration by using a local penalization function to model the interaction between elements of the batch. It is important to note that the penalization strategy from [46] is for a maximization problem. The maximization-penalization strategy selects a point $\mathbf{p}_p^j$ by the following

$$\mathbf{p}_p^j = \arg\max_{\mathbf{p}_p \in P} \left[g(\alpha(\mathbf{p}_p)) \prod_{a=1}^{j-1} \phi(\mathbf{p}_p; \mathbf{p}_p^{t,a}) \right]. \tag{44}$$

Here, $\phi(\mathbf{p}_p; \mathbf{p}_p^{t,a})$ are local penalizers that are centered at the points in the batch and $g(\cdot)$ is a differentiable transformation that forces the acquisition function, $\alpha(\mathbf{p}_p)$, to be strictly positive without changing the location of the extrema. The acquisition function used in this work is simply the prediction variance, which is already positive across the design space (i.e. $g(z) = z$). The local penalizer is given by

$$\phi(\mathbf{p}_p; \mathbf{p}_p^{t,a}) = 0.5erfc(-z), \tag{45}$$

$$z = \frac{1}{\sqrt{2\sigma_f^2(\mathbf{p}_p^j)}}(L||\mathbf{p}_p^j - \mathbf{p}_p|| - M + \mu_f(\mathbf{p}_p^j)), \tag{46}$$

where μ_f is the mean function estimate across the entire design space, σ_f^2 is the prediction variance at $\mathbf{p}_p^j$, M is the maximum value of the objective function over the design space (i.e. $M = \max f(\mathbf{p}_p)$), and L is a valid Lipschitz constant. The Lipschitz constant is approximated locally according to the rules provided in [46]. This local penalizer builds a ball centered around $\mathbf{p}_p^j$, with radius dependent on the values of M, L, and value of f at $\mathbf{p}_p^j$. Large values for L (i.e. high variability in f) and proximity of $f(\mathbf{p}_p^j)$ to the optimum shrink the radius. As points are added to the batch, the acquisition function is decreased locally to discourage selecting another point in the batch to be in the same region. The amount of discouragement depends on the variables above.

3.2.3 Plant response surface characterization and design space reduction

After evaluating the performance of each candidate plant design while operating at the optimal controller parameter(s), GP modeling is used as the regression technique to approximate the response at off-design points. Since the outer loop is conducted in an iteration domain, traditional GP modeling can be applied directly. The power of the GP model lies in its ability to provide an estimate of the expected function value and uncertainty at any arbitrary input vector. The predictive mean and variance across the plant design space are updated at each iteration of the outer loop based on equations (42) and (43), respectively.

Since the proposed exploration strategy described in Section 3.2.2 relies on selecting points that maximize uncertainty, the exploration strategy must be augmented with design space reduction to exploit the optimal combined plant and controller design parameters. Based off of the prediction variance characterization from the GP model, error bars around the mean estimate, corresponding to a user-specified level of confidence, can be calculated across the design space. Upper and lower confidence intervals at a generic location i within the plant design space are given by

$$\hat{J}_{upper}(\mathbf{p}_c^*(\mathbf{p}_{p,i}),\mathbf{p}_{p,i}) = \hat{J}(\mathbf{p}_c^*(\mathbf{p}_{p,i}),\mathbf{p}_{p,i}) + k_{conf}\sqrt{\sigma_{pred}^2(\mathbf{p}_c^*(\mathbf{p}_{p,i}),\mathbf{p}_{p,i})}, \tag{47}$$

$$\hat{J}_{lower}(\mathbf{p}_c^*(\mathbf{p}_{p,i}),\mathbf{p}_{p,i}) = \hat{J}(\mathbf{p}_c^*(\mathbf{p}_{p,i}),\mathbf{p}_{p,i}) - k_{conf}\sqrt{\sigma_{pred}^2(\mathbf{p}_c^*(\mathbf{p}_{p,i}),\mathbf{p}_{p,i})}, \tag{48}$$

respectively. The level of confidence, corresponding to the notion of a z-score, is set by the constant k_{conf}. In a minimization problem, a candidate design point is rejected if the lower bound at any candidate point is greater than the upper bound at the perceived optimum, which is mathematically given by the following condition:

$$\hat{J}_{lower}(\mathbf{p}_c^*(\mathbf{p}_{p,i}),\mathbf{p}_{p,i}) \geq \hat{J}_{upper}(\mathbf{p}_c^*(\mathbf{p}_p^*),\mathbf{p}_p^*)\ \forall i\ \in\ [1,s]. \tag{49}$$

CHAPTER 4: REAL-TIME CONTROLLER OPTIMIZATION TOOLS

The nested co-design approach involves a full optimization of the controller for every iteration on the outer loop. In complex systems where it is necessary to run time-consuming simulations and/or experiments to evaluate the performance index of a given design, it becomes advantageous to consider *real-time* control optimization strategies, which allow the control parameter to be optimized *during* the simulations/experiments. Since the desired applications for this controller optimization are for complex systems where a closed-form dynamic model may not exist, there are a number of limitations on the type of controller optimization strategies that can be employed. Those limitations require a controller optimization that:

- does not make any parametric assumptions about the system,
- is implementable in a real-time environment on real hardware,
- explores the design space in a continuous fashion.

4.1 Background: Adaptive control literature

Designing a controller that operates optimally for a system with uncertainties and variations in operating conditions/parameters can be challenging. These uncertainties typically arise when designing a controller for complex systems and can be in the form of uncertain parameter values, varying environmental characteristics, or unmodeled higher-order phenomena. Adaptive control strategies exist for the very purpose of controlling such systems (see [47]).

Most traditional adaptive control strategies depend on significant structural knowledge and assumptions with regard to the system to be controlled. For example, indirect techniques employing gradient-based or recursive least-squares (RLS) parameter estimations are restricted to parametric uncertainties. Direct techniques, such as model reference adaptive control (MRAC) (see [47, 48, 49, 50]) and nonlinear backstepping techniques (see [51] and [52]), require a minimum-phase plant with known relative degree. Unfalsified control, discussed in [53] and [54], identifies the optimal controller from a finite set of candidate control solutions by falsifying candidate solutions that violate desired performance specifications. Performance of each candidate controller is evaluated by switching between controllers from the finite set. The optimal controller is identified by falsifying candidate controllers that violate desired performance specifications. While the aforementioned techniques come with significant performance guarantees, their underlying assumptions limit their application to complex systems.

As the complexity of engineering systems increases, our ability to accurately describe the system dynamics by a small, finite set of equations (and corresponding parameters) diminishes. Many real-world engineering systems rely on complex simulation models or require experimental evaluation. Therefore, optimization strategies that do not require assumptions about the parametric structure of the system are desired. One such strategy in the control literature is extremum seeking (ES) [39]. ES is a non-model based adaptive control technique that relies on a sinusoidal (or random) perturbation and system performance feedback to converge to optimal controller parameter(s). Because of an inherent assumption of time scale separation between the system time constants, filter time constants, and perturbation period in ES, the technique is known to suffer from slow convergence times in many instances.

A number of techniques from the iterative machine learning and design optimization literature avoid making significant assumptions about the system at hand and could conceivably be used as adaptive control laws. In Chapter 3, two compelling strategies from these communities were discussed. The first was optimal design of experiments (DoE), which uses a statistical information metric to make informed design decisions at each iteration. Through the use of response surface characterization and design space reduction, those design decisions are tailored towards a perceived optimum. An analogous continuous-time implementation would require an adaptive control law that maintains a regression and uncertainty characterization in real time, while only visiting a single point at each time step. Secondly, Gaussian process (GP) modeling — a traditionally iterative machine learning strategy — could be used as an adaptive control law by using GP modeling to characterize the system of interest in real time and drive the parameter exploration procedure. To characterize a predictive mean and variance function, GP modeling requires maintenance of a database that includes all of the previously observed points. In adaptive control maintaining this database becomes computationally burdensome because multiple observations are collected every second. For real-time implementation, a GP-based adaptive control law would require a computationally-efficient update law that circumvents the need to maintain the entire database. In the remainder of this chapter, we introduce three options for performing the control parameter adaptation within the inner loop:

- ES, which serves as a benchmark in our simulation and experimental studies;
- A novel heuristic ”continualized” entropy-based adaptation technique;
- A novel recursive Gaussian Process (RGP)-based adaptation technique, which tailors tools from Gaussian Process modeling to a real-time framework.

4.2 Continuous-time control parameter optimization: option 1 - extremum seeking based adaptation law

Extremum seeking (ES) is a common adaptive control algorithm used in continuous-time optimization, where there is limited or no knowledge of how the performance index depends upon the control variable(s) (i.e. no first or second order information is available). A brief summary of ES is provided here. The general block diagram of multivariable ES from [39], with added context to the co-design problem at hand, is shown in Figure 12. In general, ES uses a sinusoidal or random perturbation applied to the approximated optimal controller parameter, in conjunction with feedback of the instantaneous performance index value, to converge to the true optimal controller parameter. The control parameter update shown in Figure 12 corresponds to a single control parameter contained within the control design vector (i.e. $\mathbf{p}_c = \begin{bmatrix} p_{c,1} & p_{c,2} & \cdots & p_{c,L} \end{bmatrix}^T$). In the multiparameter ES case, L perturbation frequencies $\omega_1 < \omega_2 < \cdots < \omega_L$ are used for the identification of L parameters (see [55]). The perturbation frequencies are selected to be sufficiently small with respect to system dynamics and not equal to the frequencies associated with noise. Each perturbation frequency corresponds to a single element within the control design vector. The filter cutoff frequencies ($\omega_{l,i}$ and $\omega_{h,i}$) are selected to be slower than the perturbation cutoff frequency. It has been shown that with proper selection of ω_i, k_i, $\omega_{h,i}$, $\omega_{l,i}$, and a_i (all of the parameters associated with ES are listed in Table 2), ES will converge to a local minimizer or maximizer [39]. As presented in Figure 12, ES is used for maximizing the instantaneous performance index, but minimization problems can be handled by changing the integrator gain to $-k_i$.

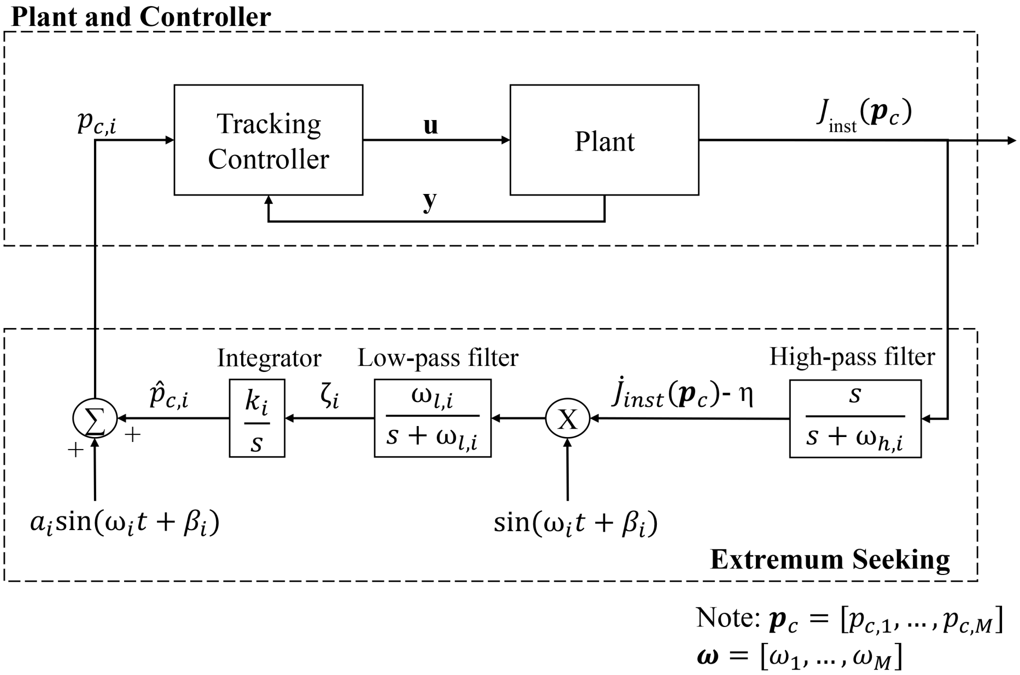
Plant and Controller
$p_{c,i}$
Tracking Controller
u
Plant
y
$J_{\text{inst}}(\boldsymbol{p}_c)$
Integrator
Low-pass filter
High-pass filter
$\hat{p}_{c,i}$
$\frac{k_i}{s}$
ζ_i
$\frac{\omega_{l,i}}{s+\omega_{l,i}}$
X
$J_{inst}(\boldsymbol{p}_c)$- η
$\frac{s}{s+\omega_{h,i}}$
Σ
+
+
$a_i \sin(\omega_i t + \beta_i)$
$\sin(\omega_i t + \beta_i)$
Extremum Seeking
Note: $\boldsymbol{p}_c = [p_{c,1}, \ldots, p_{c,M}]$
$\boldsymbol{\omega} = [\omega_1, \ldots, \omega_M]$

Figure 12: Multiparameter ES block diagram displaying the update for a single element of the control vector. In general, $i = 1,2,3,...,L$ and L is the number of elements in the control design vector. For odd i, $\omega_{i+1} = \omega_i$, $\beta_i = 0$, and $\beta_{i+1} = 0$.

Once standard ES has converged to an optimal solution, the algorithm will continue to perturb the system about the perceived optimal solution. In this framework, the performance index at the optimal controller design ($J(\mathbf{p}_c^*(\mathbf{p}_p),\mathbf{p}_p)$) is needed for the response surface characterization at the outer loop of the co-design strategy. Therefore, at the inner loop, $\mathbf{p}_c$ must converge to a sufficiently small space around $\mathbf{p}_c^*$. Convergence of $\mathbf{p}_c$ to $\mathbf{p}_c^*$ is detected by comparing a *filtered derivative* of $\mathbf{p}_c$ to a threshold. Denoting this filtered derivative by $\dot{\mathbf{p}}_c^{filt}$, convergence is detected at time t_c if:

$$\|\dot{\mathbf{p}}_c^{filt}\| \leq \varepsilon_{ES}, \forall t \in [t_c, t_c - \Delta t]. \quad (50)$$

Proper tuning of ε and Δt ensures that $\dot{\mathbf{p}}_c^{filt}$ must remain *sufficiently small* for a *sufficiently long time period* before convergence is detected.

Table 2: Description of extremum seeking signals and parameters for single control parameter update

Variable	Description
J_{inst}	Instantaneous performance index value
$\dot{J}_{inst}$	Filtered derivative of instantaneous performance index value
$\mathbf{p}_c$	Commanded control parameter vector that has the form: $\mathbf{p}_c = \begin{bmatrix} p_{c,1} & p_{c,2} & \cdots & p_{c,L} \end{bmatrix}^T$
η	Low-level noise
$p_{c,i}$	Commanded control parameter
$\hat{p}_{c,i}$	Estimated optimal control parameter
ζ	Low-pass filter output
$\mathbf{u}$	Control command
$\mathbf{y}$	Feedback vector
k_i	Integrator gain vector
a_i	Perturbation amplitude
ω_i	Perturbation frequency
$\omega_{l,i}$	Low-pass filter frequency
$\omega_{h,i}$	High-pass filter frequency

For $t \geq t_c$, the extremum seeking perturbation is suspended, allowing for calculation of the integral performance index under the optimized control parameter vector, $\mathbf{p}_c^*$. The portion of data acquired *after* convergence is detected is used for the purpose of calculating the integral performance index; i.e., the integral performance index is calculated as:

$$J(\mathbf{p}_c^*(\mathbf{p}_p), \mathbf{p}_p) = \int_{t_c}^{t_c+t_f} g(\mathbf{x}(t), \mathbf{u}(t); \mathbf{p}_c, \mathbf{p}_p) dt. \quad (51)$$

4.3 Continuous-time control parameter optimization: option 2 - entropy-based DoE adaptation law

One clear limitation of the extremum seeking-based adaptation law is that it is local in nature, thereby resulting in a co-design formulation where the outer loop explores the global plant design space, but the inner loop is limited to local control parameter optimization. To remedy this limitation, it is desirable to develop a continuous-time control parameter optimization technique that mirrors the global design space exploration performed by the outer loop plant DoE.

Ultimately, the optimal DoE technique used at the outer loop is based on selecting designs that maximize a statistical *information metric* (for G-optimal design, this is prediction variance). At the end of each iteration, the design space is reduced based on a response surface characterization and quality of fit estimate. To mirror these techniques at the inner loop, we adjust the control parameters, $\mathbf{p}_c$, based on a *continuous-time information metric*. Furthermore, we reject parameters from the design space in *continuous time*, based on a continuous quality of fit update. A fundamental tenet in performing this continuous-time information-based control parameter adjustment and design space reduction is the continuous-time estimate of *normalized information entropy*, $H(\mathbf{p}_c)$, which characterizes how uncertain we are about the performance index for a given value of $\mathbf{p}_c$. While this process is conceptually similar to existing literature on adaptive DoE ([56] and [57]), a fundamental distinction lies in the fact that the adaptive DoE results of [56] and [57] are iteration-based, not continuous time. The novel contribution of the entropy-based DoE is the ability to optimize a parameter vector of interest based on an information metric in *continuous time*. The entropy-based control parameter optimization framework is shown graphically in Figure 13.

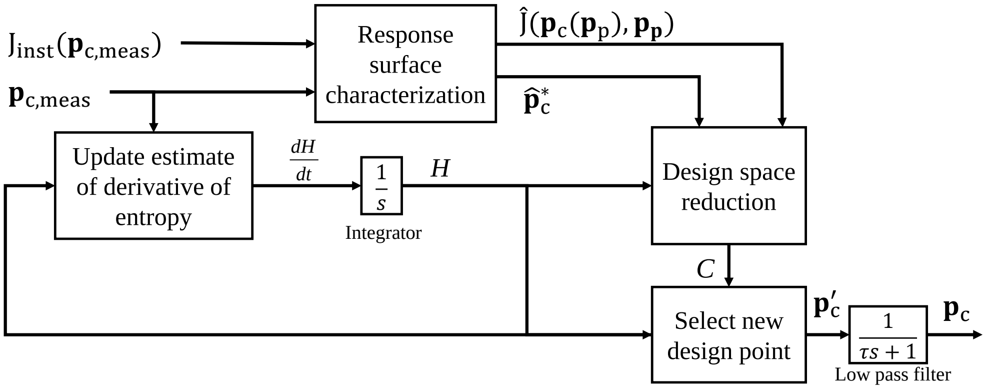

Figure 13: General block diagram of entropy-based DoE adaptation.

4.3.1 Continuous-time quantification of normalized information entropy

In general, information entropy describes the uncertainty associated with a particular variable [40]. Within this framework, information entropy is used to characterize how much we have left to learn about a specific design point. To begin the process, *normalized* information entropy, denoted by $H(\mathbf{p}_c)$, is initialized to a maximum value of one for all points in the design space. In order to implement an the update law for information entropy in *continuous time*, we begin with an iterative entropy update law from equation (52), where the estimated entropy at each point in the design space (where an arbitrary point in the design space is denoted by $\bar{\mathbf{p}}_c$) is updated based on the proximity of $\bar{\mathbf{p}}_c$ to the present design point, $\mathbf{p}_c$. We then “continualize” the iterative entropy update law of equation (52) as follows:

$$
\begin{aligned}
H_{k+1}(\bar{\mathbf{p}}_c) &= H_k(\bar{\mathbf{p}}_c)\exp\left(\frac{-K_{ent}}{d(\mathbf{p}_c,\bar{\mathbf{p}}_c)+\varepsilon}\right), && (52)\\
\Rightarrow H_{k+1}(\bar{\mathbf{p}}_c) - H_k(\bar{\mathbf{p}}_c) &= H_k(\bar{\mathbf{p}}_c)\left(\exp\left(\frac{-K_{ent}}{d(\mathbf{p}_c,\bar{\mathbf{p}}_c)+\varepsilon}\right)-1\right). && (53)
\end{aligned}
$$

Because $H_{k+1}(\bar{\mathbf{p}}_c) - H_k(\bar{\mathbf{p}}_c) \approx \frac{dH}{dt}(\bar{\mathbf{p}}_c)\Delta T$ it follows that

$$
\frac{dH}{dt}(\bar{\mathbf{p}}_c)\Delta T \approx H(\bar{\mathbf{p}}_c)\left(\exp\left(\frac{-K_{ent}}{d(\mathbf{p}_c,\bar{\mathbf{p}}_c)+\varepsilon}\right)-1\right). \qquad (54)
$$

Here, $H(\bar{\mathbf{p}}_c)$ is the normalized information entropy for a candidate design point, ΔT is the amount of time between parameter adjustments in the iterative framework, K_{ent} is a gain that reflects the "value" of a single point, $d(\mathbf{p}_c, \bar{\mathbf{p}}_c)$ is the distance between the current control design and candidate control design $\bar{\mathbf{p}}_c$, and ε is a small constant that prevents division by zero. K_{ent} is tunable by the designer based on the system of interest. Ultimately, the above derivation leads to the following continuous-time normalized entropy update law:

$$
\frac{dH}{dt}(\bar{\mathbf{p}}_c) = KH(\bar{\mathbf{p}}_c)\left(\exp\left(\frac{-K_{ent}}{d(\mathbf{p}_c,\bar{\mathbf{p}}_c)+\varepsilon}\right)-1\right). \qquad (55)
$$

To implement this entropy update law, the control parameter design space is quantized into a finite grid of $\bar{\mathbf{p}}_c$ values, and the update law is applied for each value in the grid.

Figure 14 illustrates how information entropy evolves during a ten second sample simulation for a one-dimensional design space. The upper half of the figure shows the values of p_c (a scalar in this case) as a function of time. In the lower portion, the entropy at the beginning (left) and end (right) of the simulation can be seen. Because values of p_c greater than 0.2 were not visited during the 10 second period, no reduction in normalized entropy was observed over this range of design parameters.

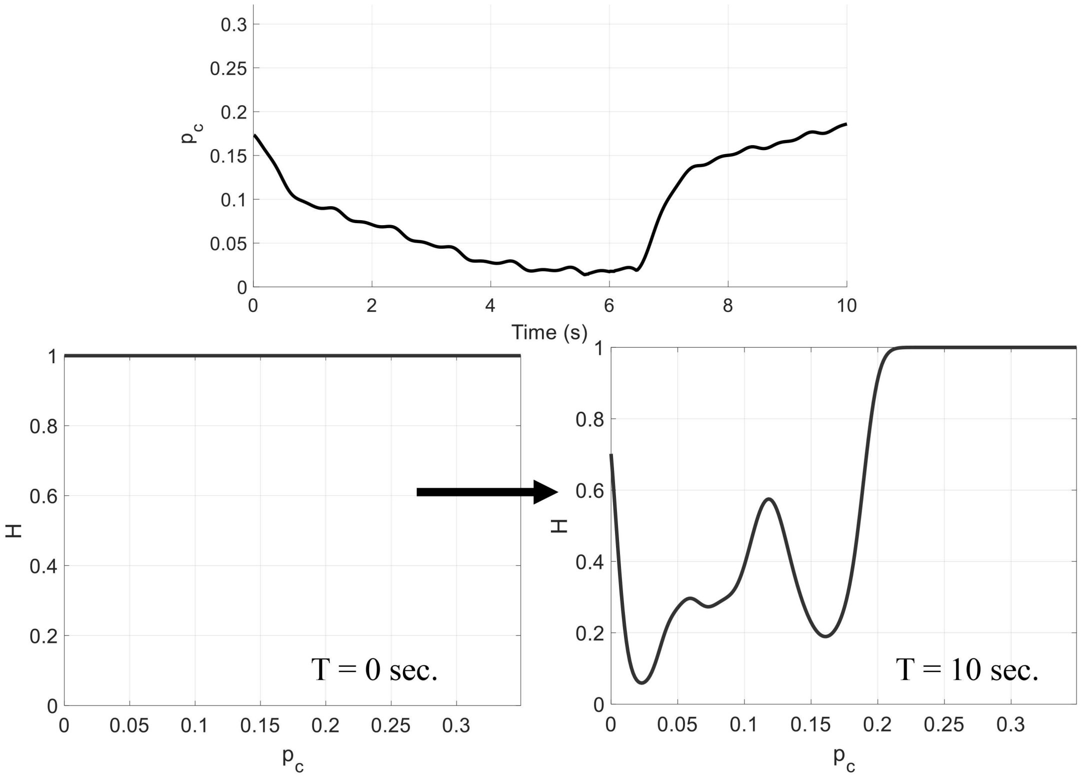

Figure 14: Visualization of normalized entropy evolution over 10 seconds. The top plot shows the design points that have been visited as a function of time. The bottom plots show the values of entropy at the over the design space.

4.3.2 Continuous-time control parameter update law

Unlike extremum seeking, where the control parameter is driven gradually toward the (local) estimated optimal value, the entropy-based control parameter update law selects the next control parameter based on which regions of the design space are likely to yield the most *information* about behavior of J_{inst} with respect to $\mathbf{p}_c$. Thus, the controller moves towards locations of high entropy. However, it is quite common that the regions of highest entropy are far from the current operating point. To capture the requirement that $\mathbf{p}_c$ cannot undergo very large jumps over very short time intervals, the following update law is used to determine the best subsequent control design to evaluate, denoted by $\mathbf{p}_c^{'}$:

$$\mathbf{p}_c^{'} \quad = \quad \arg\max_{\bar{\mathbf{p}}_c \in C} H(\bar{\mathbf{p}}_c) exp\Big(-K_{sel}(d(\mathbf{p}_c, \bar{\mathbf{p}}_c) d(\mathbf{p}_c, \mathbf{p}_{c,prev}))^2 \Big) \tag{56}$$

where $\mathbf{p}_{c,prev}(t) = \mathbf{p}_c(t - T_d)$ for a small delay time, T_d. Here, K_{sel} is a constant gain and $d(\mathbf{p}_c, \mathbf{p}_{c,prev}))$ is the distance between the controller design tested at time $t - T_d$ and the controller design presently tested. Inclusion of the term $d(\mathbf{p}_c, \mathbf{p}_{c,prev})$ in equation (56) ensures that $\mathbf{p}_c^{'}$ does not continue to get adjusted after it was recently changed by a large amount. Inclusion of the term $d(\mathbf{p}_c, \bar{\mathbf{p}}_c)$ helps to ensure that near-term subsequent candidate designs do not veer too far from the present value of $\mathbf{p}_c$. To ensure that the lower-level controller can track the commanded control design, the output of the entropy-based DoE is passed through a first-order filter, as illustrated in Figure 13.

4.3.3 Continuous-time controller response surface characterization and design space reduction

As discussed in the previous subsection, the entropy-based control parameter adaptation law adjusts the control parameter in the direction of maximum entropy, not necessarily in the direction of the optimal value. To estimate the optimal control parameter, $\mathbf{p}_c^*$, another step is needed. This step is the continuous-time response surface characterization and design space reduction, wherein (i) a continuous estimate of $J_{inst}(\mathbf{p}_c(\mathbf{p}_p))$ is maintained and (ii) control parameters that are nearly certain to be sub-optimal based on this response surface are excluded from the candidate controller design space, C. It is important to note that the response surface characterization at this stage is performed *with respect to the control parameters alone, for a given plant design.* Once the inner loop control parameter optimization is complete, a separate response surface characterization, with respect to the plant parameters, is performed.

To perform a response surface characterization in continuous time, we rely on recursive least squares (RLS) estimation. To do this, the instantaneous performance index is parameterized as:

$$\hat{J}_{inst}(\mathbf{p}_c(\mathbf{p}_p)) = X^T(\mathbf{p}_c)\beta, \tag{57}$$

where X is a regressor vector and β is a vector of undetermined coefficients. These undetermined coefficients are estimated through the following discrete-time recursive least squares (RLS) update law, detailed in [58], where k represents the time step:

$$\begin{aligned}
V(k+1) &= V(k) - \frac{V(k)X^T(k+1)X(k+1)V(k)}{1+X(k+1)V(k)X^T(k+1)}, && (58)\\
\gamma(k+1) &= V(k+1)X^T(k+1), && (59)\\
e &= J_{inst}(\mathbf{p}_c(\mathbf{p}_p)) - X(k+1)\hat{\beta}(k), && (60)\\
\hat{\beta}(k+1) &= \hat{\beta}(k) + \gamma(k+1)e. && (61)
\end{aligned}$$

Once $\hat{J}_{inst}(\mathbf{p}_c(\mathbf{p}_p))$ has been obtained, the corresponding minimizer, $\mathbf{p}_c^*(\mathbf{p}_p)$, can be found. To avoid future exploration of regions of the design space that cannot possibly be optimal, a mechanism for reducing the size of the design space must be introduced. The goal of the design space reduction is to reject portions of the design space that are determined to be sub-optimal, with some amount of certainty. Here, the normalized entropy was used as an estimate of uncertainty. In particular, candidate designs are rejected when the following inequality holds:

$$\hat{J}_{inst}(\hat{\mathbf{p}}_c^*(\mathbf{p}_p)) < \hat{J}_{inst}(\mathbf{p}_c(\mathbf{p}_p)) - K_{rej}H(\mathbf{p}_c), \tag{62}$$

where K_{rej} is tuned to capture the relationship between normalized entropy and uncertainty. Thus, a candidate control design will be rejected if its associated performance index exceeds the estimated optimal performance index by an amount that depends on uncertainty (the larger the uncertainty, the larger this amount will be). Note the inequality presented above is for a minimization problem, but can be reformulated for maximization problems. For the entropy-based DoE adaptation, design space reduction is carried out by utilizing a normalized information metric as a surrogate for uncertainty. Design space reduction based on normalized entropy is an imperfect approach because the statistical significance of normalized entropy has not been quantified. However, this approach is valuable because the normalized information metric characterizes the uncertainty for each point within the design space, thereby acting as *vectorized* quality of fit. In a continuous-time environment, a vectorized quality of fit is preferred over an aggregate quality of fit because:

1. aggregate quality of fit metrics require partitioning points into training and validation points, which in continuous time would require discrete intervals of training and validation;

2. extensive exploration of some portions of the design space may result in a good aggregate quality of fit that could result in rejecting portions of the design space that remain unexplored.

4.3.4 Summary of tuning parameters for entropy-based DoE

This section is provided as an aid for tuning the parameters involved with entropy-based DoE. There are four parameters that require tuning by the user, namely K, K_{ent}, K_r, and K_{sel}. Table 3 provides insight for tuning these parameters. Proper tuning of these parameters requires some knowledge of the system to be optimized. Additionally, the tuning here is heuristic, and formalization of this adaptive technique is a topic discussed in Chapter 4.4.

Table 3: Description of tuning parameters for entropy-based DoE control strategy.

Parameter	Description	Tuning Description
K_{ent}	Entropy reduction length scale	Reflects the “distance” that two design points must lie from each other before information at one design point no longer reduces entropy at another.
K	Entropy gain	Reflects the significance of a collecting the results at a single design point
K_r	Exponential regression length scale	Reflection of covariance between design points; tuned using *Golden Section Search*
K_{sel}	Selection gain	Incentivizes selecting design points at time step $k+1$ that are around previous design point at time step k

4.4 Continuous-time control parameter optimization: option 3 - Recursive Gaussian process-based adaptive control

One adaptive control solution from the machine learning community involves maintaining a Gaussian process (GP) model characterization of the system at hand, then using that

GP model to drive a parameter exploration law. GP modeling arose as an iterative technique for developing models of uncertain systems by treating the systems as black-box functions [45]. However, traditional GP modeling requires new observations to be appended to an ever-growing database of all past data. Sparse GP strategies, such as [59], reduce the computational burden by fixing the size of the database. Once the cap for the database has been met, a criterion is evaluated to determine if the new point should replace a point in the database or be considered via a projection onto the points in the database. The author in [60] and [61] describes a recursive Gaussian Process (RGP) modeling technique that avoids the need to maintain a library at all by recursively updating the statistics of the GP model as observations from the system are collected. Since the RGP modeling strategy was developed for iterative applications, two key challenges still exist for real-time adaptive control applications:

- RGP modeling traditionally assumes noise to be uncorrelated from one iteration to the next. However, in a real-time control environment with short time steps, temporal correlation must be taken into account.
- RGP modeling provides a mechanism for characterizing a complex system's performance over the design space, but it is still necessary to define a strategy for *exploring* that design space and *identifying* the optimal parameter value in that design space.

As a standalone contribution of this research, the RGP-based adaptation replicates traditional iterative design optimization (namely Bayesian optimization) techniques in real-time by overcoming the limitations listed above. The RGP-based adaptation places the concepts from the entropy-based DoE adaptation into a statistical foundation. Specifically, the novel concept of recursively updating a regression and uncertainty characterization in continuous time from the entropy-based DoE adaptation was statistically formalized by extending RGP modeling to real-time applications. The process diagram for the RGP-based adaptation is shown in Figure 15.

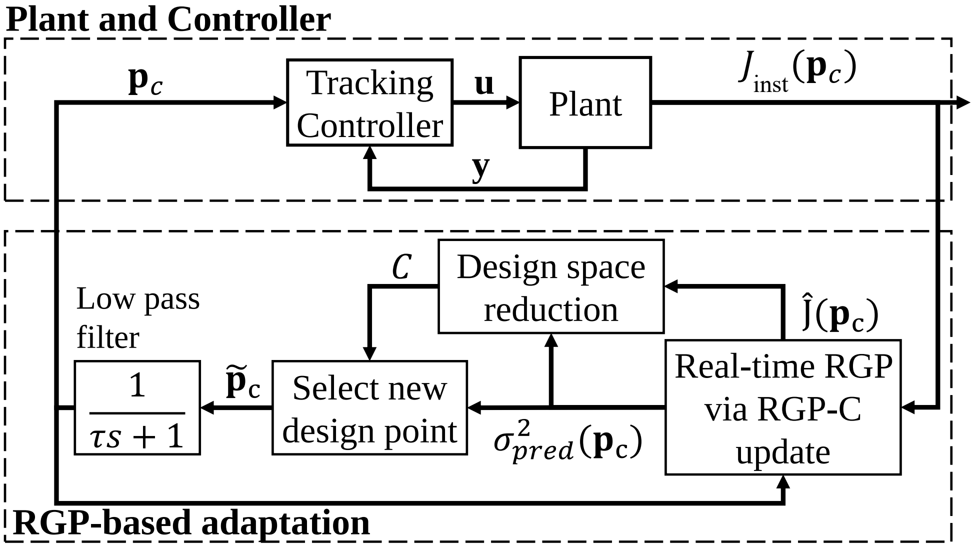

Figure 15: General block diagram of RGP-based adaptation.

4.4.1 Background: recursive Gaussian process modeling

From predictive mean and variance equations (see equations (42) and (43)), it is clear that the prediction mean and variance at any candidate point within the design space relies on the inverse of a covariance matrix, $\mathbf{K}_x$, which is increasing in size at each time step. In the context of control, this corresponds to inverting the covariance matrix and augmenting the covariance matrix with an additional row and column at each time step. Obviously, with a small time step (i.e. a sub-second time step), this matrix can become extremely large in a relatively short amount of time. For this reason, directly applying update laws that are driven by GP modeling (such as Bayesian optimization) to continuous-time controller adaptation is not practical.

To overcome this limitation of iterative GP modeling techniques, the author in [60] and [61] presents a recursive update for GP modeling that avoids the need to maintain a database of previously tested points. The mean and covariance functions are updated sequentially as new observations are collected. A finite number of basis vectors are used to discretize a continuous design space. The collection of basis vectors that span a given design space will be denoted by the matrix $\mathbf{X} = \begin{bmatrix} \bar{\mathbf{p}}_{c,1} & \bar{\mathbf{p}}_{c,2} & \cdots & \bar{\mathbf{p}}_{c,s} \end{bmatrix}^T$. Each element of this matrix, $\bar{\mathbf{p}}_{c,i}$, corresponds to an individual design in the candidate design space. If the candidate design space is 1D, the basis vectors will correspond to a simple grid of the design space (i.e. $\bar{\mathbf{p}}_{c,i} \in \mathbb{R}^{1\times 1}$), but higher order design spaces may require more efficient basis vectors. Introducing basis vectors allows the algorithm to maintain a consistent computation time as time evolves.

The RGP algorithm uses two steps to calculate the posterior distribution of the latent function. The first is an inference step that calculates the expected mean and covariance at the current point based upon the prior distribution. The inference step is given by:

$$\begin{aligned} \mathbf{J}_k &= k(\mathbf{p}_c^k, \mathbf{X})k(\mathbf{X}, \mathbf{X})^{-1}, && (63) \\ \mu_k^p &= \mathbb{E}[f(\mathbf{p}_c^k)] = m(\mathbf{p}_c^k) + \mathbf{J}_k(\mu_{k-1}^f - m(\mathbf{X})), && (64) \\ C_k^p &= k(\mathbf{p}_c^k, \mathbf{p}_c^k) - \mathbf{J}_k(C_{k-1}^f - k(\mathbf{X}, \mathbf{X}))\mathbf{J}_k^T. && (65) \end{aligned}$$

Here, μ_k^p is the predicted mean function value for the input vector at the current time step and C_k^p is the expected variance for the input vector at the current time step. The vector $\mathbf{J}_k$

is used to interpolate the mean function and uncertainty estimates at any generic point $\mathbf{p}_c^k$, which may not correspond exactly to a basis vector. The RGP formulation treats the mean function at each basis vector location as a state within a Kalman filter update. Therefore, at each time step the approximation of the mean, μ_k^f, is updated. As new observations are collected, the prior mean and covariance estimates are updated to reflect the new observation. The update step is given by:

$$\begin{aligned} \bar{G}_k &= C_{k-1}^f \mathbf{J}_k^T (C_k^p + \sigma^2 I)^{-1}, && (66) \\ \mu_k^f &= \mathbb{E}[f(\mathbf{X})] = \mu_{k-1}^f + \bar{G}_k (y_k - \mu_k^p), && (67) \\ C_k^f &= C_{k-1}^f - \bar{G}_k \mathbf{J}_k C_{k-1}^f. && (68) \end{aligned}$$

4.4.2 Extension of RGP modeling to real-time systems

The RGP update described in [60] and [61] makes a key assumption that restricts its ability to be directly applied to a real-time control application. Namely, it is assumed that each successive observation can be characterized by a latent function, $J(\mathbf{p}_c)$, which is corrupted by white noise. In real-time control applications, however, the instantaneous performance index of the system depends not only on the instantaneous choice of control parameter ($\mathbf{p}_c$), but also upon the system's state ($\mathbf{x}$) at that time. Because of this, two measurements taken in rapid succession are likely to be correlated even in the event that $\mathbf{p}_c$ changes by a large amount between time steps. That is because the amount by which the state changes over a single time step is restricted by the system's dynamics. Therefore, to use an RGP algorithm effectively for online adaptive control, temporal correlation must be taken into account. In order to do this, the RGP model is extended to consider colored noise in the observations, the parameters of which are tuned to reflect the temporal correlation between measurements taken in rapid succession, which arises from the system dynamics.

One method for considering colored noise is to introduce a state model that characterizes the temporal correlation in the noise, resulting in the following set of dynamic equations:

$$\begin{aligned} \begin{bmatrix} \dot{\xi} \\ \dot{n} \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ 0 & \frac{-1}{\tau} \end{bmatrix} \begin{bmatrix} \xi \\ n \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\tau} \end{bmatrix} w, \\ y &= \begin{bmatrix} \mathbf{J}_k & 1 \end{bmatrix} \begin{bmatrix} \xi \\ n \end{bmatrix}. \end{aligned} \tag{69}$$

Here, ξ represents the mean function value at the basis vector locations, n represents the colored noise state, τ represents the colored noise time constant (providing a measure of temporal correlation), w is Gaussian white noise with zero mean, and $\mathbf{J}_k$ has the same interpretation as equation (63). Note that the present work, as suggested in equation (69), assumes a time-invariant latent function. Time-varying latent functions represent a future area of research that remains open.

Using the above dynamic model formulation, colored noise can be considered within a Kalman filter update through the use of two different strategies. The first is augmented state approach, which uses the system dynamics presented in equation (69) in a Kalman filter; however, this has been shown to lead to numerical issues in certain cases [62]. The second is the measurement differencing approach detailed in [62]. Since the Kalman filter equations are more easily digestible in discrete time than continuous time, let's convert the system dynamics into discrete time. The system dynamics in discrete time are given by:

$$x_k = F_{k-1}x_{k-1} + w_{k-1}, \tag{70}$$

$$y_k = H_k x_k + v_k, \tag{71}$$

$$v_k = \psi_{k-1}v_{k-1} + \zeta_{k-1}. \tag{72}$$

Here, F_{k-1} is the state transition matrix in discrete time, H_k is the output vector in discrete time, v_k is the colored noise state, w_{k-1} is white noise, ζ_{k-1} is white noise, x_k is the unobserved state, and y_t is the observation of the state corrupted by colored measurement noise. An auxiliary signal, y', that contains only white noise instead of colored noise, is defined as follows:

$$y'_{k-1} = y_k - \psi_{k-1}y_{k-1}, \tag{73}$$

$$= (H_kF_{k-1} - \psi_{k-1}H_{k-1})x_{k-1} + (H_kw_{k-1} + \zeta_{k-1}), \tag{74}$$

$$= H'_{k-1}x_{k-1} + v'_{k-1}. \tag{75}$$

By using the new output vector, H'_k, the system dynamics are now represented by the following set of equations:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + w_{k-1}, \\ y'_k &= H'_k x_k + v'_k. \end{aligned} \tag{76}$$

The dynamic equations presented in equation (76) are corrupted by random noise with zero-mean and finite variance (i.e. $v'_k \sim \mathscr{N}(0, \bar{Q}_k)$ and $w_{k-1} \sim \mathscr{N}(0, Q_k)$), which allows for the

Kalman filter to be directly applied to the new system dynamics presented in equation (76).

If the system dynamics of the GP model shown in equation (69) are applied to the Kalman filter considering colored noise discussed in [62], the update step for the RGP now changes to the following:

$$\bar{G}_k = C^f_{k-1} H'^T_k (H'_k C^f_{k-1} H'^T_k + \sigma^2 I)^{-1}, \tag{77}$$

$$\mu^f_k = \mu^f_{k-1} + \bar{G}_k (y'_k - H'_k \mu^f_{k-1}), \tag{78}$$

$$C^f_k = C^f_{k-1} - \bar{G}_k H'_k C^f_{k-1}. \tag{79}$$

In summary, the proposed extension of the RGP algorithm utilizes a measurement differencing approach in order to consider colored noise within a Kalman filter update, where the states are the mean function value at each basis vector location. The real-time RGP algorithm will be referred to as the RGP-C update to distinguish it from the iterative RGP algorithm in [60]. The inference step of the RGP-C algorithm remains the same as the RGP update, while the update step for the RGP-C algorithm is given by equations (77)-(79).

4.4.3 Recursive Gaussian process-based adaptive control: Real-time design space exploration strategy

Even though the RGP-C update provides a mechanism for updating a predictive mean and variance estimate over the design space, it does not tell us which point to visit at the next time step. For this reason, we must define an exploration strategy. Inspired by iterative optimal design of experiments, the exploration strategy used here selects points that maximize an information metric (e.g. prediction variance) at each time step.

The overall goal with this exploration strategy at each time step is to select the point that maximizes the prediction variance ($\sigma^2_{pred}(\mathbf{p}_c)$), but it must be acknowledged that large design space moves cannot be realized within a single time step. For physical systems described by slow time constants, large design space moves at each time step will result in large transients that need to settle before additional design decisions can be made. Recognizing the physical limitations of large design space moves, the control parameter update law is given by:

$$\bar{\mathbf{p}}_c^k = \arg\max_{\bar{\mathbf{p}}_c \in C} \quad \sigma^2_{pred}(\bar{\mathbf{p}}_c) exp\Big(-K_{sel}(d(\bar{\mathbf{p}}_c^k, \bar{\mathbf{p}}_c^{k-1}))^2\Big). \tag{80}$$

This update law accounts for uncertainty (through prediction variance) and distance between potential control parameters at subsequent time steps. The exponential term discourages the selection of design points that are far from the design point at the previous time step.

4.4.4 Recursive Gaussian process-based adaptive control: Real-time design space reduction

As with the entropy-based DoE adaptation, the design space exploration strategy focuses on moving to control parameters corresponding to high uncertainty, without explicitly considering expected performance at those parameters. Thus, the design space exploration strategy must be augmented with a design space reduction scheme in order to exploit areas of improved performance. Portions of the design space are rejected when that region is determined to be sub-optimal with some level of confidence. To determine if a point is sub-optimal, we determine if the difference between the mean function value at any point within the design space and the mean function value at the perceived optimum point is significant by calculating error bars throughout the design space. From the RGP-C update, an estimate of the prediction mean and variance are available to calculate an error bar at each basis vector location. The upper and lower error bars at an arbitrary point i, are given by:

$$\hat{J}_{upper}(\bar{\mathbf{p}}_{c,i}) = \hat{J}(\bar{\mathbf{p}}_{c,i}) + k_{conf}\sigma_{pred}(\bar{\mathbf{p}}_{c,i}), \tag{81}$$

$$\hat{J}_{lower}(\bar{\mathbf{p}}_{c,i}) = \hat{J}(\bar{\mathbf{p}}_{c,i}) - k_{conf}\sigma_{pred}(\bar{\mathbf{p}}_{c,i}), \tag{82}$$

respectively. Here, k_{conf} is a constant that corresponds to the desired confidence level in the form of a z-score, and σ_{pred} is the prediction standard deviation, which is calculated as the square root of the prediction variance, σ^2_{pred}. In a maximization problem, a candidate design point is rejected if the upper bound at the candidate point is less than the lower bound at the perceived optimum, which is mathematically given by the following condition:

$$\hat{J}_{upper}(\bar{\mathbf{p}}_{c,i}) \leq \hat{J}_{lower}(\bar{\mathbf{p}}^*_c)\ \forall i\ \in\ [1,s]. \tag{83}$$

Note that s denotes the number of basis vectors in the RGP formulation.

4.4.5 Convergence detection for RGP-based adaptive control law

As the RGP-based adaptive control proceeds, the control parameter design space (C) continues to shrink in size. To evaluate the performance of this adaptation and calculate an integral performance value, an algorithm for detecting sufficient convergence must be defined. To accomplish this, convergence is detected by calculating a metric that quantifies the “flatness” of the response surface over the reduced design space. The metric used here is the percent difference between the maximum and minimum value of the response surface over the reduced design space. Once the percent difference falls below a user-specified threshold value (e.g. 1%, 5%, etc.), the design space exploration is turned off and the point that minimizes the objective function over the reduced design space is selected as the operating point. Quantitatively, convergence is detected when the following inequality is met:

$$\frac{|\hat{J}_{max,r}(\bar{\mathbf{p}}_c) - \hat{J}_{min,r}(\bar{\mathbf{p}}_c)|}{0.5(\hat{J}_{max,r}(\bar{\mathbf{p}}_c) + \hat{J}_{min,r}(\bar{\mathbf{p}}_c))} \leq \varepsilon_R, \tag{84}$$

where

$$\hat{J}_{max,r}(\bar{\mathbf{p}}_c) = \max(\hat{J}(\bar{\mathbf{p}}_c)) \; \forall \; \bar{\mathbf{p}}_c \in C(k), \tag{85}$$

$$\hat{J}_{min,r}(\bar{\mathbf{p}}_c) = \min(\hat{J}(\bar{\mathbf{p}}_c)) \; \forall \; \bar{\mathbf{p}}_c \in C(k). \tag{86}$$

Here, $C(k)$ denotes the reduced design space at the k^{th} time step and ε_R is the desired percent difference in performance at convergence.

4.4.6 Uncertainty characterization of RGP-based adaptive control

The statistical nature of GP modeling requires convergence to be detected probabilistically, rather than deterministically. Specifically, convergence can be detected in a statistical sense (i.e., with some probability, the converged-to design falls within some threshold of the true optimum). To characterize the uncertainty in the maximum (or minimum) value of the latent function, the GP model over the basis vectors is used to characterize a joint cumulative distribution function (CDF). In general, a joint CDF gives the probability that two or more random variables fall within a specific range of values. Quantitatively, the joint CDF is specified as follows:

$$F_{X_1,\ldots,X_N}(x_1,\ldots,x_N) = P(X_1 \leq x_1,\ldots,X_N \leq x_N). \tag{87}$$

From the basis vectors in the RGP-based adaptation, there is a finite set of s normally distributed random variables, each with a mean and prediction variance approximation. Since the design space is discretized into s design points, there will always be a nonzero probability that any point in the design space is the true optimum (and the corresponding latent function value is the true optimum objective function value). To characterize the expectation and uncertainty of the maximum of the latent function value, each of the s normally distributed random variables are used to generate a joint CDF over potential latent function values. From the joint CDF, the probability of the latent function value at each basis vector location taking on any value ($a_\mu \in \mathbb{R}$) can be calculated by

$$F_{\mu(\bar{\mathbf{p}}_{c,1}),\ldots,\mu(\bar{\mathbf{p}}_{c,s})}(a_\mu,\ldots,a_\mu) = P(\mu(\bar{\mathbf{p}}_{c,1}) \leq a_\mu,\ldots,\mu(\bar{\mathbf{p}}_{c,s}) \leq a_\mu). \tag{88}$$

To characterize a CDF over potential values of the maximum of the latent function, a large range of values for a_μ was considered. The resulting CDF, denoted by F_{max}, provides the probability that the maximum latent function value from the RGP model falls below any value of a_μ. A probability density function (PDF), denoted by f_{max}, was calculated by taking the derivative of F_{max} with respect to a_μ.

The goal here is to calculate the expectation and variance of the maximum of the latent function, which are given by

$$\mathbb{E}[\max(\hat{J})] = \mu(\max(\hat{J})) = \int_{-\infty}^{\infty} x f_{max}(x)dx, \tag{89}$$

$$\text{var}[\max(\hat{J})] = \sigma^2(\max(\hat{J})) = \int_{-\infty}^{\infty} (x-\mu)^2 f_{max}(x)dx, \tag{90}$$

respectively. The notation of this section is generalizable to the minimum or maximum of the latent function.

When the RGP-based adaptation is placed in the inner loop of the nested co-design framework, $\sigma^2(\max(\hat{J}))$ is used to approximate the uncertainty associated with the performance values at the perceived optimum by replacing σ^2_{noise} in equations (42) and (43).

4.4.7 Recursive Gaussian process-based adaptive controller adaptation simulation-based results

Having described the mathematical formulation for the RGP-based control parameter adaptation, we now examine the performance of this technique *in isolation* on the AWE system. Stationary and crosswind flight case studies were used to evaluate the performance of the RGP-based adaptation. Due to its esteemed place in the adaptive control literature, ES is used as a benchmark to illustrate the efficiency and potential of the novel adaptation strategies in this work, namely the entropy-based approach and RGP-based adaptation.

4.4.7.1 Stationary flight controller optimization

Initially the RGP-based adaptation was validated by optimizing the attitude/altitude controller for stationary flight of the BAT. The attitude/altitude controller of the BAT discussed in Chapter 2.3 was augmented with the RGP-based adaptation to optimize the pitch angle setpoint in real time. Table 4 summarizes the decision variables and objective function used for this case study.

Table 4: Summary of optimization problem features for simulation-based stationary flight optimization

Feature	Symbol	Description
Control Parameters	θ_{sq}	Pitch angle setpoint
Instantaneous objective function	$k_1\Phi^2 + k_2\psi_e^2 + k_3\phi_e^2$	Instantaneous quality of flight
Integral objective function	$\int_{t_c}^{t_c+\Delta t}(k_1\Phi^2 + k_2\psi_e^2 + k_3\phi_e^2)dt$	Integral quality of flight

Figure 16 displays the design space reduction for the trim pitch angle when initializing the trim pitch angle at 2°. The upper and lower limit on the design space at the end of the simulation are 14.86° and 12.7°, respectively. The percent difference between the maximum mean function value and the minimum mean function value within the reduced design space is 1.6%; thus any point within this reduced space results in similar performance. In contrast, the percentage difference between the maximum and minimum mean function value over the full design space is 40.3%.

Figure 17 provides the raw instantaneous performance measurements obtained over the course of the simulation, along with the RGP and RGP-C estimates of the objective function (to be minimized in this case). The raw data exhibits tremendous variation about the mean due to the perturbation; however, a clear trend can be observed from observation, which is properly accounted for by the RGP-C model. Also, from the raw data, it can seen that the optimal pitch angle setpoint lies somewhere between 12.5° and 15°, which is extremely close to the final design space to which the adaptive control algorithm converges

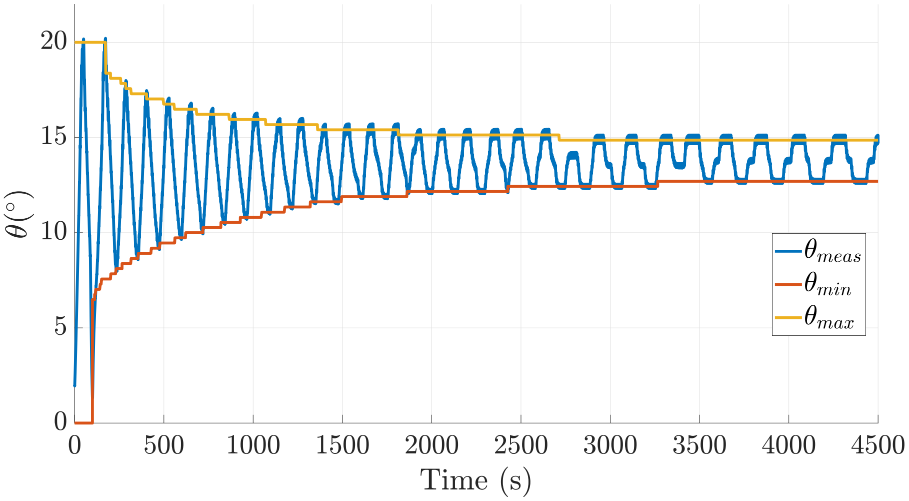

Figure 16: Visualization of design space reduction for AWE controller optimization with 1D design space (trim pitch angle setpoint). The initial condition of the trim pitch was 2°.

(as indicated by Figure 16).

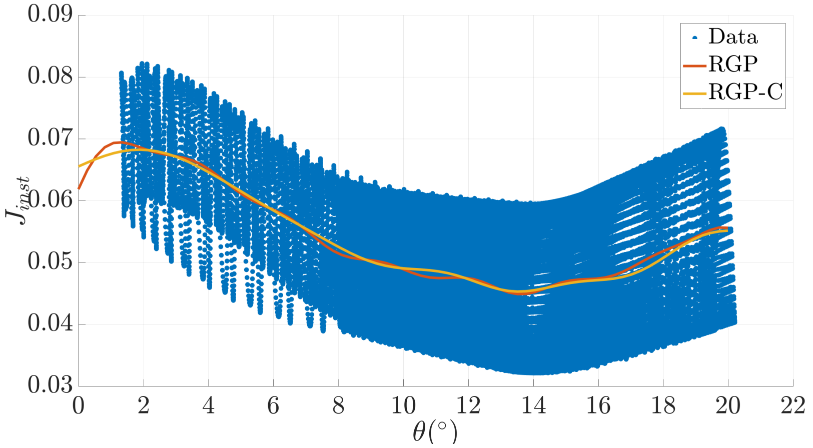

Figure 17: Visualization of the mean function approximation of the RGP-C update compared to the original RGP algorithm after the simulation.

To evaluate the performance of the RGP-based adaptation, the convergence time and performance of the point to which the ES and RGP-based adaptions converged to was compared. A Pareto front for each algorithm was generated by varying the convergence criteria for each algorithm. For ES, this corresponds to adjusting the threshold on the filtered derivative (i.e. ε_{ES} in equation (50)). Meanwhile, for the RGP-based adaptation, the percent difference threshold was adjusted (i.e. ε_R in equation (84)). For each convergence criterion, multiple simulations were executed for different plant designs. The average convergence time and resulting integral performance value for each convergence criterion were calculated. As the criterion is made smaller, the convergence time increases, but the resulting performance is improved, as seen in Figure 18. It is clear from Figure 18 that the RGP-based adaptation results in similar convergence to ES with tight convergence criteria. When the criteria are loosened, convergence times decrease and performance of the ES controller degrades, whereas RGP-based adaptive control maintains consistent performance under these looser convergence criteria (and correspondingly shorter convergence times).

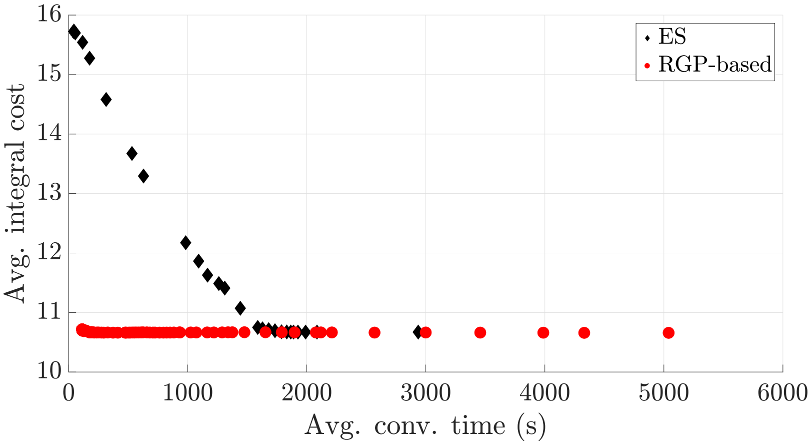

Figure 18: Pareto front for comparing performance of ES and RGP-based adaptation in 1D.

Figure 19 displays each of the components of the instantaneous performance index value while operating at a nominal controller parameter, an RGP-based optimized controller parameter, and ES optimized controller parameter. Since there is only a single (global and local) optimum for the stationary flight case (see Figure 17), the optimized controller parameters from ES and the RGP-based adaptation are extremely similar. Additionally, the performance values while operating at the optimized parameters are extremely close in value.

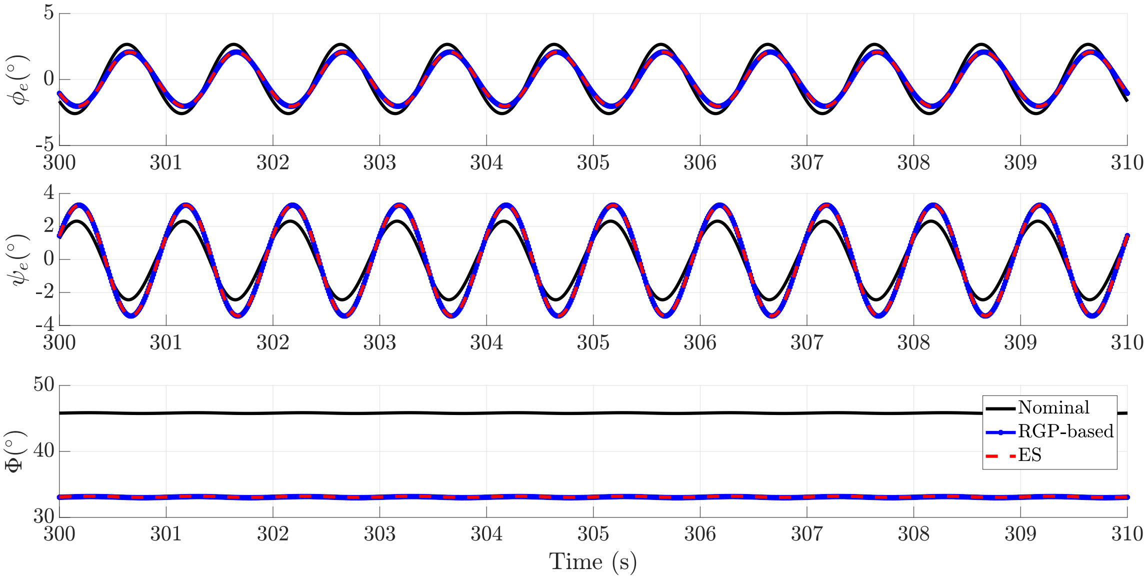

Figure 19: Comparison of controller parameter performance between a nominal design, design from the RGP-based adaption, and design from ES.

4.4.7.2 Crosswind flight controller optimization

The RGP-based adaptation was also applied to real-time crosswind flight optimization in simulation. This analysis demonstrates the RGP-based adaptation's efficiency in multidimensional control parameter design spaces. The crosswind flight controller for simulations (see Chapter 2.3.2.1) was augmented with the RGP-based adaptation in order to optimize the azimuth angle limit and sideslip angle setpoint in real time. Table 5 summarizes the decision variables and objective function used for this case study.

Although the design space exploration in the 1D case study simply moves back and forth between the extremes of the design space, the 2D exploration strategy, shown in Figure 20

Table 5: Summary of optimization problem features for simulation-based crosswind flight controller optimization

Feature	Symbol	Description
Control Parameters	Φ_{lim}	Azimuth angle limit
	β_{sp}	Sideslip angle setpoint
Instantaneous objective function	$\left(\frac{\lvert v_{app}\rvert}{\lvert v_w\rvert}\right)^3$	Instantaneous power factor
Integral objective function	$\frac{1}{\Delta t}\int_{t_c}^{t_c+\Delta t}\left(\frac{\lvert v_{app}\rvert}{\lvert v_w\rvert}\right)^3 dt$	Avg. power factor

is much more complex. In Figure 20, the variation in color from blue to yellow reflects the exploration of the design space in time (blue corresponds the initial time, whereas bright yellow corresponds to the final time of 3000 seconds). Figure 21 displays the (reduced) design space at selected times throughout the simulation, as well as contours that represent the RGP-C based performance index characterization across the reduced design space at the corresponding times. After 3000 seconds of the RGP-based adaptation, the design space has been significantly reduced to 5% of the original design space and consists of control parameters that all result in power augmentation of at least 40% through efficient crosswind motion.

Similar to the stationary flight (1D) comparison, a 2D comparison for the crosswind flight case was carried out between ES and the RGP-based adaptation. Again, the convergence criterion was adjusted for each algorithm. For each convergence criterion, multiple simulations were run for different system designs to calculate an average convergence time and integral performance (to be maximized in this case) across all system designs. A single point in either of the Pareto fronts seen in Figure 22 corresponds to a single convergence criterion.

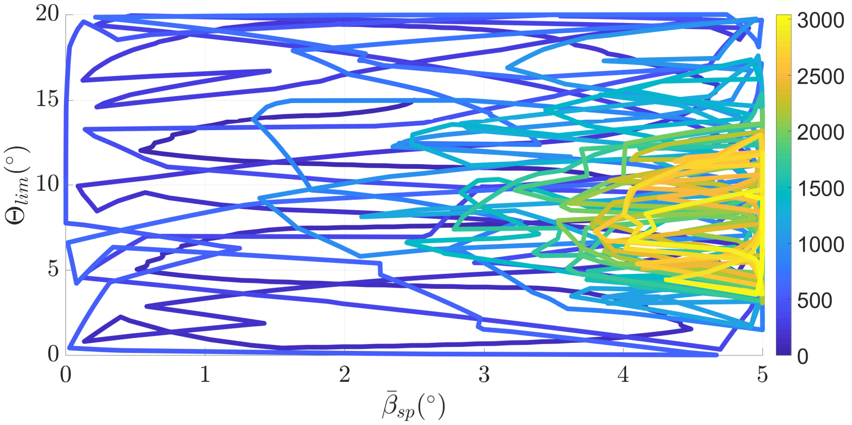

Figure 20: Design exploration for 2D controller design space, where the color represents the time at which each point was visited.

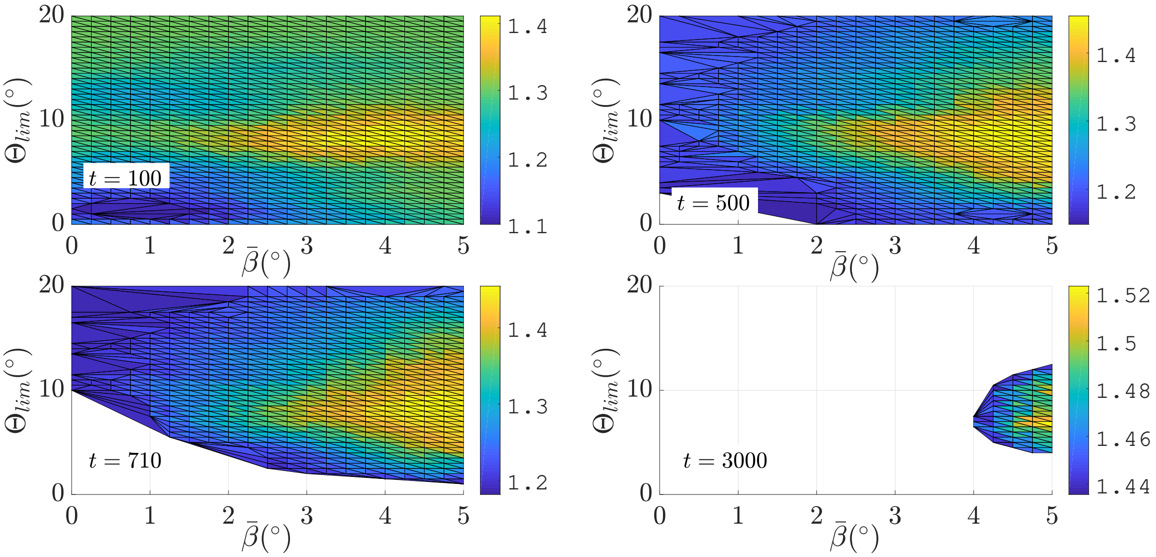

Figure 21: Design space and response surface characterization (contours) for crosswind flight case study at $t = 100$, $t = 500$, $t = 710$, and $t = 3000$ seconds.

Figure 23 displays the instantaneous performance index versus time for a nominal set of controller parameters, the controller parameters converged to by the RGP-based adaptation, and the controller parameters converged to by ES. In this case study, the RGP-based controller adaptation significantly outperforms the ES algorithm. This arises from the fact that the optimal control parameters in the crosswind flight case lie on the boundary of the stable design space. When using ES, a barrier function was required to ensure that the ES algorithm did not visit points within the unstable region; therefore, the performance associated with the optimal controller parameters from ES was significantly less than those of the RGP-based adaptation.

The flight path of the BAT while operating at the controller parameters from the nominal design, ES adaptation, and the RGP-based adaptation are displayed in Figure 24. Recall that the dimensions here are lab-scale dimensions. Additionally, the behavior of the roll and heading angle for each controller can be seen in Figure 25.

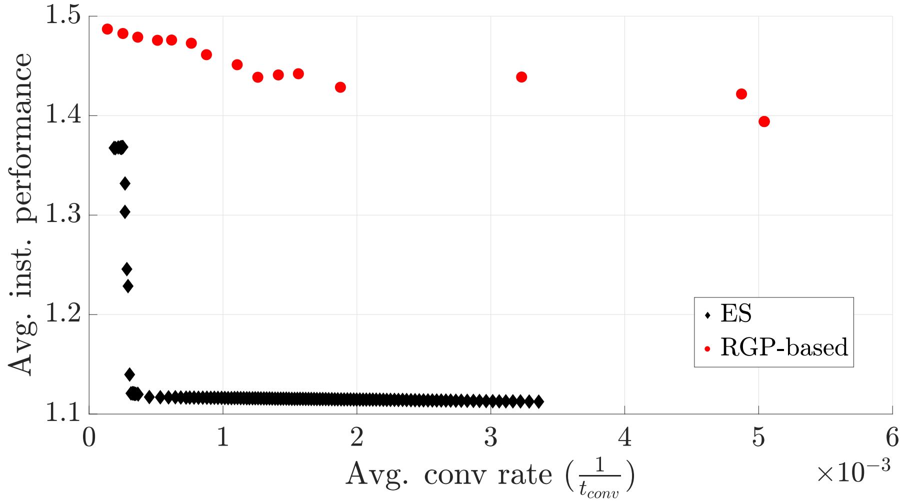

Figure 22: Pareto front for comparing performance of ES and RGP-based adaptation in 2D.

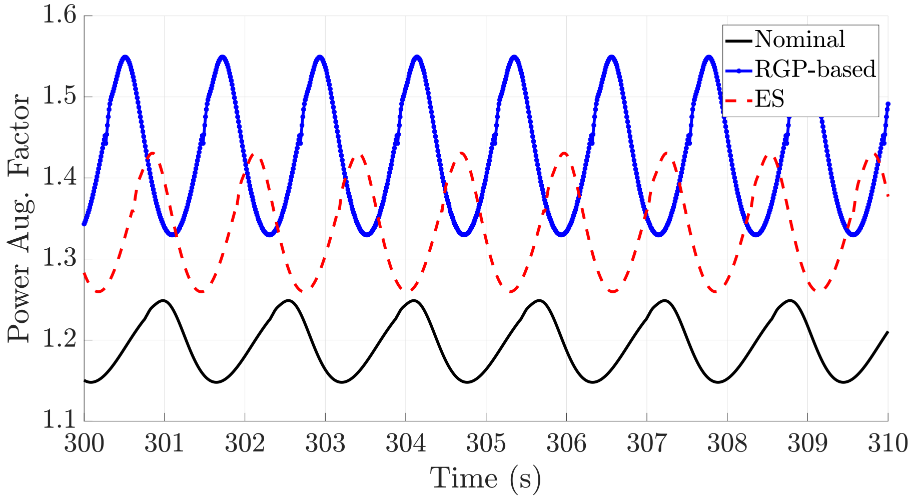

Figure 23: Comparison of instantaneous performance index values of controllers from the RGP-based and ES adaptations.

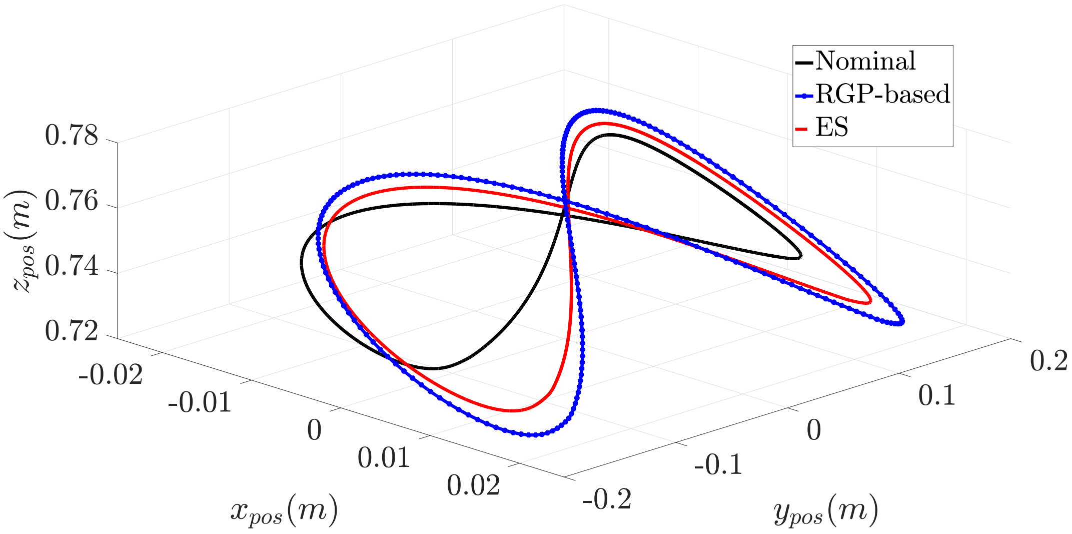

Figure 24: Comparison of crosswind flight paths between controller parameters from the nominal case, ES adaptation, and RGP-based adaptation.

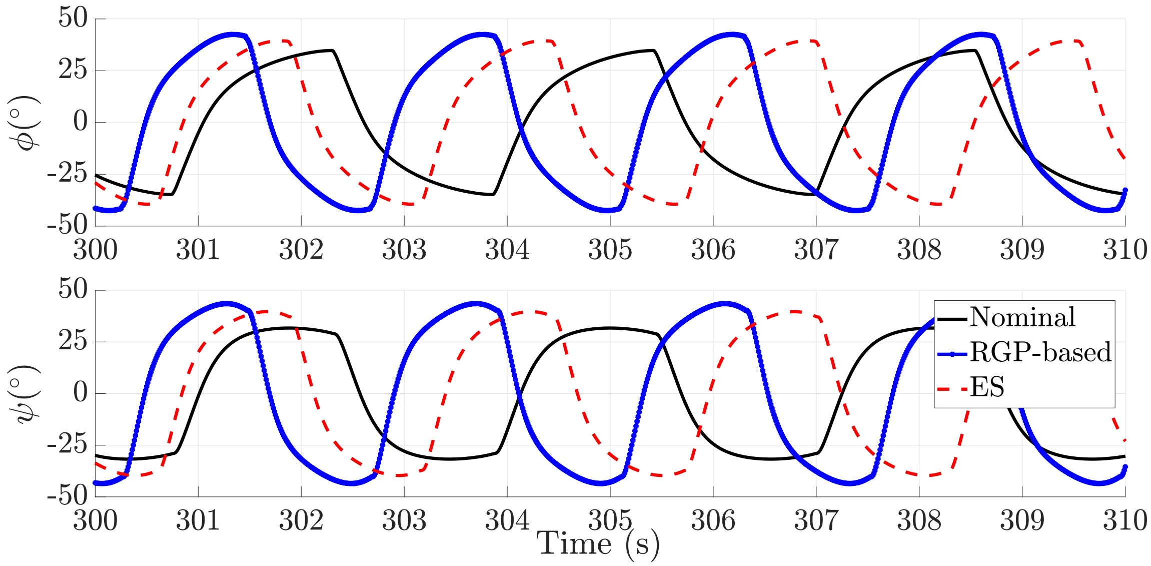

Figure 25: Comparison of roll and heading angle between controllers from the nominal case, ES adaptation, and RGP-based adaptation.

4.4.8 Experimental controller adaptation results: Crosswind flight optimization

By leveraging the experimental platform discussed in Chapter 2.2, the RGP-based adaptation was applied to the experimental crosswind flight optimization of the BAT. To induce crosswind flight experimentally, the controller detailed in Chapter 2.3.2.2 was used. The RGP-based adaptation seeks to optimize the square wave amplitude and period, which ultimately govern the geometric characteristics of the figure-8 flight path. Table 6 summarizes the decision variables and objective function used for this case study.

To demonstrate the efficient operation of the RGP-based adaptation, multiple experiments were conducted with various initial conditions for the controller design parameters. Table 7 provides a summary of the results for multiple initial conditions. It is clear that the adaptation converges to nearly the same design point, where small observed differences may be attributed to experimental noise. Figure 26 displays the control parameter design

Table 6: Summary of optimization problem features for experimental crosswind flight controller optimization

Feature	Symbol	Description
Control Parameters	a_{sq}	Square wave amplitude
	T_{sq}	Square wave period
Instantaneous objective function	$\left(\frac{\lvert v_{app}\rvert}{\lvert v_w\rvert}\right)^3$	Instantaneous power factor
Integral objective function	$\frac{1}{\Delta t}\int_{t_c}^{t_c+\Delta t}\left(\frac{\lvert v_{app}\rvert}{\lvert v_w\rvert}\right)^3 dt$	Avg. power factor

space exploration from the RGP-based adaptation. Approximately the first 100 seconds are used to initialize the experiment, which involves initializing the motion capture algorithm and crosswind flight at the prescribed initial condition. Once convergence is detected, around 1600 seconds in Figure 26, the adaptation is turned off and the design variables are set to the optimal values to calculate the average power factor over a prescribed time window. The effectiveness of crosswind flight is illustrated in Figure 27, which displays the lateral velocity of the BAT while operating at the optimized controller parameters. Figure 28 displays the mean function estimate from the RGP modeling over the reduced design space at multiple times throughout the experiment. By the time convergence is detected, the design space has been reduced to 12% of the original design space.

Table 7: Summary of experimental RGP-based adaptation results for multiple controller parameter initial conditions

Initial Condition		Opt. Control Design		Conv.	Avg.
$a_{sq,0}(^\circ)$	$T_{sq,0}(s)$	$a^*_{sq}(^\circ)$	$T^*_{sq}(s)$	Time(s)	PF
7	5	15	5.588	1717.75	1.67
10	8	15	5.588	1752.125	1.60
10	10	15	5.882	1590.35	1.60

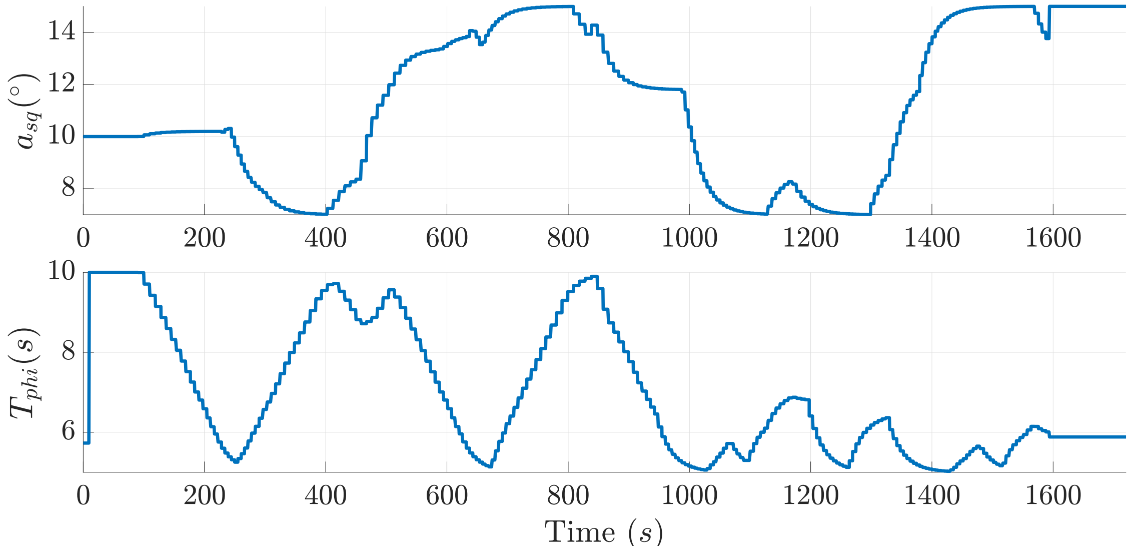

Figure 26: Experimental design space exploration for initial condition setting with $a_{sq,0} = 10^\circ$ and $T_{sq,0} = 10\ s$. Note the first 100 seconds is used to initialize the experiment.

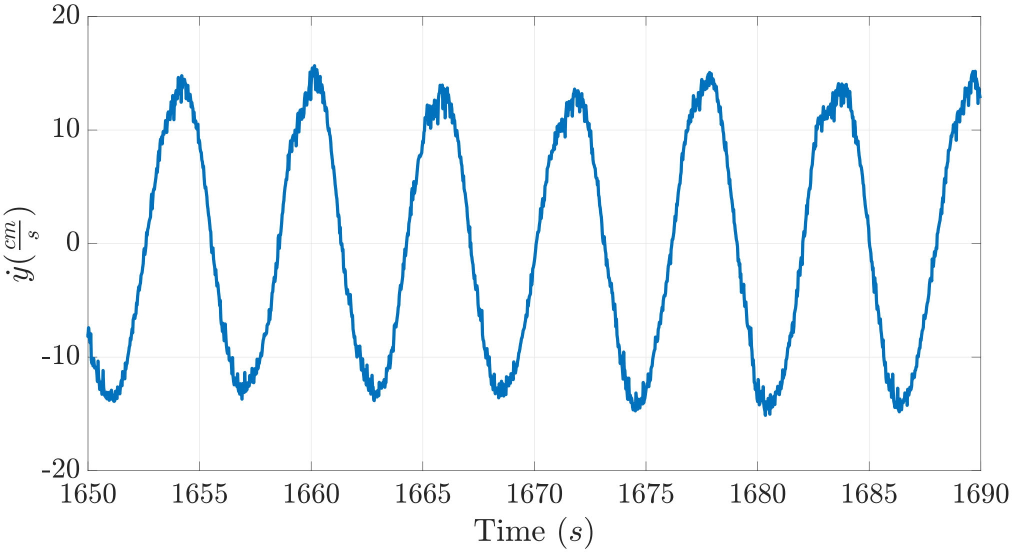

Figure 27: Lateral velocity of the BAT while operating at the optimized control parameters.

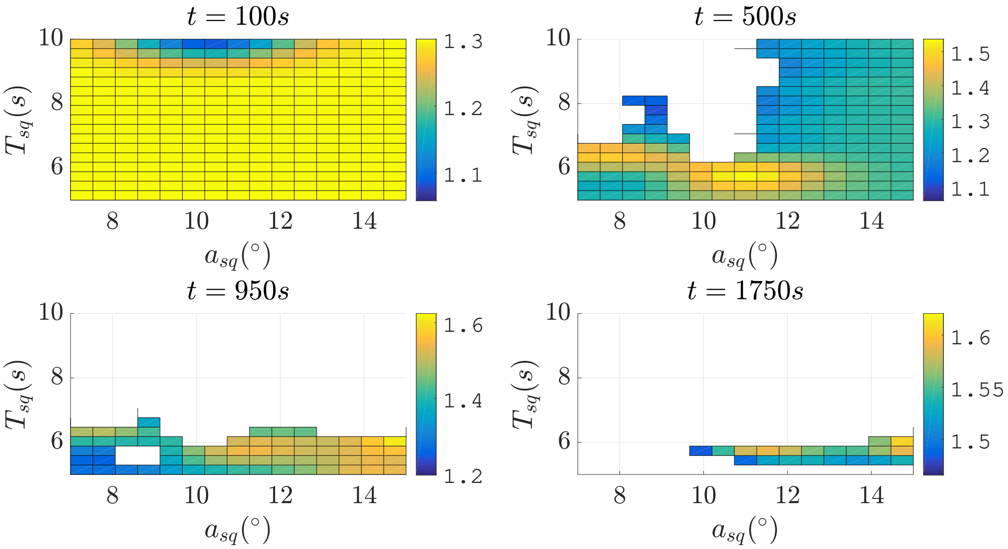

Figure 28: Design space and response surface characterization (contours) at $t = 100$, $t = 500$, $t = 950$, and $t = 1600$ seconds.

CHAPTER 5: SIMULATION-BASED COMBINED PLANT AND CONTROLLER OPTIMIZATION

Simulation-based studies were used to initially demonstrate the impact of nested co-design on complex system design. This chapter details each of the simulation-based studies that were used to progress toward the GP-driven, nested co-design framework. Simulation-based studies provided an initial mechanism for evolving the nested co-design process without having to overcome the challenges associated with experimental evaluation. A brief summary is provided to convey the unique features of each study. All of the simulation-based studies explored here use the dynamic model of the BAT detailed in Chapter 2.1.

5.1 Simulation-based stationary flight airborne wind energy system optimization

To evaluate the efficacy of the nested co-design framework for complex system design, stationary flight of the dynamic model presented in Chapter 2.1 was analyzed initially. Stationary flight is the common operational model for AWE with secondary objectives, such as telecommunications or surveillance, which require the system to remain motionless. Three proportional plus derivative controllers regulate the altitude, pitch angle, and roll angle to constant setpoints by adjusting tether length in order to keep the system motionless throughout operation. To capture the performance under this mode of operation, the quality of flight metric detailed in Chapter 2.5.1 was used. The plant parameters to be optimized are the longitudinal location of the center of mass and reference area scale factor, while the controller optimization only optimizes a single parameter, the pitch angle setpoint. A

Table 8: Summary of optimization problem features for simulation-based co-design of AWE system operating in stationary flight.

Feature	Symbol	Description
Plant Parameters	x_{cm}	Longitudinal location of center of mass
	K_A	Reference area scale factor
Control Parameters	θ_{sq}	Pitch angle setpoint
Instantaneous objective function	$k_1\Phi^2 + k_2\psi_e^2 + k_3\phi_e^2$	Instantaneous quality of flight
Integral objective function	$\int_{t_c}^{t_c+\Delta t}(k_1\Phi^2 + k_2\psi_e^2 + k_3\phi_e^2)dt$	Integral quality of flight

summary of the objective function and design variables is provided in Table 8.

5.1.1 Combined plant and controller optimization results using optimal design of experiments and extremum seeking

The tools employed on each level of the nested co-design framework play a critical role in the efficiency of the co-design process. Initially, the motivation was to transform our previous work in [38] from a simultaneous strategy to a nested strategy by leveraging a unique characteristic of controller parameters. Specifically, controller parameters are unique because these parameters can be optimized *during* an experiment, whereas a plant parameter can only be adjusted in between experiments. To evolve to a nested co-design framework, the outer loop of the nested strategy replicated the simultaneous strategy from [38] by using G-optimal DoE to generate a batch of candidate plant designs and design space reduction to exploit the perceived optimum. However, to optimize the controller parameters in real time on the inner loop, ES, a classic, non-model based adaptive control law, was used. The process diagram for the nested structure that utilized G-optimal DoE on the outer loop and ES on the inner loop can be seen in Figure 29.

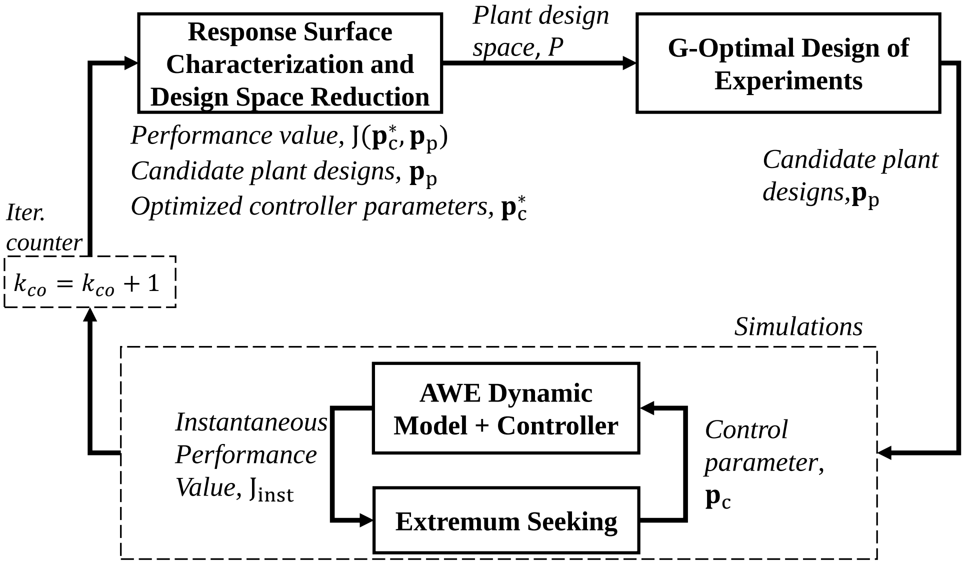
Response Surface Characterization and Design Space Reduction
Plant design space, P
G-Optimal Design of Experiments
Performance value, $J(\mathbf{p}_c^*, \mathbf{p}_p)$
Candidate plant designs, $\mathbf{p}_p$
Optimized controller parameters, $\mathbf{p}_c^*$
Candidate plant designs, $\mathbf{p}_p$
Iter. counter
$k_{co} = k_{co} + 1$
Simulations
AWE Dynamic Model + Controller
Instantaneous Performance Value, J_{inst}
Extremum Seeking
Control parameter, $\mathbf{p}_c$

Figure 29: Nested optimization framework utilizing optimal DoE on the outer loop and ES on the inner loop.

The results presented in this section were generated when using ES as the inner loop optimization strategy. Table 9 displays the perceived optimal plant design, the optimal controller parameter for the given plant design, and the percent reduction in the size of the design space following the corresponding iteration. The candidate plant design points, generated by G-optimal DoE at each iteration, are represented by the black diamonds in Figure 30. The contours in this figure correspond to the response surface characterization following the evaluation of each candidate plant design point at the corresponding iteration. The plant design space reduction following each iteration can be visualized in Figure 30.

Table 9: Optimal system design at each iteration when using ES for the adaptation law.

Iteration	K_A^*	x_{cm}^* (% chord)	$\theta_{sp}^*(^\circ)$	% Reduction
1	1.455	52.40	12.99	95.36
2	1.391	52.40	13.10	96.49
3	1.406	52.40	13.08	96.49
4	1.435	52.40	13.03	98.38

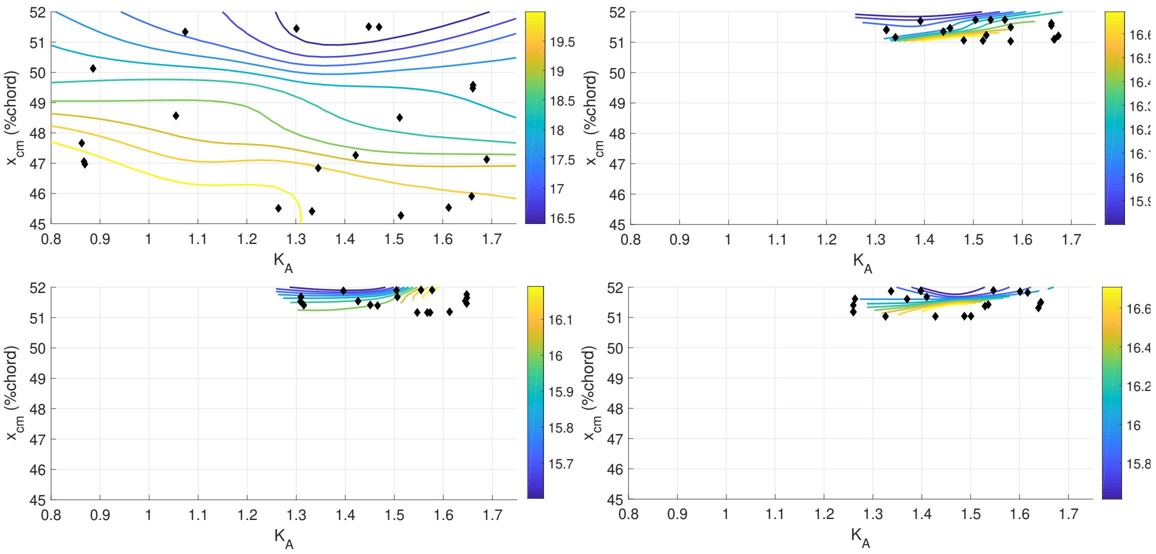

Figure 30: Response surface characterization at 1^{st} (top left), 2^{nd} (top right), 3^{rd} (bottom left), and 4^{th} (bottom right) iterations with candidate design points overlaid, when ES for the adaptation law in the nested co-design strategy.

5.1.2 Combined plant and controller optimization results using optimal design of experiments and entropy-based DoE

The entropy-based DoE adaptation was used as the inner loop optimization strategy to form a nested strategy that relied on the same mathematical machinery at both the inner and outer loops. The process diagram for the nested structure that utilized G-optimal DoE on the outer loop and entropy-based DoE on the inner loop can be seen in Figure 31.

The results presented in this section were generated when using the nested framework shown in Figure 31. Figure 32 illustrates the design space exploration and convergence of the entropy-based DoE adaptation. Table 10 displays the perceived optimal plant design, the optimal controller parameter for the given plant design, and the percent reduction in the size of the plant design space following the corresponding iteration. The candidate plant design points, generated by G-optimal DoE at each iteration, are represented by the

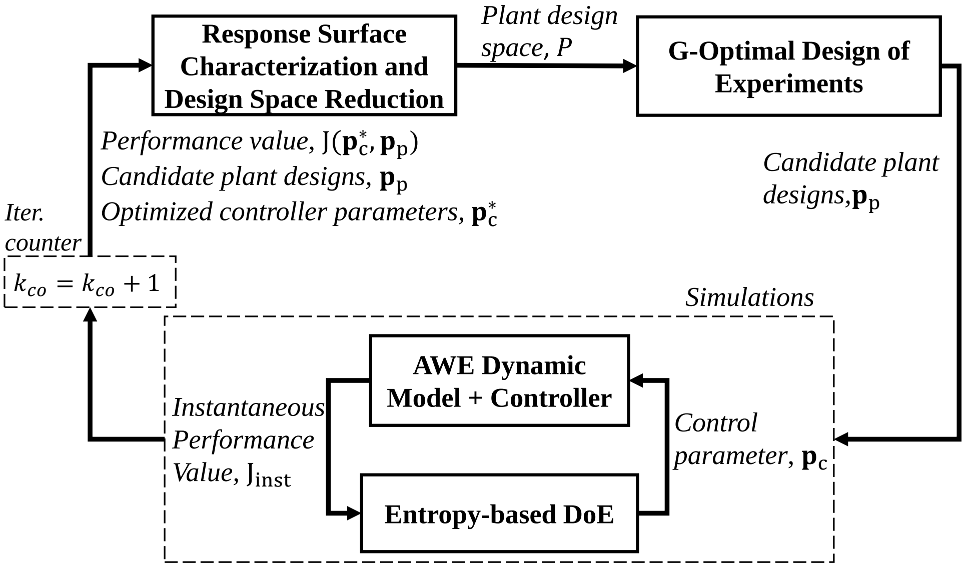
Response Surface Characterization and Design Space Reduction
Plant design space, P
G-Optimal Design of Experiments
Performance value, J($\mathbf{p}_c^*$, $\mathbf{p}_p$)
Candidate plant designs, $\mathbf{p}_p$
Optimized controller parameters, $\mathbf{p}_c^*$
Candidate plant designs, $\mathbf{p}_p$
Iter. counter
$k_{co} = k_{co} + 1$
Simulations
AWE Dynamic Model + Controller
Instantaneous Performance Value, J_{inst}
Control parameter, $\mathbf{p}_c$
Entropy-based DoE

Figure 31: Nested optimization framework utilizing optimal DoE on the outer loop and entropy-based DoE on the inner loop.

black diamonds in Figure 33. The contours in this figure correspond to the response surface characterization following the evaluation of each candidate plant design point. The plant design space reduction following each iteration can be visualized in Figure 33.

Table 10: Optimal system design at each iteration when using entropy-based DoE for the adaptation law.

Iteration	K_A^*	x_{cm}^* (% chord)	$\theta_{sp}^*(^\circ)$	% Reduction
1	1.160	52.4	13.7	75.3
2	1.212	52.4	13.3	81.7
3	1.750	52.4	14.3	99.0
4	1.750	52.4	14.3	99.8

5.1.3 GP-driven, nested co-design framework

A GP-driven, nested co-design framework was established by utilizing batch Bayesian optimization for the outer loop plant iteration and the RGP-based adaptation (detailed in Chapter 4.4.1) for the inner loop controller optimization. There is a close mathematical

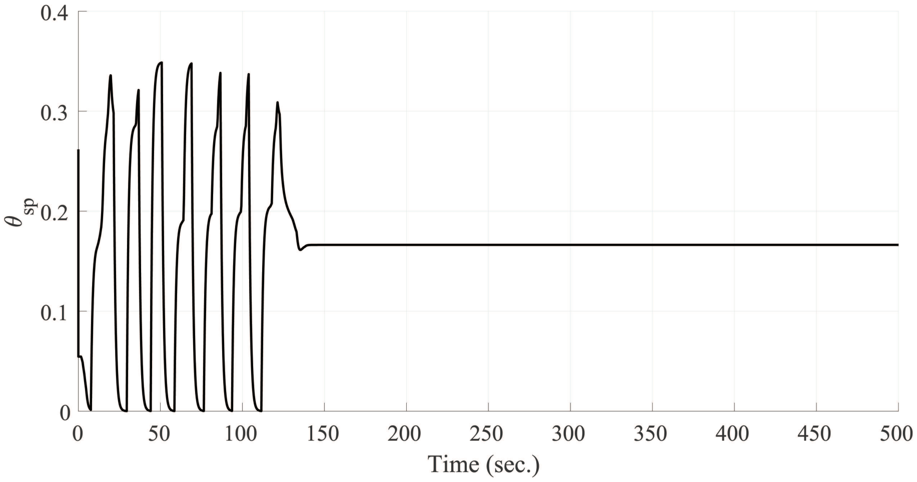

Figure 32: Convergence for entropy-based DoE algorithm for a sample plant design ($x_{cm} = 45.4\%$ and $K_A = 1$).

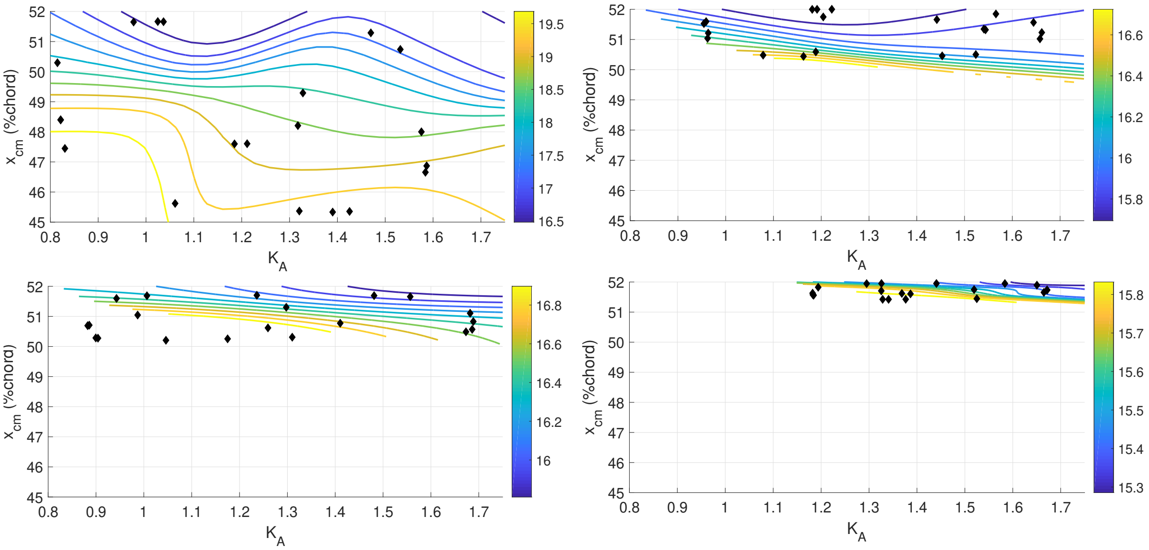

Figure 33: Response surface characterization at 1st (top left), 2nd (top right), 3rd (bottom left), and 4th (bottom right) iterations with candidate design points overlaid when entropy-based DoE for the adaptation law in the nested co-design strategy.

relationship between batch Bayesian optimization and the RGP-based adaptation. Specifically, the RGP-based adaptation is essentially the continuous-time version of the batch Bayesian optimization outer loop. The process diagram for the GP-driven, nested co-design framework that employs batch Bayesian optimization on the outer loop and the RGP-based adaptation on the inner loop can be seen in Figure 34. It is important to note that the amplitude of the sinusoidal wind perturbation was reduced by 50% for these simulation results, which results in a much lower integral performance index.

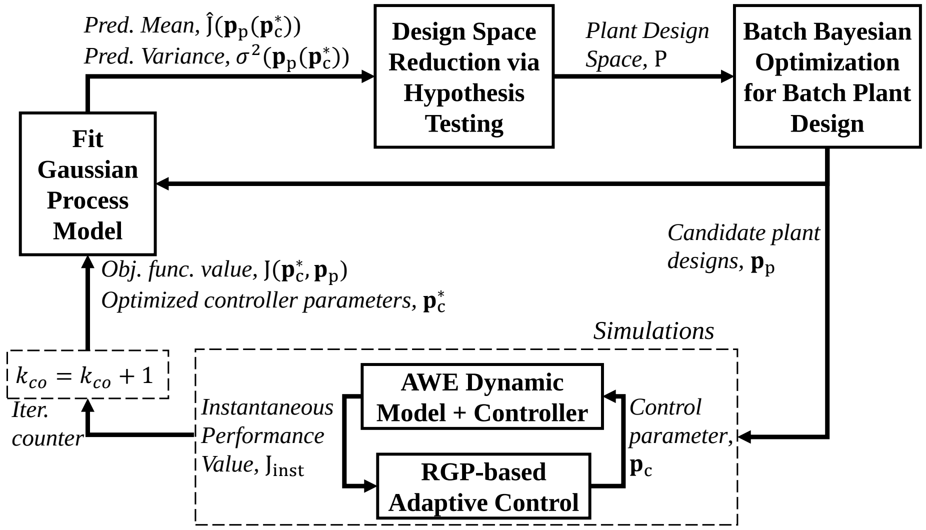
Pred. Mean, $\hat{J}(\mathbf{p}_p(\mathbf{p}_c^*))$
Pred. Variance, $\sigma^2(\mathbf{p}_p(\mathbf{p}_c^*))$
Design Space Reduction via Hypothesis Testing
Plant Design Space, P
Batch Bayesian Optimization for Batch Plant Design
Fit Gaussian Process Model
Candidate plant designs, $\mathbf{p}_p$
Obj. func. value, $J(\mathbf{p}_c^*, \mathbf{p}_p)$
Optimized controller parameters, $\mathbf{p}_c^*$
Simulations
$k_{co} = k_{co} + 1$
Iter. counter
Instantaneous Performance Value, J_{inst}
AWE Dynamic Model + Controller
RGP-based Adaptive Control
Control parameter, $\mathbf{p}_c$

Figure 34: Nested optimization framework utilizing batch Bayesian optimization and RGP-based adaptive control (both GP-based tools) on the outer and inner loops, respectively.

Table 11 displays the perceived optimal plant design, the optimal controller parameter for the given plant design, and the percent reduction in the size of the plant design space following the corresponding iteration. After completing four iterations of the co-design process, the candidate plant design space was reduced by 98%. The red circles in Figure 35 represent the candidate design points evaluated during the co-design process. The contours in this figure correspond to the response surface characterization following the evaluation of each candidate plant design point. The plant design space reduction following each iteration can be visualized in Figure 35. After significant design space reduction, batch Bayesian optimization begins to populate the plant design space with repetitive design points.

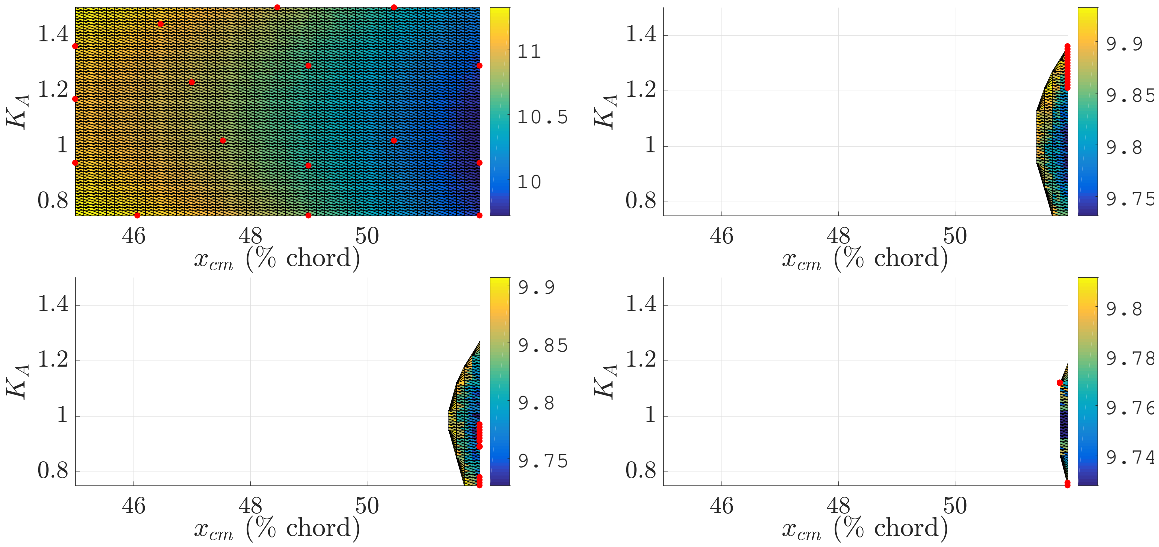

Figure 35: Response surface characterization over the reduced design space the at 1^{st} (top left), 2^{nd} (top right), 3^{rd} (bottom left), and 4^{th} (bottom right) iterations with candidate design points tested at each iteration overlaid for the GP-driven, nested co-design strategy.

Table 11: Optimal system design at each iteration when using GP-driven, nested co-design

Iteration	K_A^*	x_{cm}^* (% chord)	$\hat{J}(\mathbf{p}_c^*(\mathbf{p}_p), \mathbf{p}_p)$	% Reduction
1	1	51.93	9.724	0
2	0.98	51.93	9.731	94.3
3	0.97	51.93	9.726	95.5
4	0.97	51.93	9.728	98.2

5.2 Simulation-based crosswind flight optimization using GP-driven nested co-design framework

The final simulation-based case study focused on optimizing the plant and controller parameters for crosswind flight. For this simulation-based analysis, the controller from

Chapter 2.3.2.1 was used to achieve crosswind flight. In addition to representing an important flight mode, the crosswind flight investigation allowed us to consider both plant and controller design spaces with multiple dimensions. The process diagram for the GP-driven, nested co-design framework can be seen in Figure 34.

Recall that the goal of crosswind flight is to increase the apparent wind presented to the onboard turbines by moving in figure-8 flight patterns. For the crosswind flight controller used in simulation, the two key parameters that control the shape of the figure-8 flight path are the azimuth angle limit Φ_{lim} and sideslip angle setpoint β_{sp}. The azimuth angle limit dictates the width of the crosswind flight pattern, while the sideslip angle setpoint is known to significantly influence the aerodynamic efficiency of crosswind flight. Because the domain of stabilizing Θ_{lim} and β_{sp} is not rectangular, a new variable, $\bar{\beta}_{sp}$ is defined, such that the resulting design space is rectangular in Θ_{lim} and $\bar{\beta}_{sp}$:

$$\beta_{sp} = \bar{\beta}_{sp}(1 - 2.29\Theta_{lim}), \tag{91}$$

where $\bar{\beta}_{sp} \in [0,5]^\circ$ and $\Theta_{lim} \in [0,20]^\circ$. These parameters are selected as the controller parameters in the inner loop optimization of the co-design framework. The plant parameters are selected to be the longitudinal location of center of mass (x_{cm}) and reference area scale factor (K_A).

Table 13 displays the perceived optimal plant design, the expected performance associated with the optimal plant design while operating at the optimal controller design, and the percent reduction in the size of the plant design space following the corresponding iteration. The red circles in Figure 36 represent the candidate design points evaluated during the

Table 12: Summary of optimization problem features for simulation co-design of AWE during crosswind flight.

Feature	Symbol	Description
Plant Parameters	$K_{A,vert}$	Vert. stabilizer reference area scale factor
	$K_{A,horz}$	Horiz. stabilizer reference area scale factor
Control Parameters	Φ_{lim}	Azimuth angle limit
	β_{sp}	Sideslip angle setpoint
Instantaneous objective function	$\left(\frac{\lvert v_{app}\rvert}{\lvert v_w\rvert}\right)^3$	Instantaneous power factor
Integral objective function	$\frac{1}{\Delta t}\int_{t_c}^{t_c+\Delta t}\left(\frac{\lvert v_{app}\rvert}{\lvert v_w\rvert}\right)^3 dt$	Avg. power factor

co-design process. The contours in this figure correspond to the response surface characterization following the evaluation of each candidate plant design point. The plant design space reduction following each iteration can be visualized in Figure 36. Following four complete iterations of the co-design process, the candidate plant design space was reduced by nearly 99%.

Table 13: Optimal plant design at each iteration when using the GP-based, nested co-design framework.

Iteration	$K^*_{A,vert}$	$K^*_{A,horz}$	$\hat{J}(\mathbf{p}_c(\mathbf{p}_p),\mathbf{p}_p)$	% Reduction
1	1.1	1.06	1.526	53.18
2	1.1	1.08	1.522	93.10
3	1.1	1.1	1.506	97.62
4	1.1	1.1	1.509	98.99

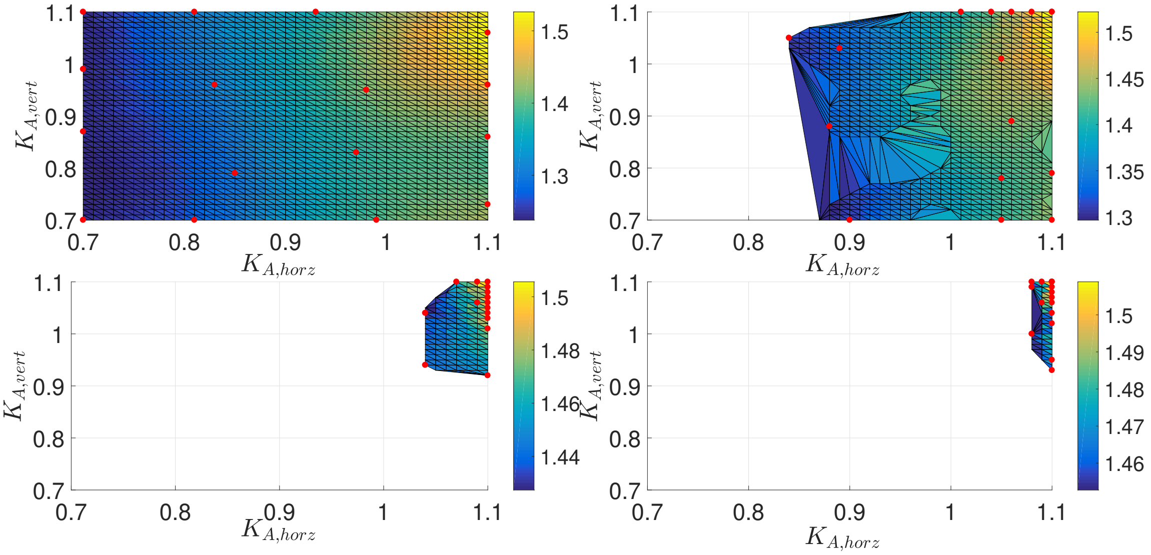

Figure 36: Plant-level response surface characterization at 1st (top left), 2nd (top right), 3rd (bottom left), and 4th (bottom right) iterations with candidate design points overlaid for the GP-driven, nested co-design strategy.

CHAPTER 6: EXPERIMENTAL COMBINED PLANT AND CONTROLLER OPTIMIZATION

The most valuable use cases of the co-design tools developed in this work are those in which expensive and time-consuming experiments are part of the co-design process. In these scenarios, rapid identification of optimal control parameters *during* the course of an experiment, using the adaptation tools developed in this work, can dramatically decrease the economic cost of carrying out a co-design process. To illustrate the efficacy of the developed co-design tools in an experimental regime, this chapter focuses on the use of a lab-scale experimental platform in optimizing both the physical design parameters and flight control parameters for an AWE system that executes power-augmenting crosswind flight.

To optimize the controller and physical system of an AWE system operating in crosswind flight, the GP-driven, nested strategy discussed in Chapter 5.2 is leveraged. The process for the experimental co-design framework is detailed in Figure 37. At the outer loop of the nested strategy, batch Bayesian optimization is used to populate a batch of candidate plant designs. During the experiment, the RGP-based adaptation, detailed in Chapter 4.4, optimizes the control design in *real time* for each candidate plant design.

As discussed in Chapter 2.5.2, the objective of crosswind flight is to increase the apparent wind velocity presented to the onboard turbine by moving perpendicular to the wind. To quantitatively capture how much more energy the system could produce by moving in a

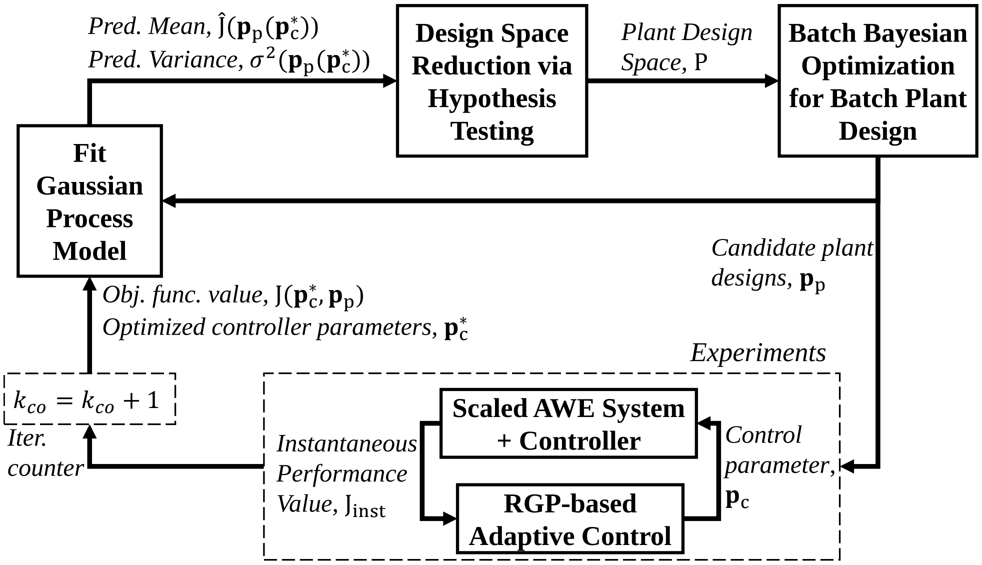
Pred. Mean, $\hat{J}(\mathbf{p}_p(\mathbf{p}_c^*))$
Pred. Variance, $\sigma^2(\mathbf{p}_p(\mathbf{p}_c^*))$
Design Space Reduction via Hypothesis Testing
Plant Design Space, P
Batch Bayesian Optimization for Batch Plant Design
Fit Gaussian Process Model
Candidate plant designs, $\mathbf{p}_p$
Obj. func. value, $J(\mathbf{p}_c^*, \mathbf{p}_p)$
Optimized controller parameters, $\mathbf{p}_c^*$
Experiments
$k_{co} = k_{co} + 1$
Iter. counter
Instantaneous Performance Value, J_{inst}
Scaled AWE System + Controller
RGP-based Adaptive Control
Control parameter, $\mathbf{p}_c$

Figure 37: Process diagram for experimental nested co-design framework.

crosswind motion, the power factor objective function from Chapter 2.5.2 was used. The plant parameters to be optimized for the analysis in this chapter are the net buoyancy of the system and longitudinal location of the center of mass (x_{cm}) because these parameters have a dramatic impact on the performance of crosswind flight. Recall from Chapter 2.3.2.2, that to induce crosswind flight experimentally, the roll angle is commanded to follow a square wave defined by a square wave amplitude (a_{sq}) and period (T_{sq}). The roll angle setpoint and period govern the geometric characteristics of the crosswind flight path. Therefore, the controller parameters to be optimized are the square wave amplitude and period. The key features of the optimization are summarized in the Table 14.

6.1 Results: Gaussian process-based nested combined plant and controller optimization

The results presented here extend the controller optimization results from Chapter 4.4.8 by placing the adaptation in the inner loop of a nested optimization framework. At each iteration of the GP-driven, nested co-design framework, sixteen plant design points were

Table 14: Summary of optimization problem features for experimental co-design.

Feature	Symbol	Description
Plant Parameters	x_{cm}	Longitudinal location of center of mass
	ρ_{Buo}	Net Buoyancy
Control Parameters	a_{sq}	Square wave amplitude
	T_{sq}	Square wave period
Instantaneous objective function	$\left(\frac{\lvert v_{app}\rvert}{\lvert v_w\rvert}\right)^3$	Instantaneous power factor
Integral objective function	$\frac{1}{\Delta t}\int_{t_c}^{t_c+\Delta t}\left(\frac{\lvert v_{app}\rvert}{\lvert v_w\rvert}\right)^3 dt$	Avg. power factor

evaluated experimentally, where the RGP-based adaptation optimized the control parameters in real time for each of the 16 candidate plant designs. Figure 38 displays the iterative evolution of the response surface over the candidate design space. The top left plot shows the response surface over the reduced design space after evaluating the batch of candidate design points at the first iteration. The top right plot shows the response surface over the reduced design space following the second iteration and candidate design points to be tested at the third iteration. Similarly, the bottom left displays the response surface over the reduced design space following the third iteration and candidate design points to be tested at the fourth iteration and so on. The design space reduction following the fourth iteration results in three distinct “islands”. The island in the lower left corner contains the perceived optimum, while the island in the middle section contains points whose performance indices fall within 1% of the performance at the perceived optimum. The island in the upper left corner was nearly rejected, but the error bars in that area were just large enough to remain in the design space for subsequent iterations. This result shows the potential of the GP-driven, nested co-design strategy to handle highly nonlinear functions that may contain many local optima.

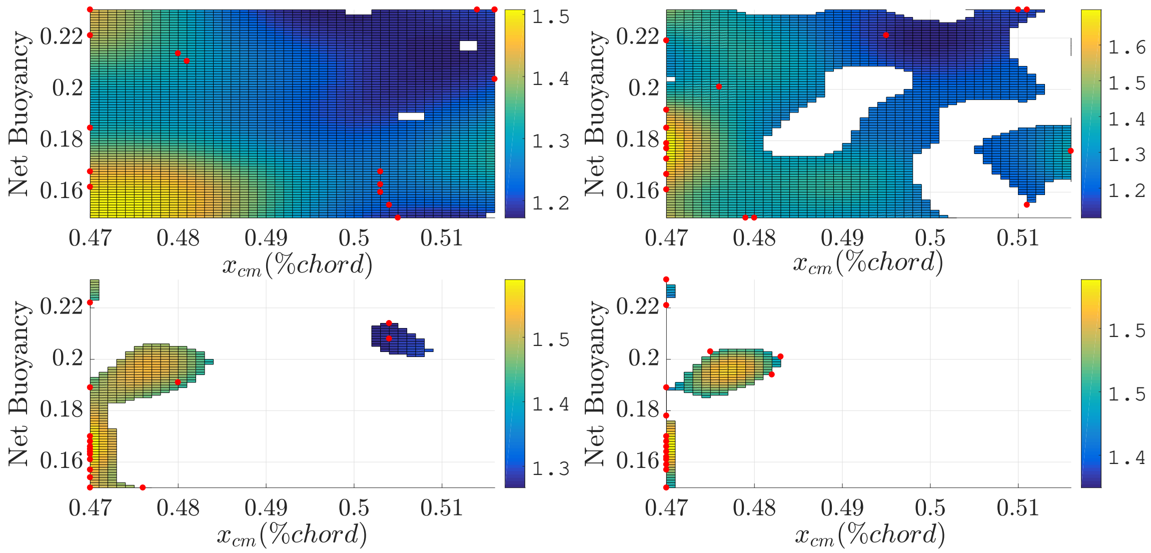

Figure 38: Response surface characterization over the reduced design space following the 1^{st} (top left), 2^{nd} (top right), 3^{rd} (bottom left), and 4^{th} (bottom right) iterations for the GP-driven, nested co-design framework. The candidate design points for the following iteration are overlaid on the response surface for the top right, bottom left, and bottom right plots.

Table 15: Optimal system design at each iteration of GP-driven, nested co-design

Iteration	$x^*_{cm}(\%)$	$\rho^*_{Buo}(\%)$	$\hat{J}(\mathbf{p}^*_c(\mathbf{p}_p), \mathbf{p}^*_p)$	% reduction
1	47.3	15.4	1.508	0.34
2	47	17.7	1.699	24.73
3	47	16.5	1.589	87.62
4	47	16.5	1.588	93.25

Table 15 displays the perceived optimal plant design at each iteration of the co-design process. The last column of this table presents the total reduction of the design space of the plant design space (P). Following four iterations of the co-design process, the original plant design space was reduced by 93.25%.

After completing four iterations of the co-design process, the percent difference in performance across the reduced design space was less than 10%. The co-design framework was considered to be converged after this point, and the optimal point was selected to be the

Table 16: Comparison of optimal system design to suboptimal system designs.

Description	$x_{cm}(\%)$	$\rho_{Buo}(\%)$	$a_{sq}(^{\circ})$	$T_{sq}(s)$	$J(\mathbf{p}_c(\mathbf{p}_p),\mathbf{p}_p)$
Optimal system design	47	16.5	13.93	7.647	1.63
Optimal plant design with suboptimal controller design	47	16.5	10	10	1.199
Suboptimal plant design with optimal controller design	51.6	23.1	15	5	1.1629
Suboptimal system design	51.6	23.1	10	7	1.093

point that maximized the mean function. Table 16 compares the perceived optimal design after four iterations to other suboptimal system designs. From the last column of this table, it is clear that optimizing the plant and controller parameters in a co-design framework is critical to maximizing system performance. Figure 39 compares the instantaneous performance index for each system design in Table 16. It is clear from Figure 39 that optimizing either the plant or controller in isolation does not maximize performance. One property of the system that is indicative of efficient crosswind flight is the lateral velocity of the system. Figure 40 displays the lateral velocity of the combined optimal system design, the optimal plant design operating at a suboptimal controller design, and a suboptimal design in the space. Compared with the selected suboptimal design, crosswind flight speeds are more than an order of magnitude greater than those of optimized flight. Even compared with an optimized plant but suboptimal controller, the crosswind flight velocity is doubled with the fully optimized design. As seen in Table 16, this ultimately results in 50 percent greater power output

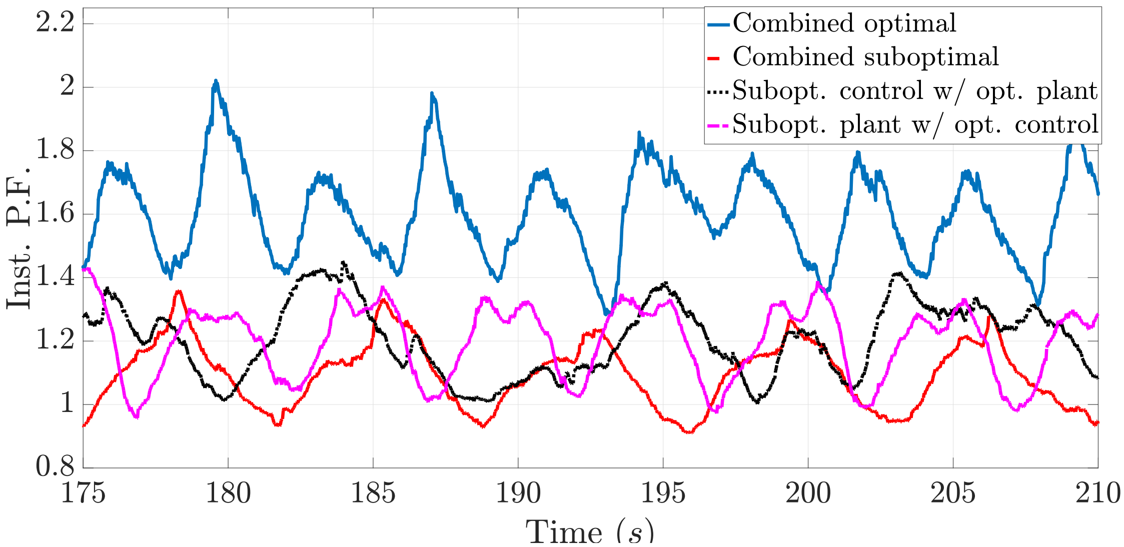

Figure 39: Comparison of instantaneous power factor for the combined optimal system design, the optimal plant design operating at a suboptimal controller design, a suboptimal plant design operating at a optimal controller design, and a suboptimal design in the space.

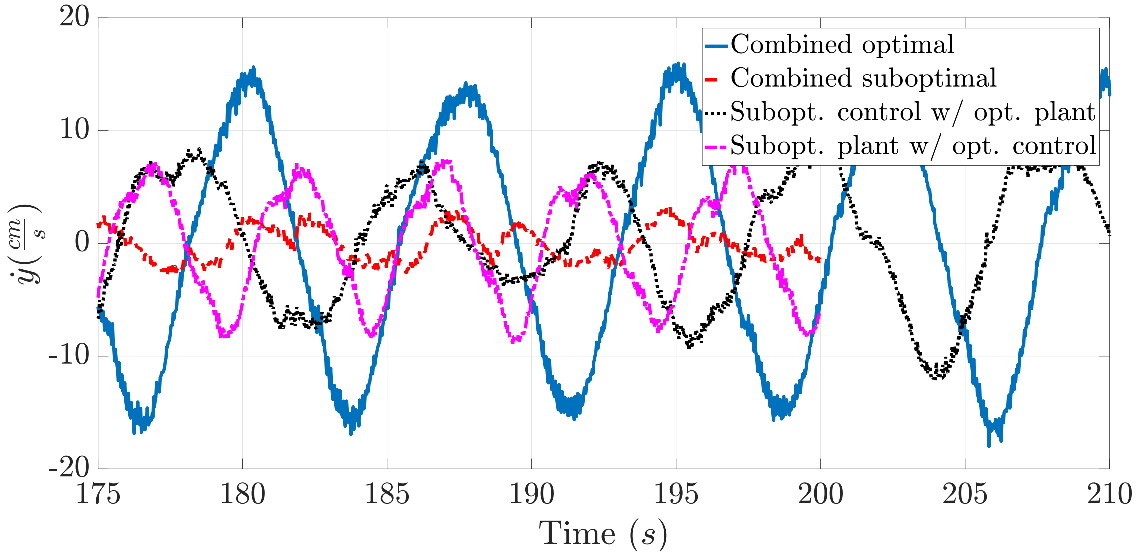

Figure 40: Comparison of crosswind flight velocity for the combined optimal system design, the optimal plant design operating at a suboptimal controller design, a suboptimal plant design operating at a optimal controller design, and a suboptimal design in the space.

CHAPTER 7: CONCLUSIONS AND FINAL REMARKS

A nested co-design framework that leverages the ability to optimize controller parameters during a simulation or experiment has been presented. By optimizing the controller parameters in *real time*, the time required to optimize a complex system can be reduced. The nested co-design framework presented here is unique relative to legacy co-design strategies in its ability to be applied to complex system designs that require either extensive simulations or experiments. Through the use of simulation-based and experimental AWE case studies, the efficacy of the nested co-design framework in converging to optimal design parameters was demonstrated.

REFERENCES

[1] H. K. Fathy, P. Y. Papalambros, A. G. Ulsoy, and D. Hrovat. Nested Plant/Controller Optimization with Application to Combined Passive/Active automotive suspensions. *Proceedings of the American Control Conference*, 2003. Denver, CO.

[2] H. K. Fathy, P. Y. Papalambros, and A. G. Ulsoy. Integrated Plant, Observer, and Controller Optimization With Application to Combined Passive/Active Automotive Suspensions. *Proceedings of ASME International Mechanical Engineering Congress and Exposition, Dynamic Systems and Control*, 2003.

[3] J. T. Allison, T. Guo, and Z. Han. Co-Design of an Active Suspension Using Simultaneous Dynamic Optimization. *Journal of Mechanical Design*, 136(8):081003/1–081003/14, 2014.

[4] H. K. Fathy, S. Bortoff, S. Copeland, P. Y. Papalambros, and A. G. Ulsoy. Nested Optimization of an Elevator and its Gain-Scheduled LQG Controller. *Proceedings of ASME International Mechanical Engineering Congress and Exposition, Dynamic Systems and Control*, pages 119–126, 2002. New Orleans, LA.

[5] M. J. Alexander, J. T. Allison, and P. Y. Papalambros. Decomposition-based Design Optimization of Electric Vehicle Powertrains Using Proper Orthogonal Decomposition. *International Journal of Powertrains*, 1(1), 2012.

[6] J. Deese, T. Muyimbwa, N. Deodhar, C. Vermillion, and P. Tkacik. Lab-Scale Characterization of a Lighter-Than-Air Wind Energy System - Closing the Loop. *Proceedings of AIAA Lighter-Than-Air Systems Conference*, 2015.

[7] N. Deodhar, C. Vermillion, and P. Tkacik. A Case Study in Experimentally-Infused Plant and Controller Optimization for Airborne Wind Energy Systems. *Proceedings of the American Control Conference*, 2015. Chicago, IL.

[8] H. K. Fathy, J. A. Reyer, P. Y. Papalambros, and A. G. Ulsov. On the Coupling Between the Plant and Controller Optimization Problems. *Proceedings of the American Control Conference*, 2001. Arlington, VA.

[9] K. Youcef-Toumi. Modeling, Design, and Control Integration: A Necessary Step in Mechatronics. *IEEE/ASME Transactions on Mechatronics*, 1:29–38, 1996.

[10] J.A. Reyer and P.Y. Papalambros. Optimal Design and Control of an Electric DC Motor. *Proceedings of ASME Design Engineering Technical Conferences*, 1999.

[11] D. Peters, P. Y. Papalambros, and A. Ulsoy. Control Proxy Functions for Sequential Design and Control Optimization. *Mechatronics*, 23(4):409–418, 2013.

[12] J. T. Allison and S. Nazari. Combined Plant and Controller Design using Decomposition-based Design Optimization and the Minimum Principle. *Proceedings of International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2010. Montreal, Quebec, Canada.

[13] J. Deese 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. *Proceedings of World Congress of the International Federation of Automatic Control*, 2017. Toulouse, FR.

[14] A. Baheri, J. Deese, and C. Vermillion. Combined Plant and Controller Design Using Bayesian Optimization: A Case Study in Airborne Wind Energy Systems. *Proceedings of ASME Dynamic Systems and Control Conference*, 2017.

[15] 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. *Journal of Dynamic Systems, Measurement, and Control*, 140(12), 08 2018.

[16] N. Deodhar and C. Vermillion. Fused Experimental/Numerical Plant and Control System Optimization Using Iterative G-Optimal Design of Experiments. *Proceedings of International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2016. Charlotte, NC.

[17] S. Kumar, A. Mohammadi, N. Gans., and M. Krstic. Automatic Tuning of Virtual Constraint-Based Control Algorithms for Powered Knee-Ankle Prostheses. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Aug 2017.

[18] D. Quintero, D. J. Villarreal, D. J. Lambert, S. Kapp, and R. D. Gregg. Continuous-Phase Control of a Powered Knee–Ankle Prosthesis: Amputee Experiments Across Speeds and Inclines. *IEEE Transactions on Robotics*, 34(3):686–701, June 2018.

[19] A. Buzy, S. Grall, and P. Cotte. A New Generation of Online Adaptive Design of Experiments: Application to Emissions Development and Full Load Calibration. In *Design of Experiments in Engine Development III*, May 2007.

[20] H. Haberland, H. Tschoke, and L. Schulze. Sensitivity Analysis of a Common Rail Injection System. In *Design of Experiments in Engine Development III*, May 2007.

[21] C. Vermillion. Altitude and Crosswind Motion Control for Optimal Power-Point Tracking in Tethered Wind Energy Systems With Airborne Power Generation. In *ASME 2013 Dynamic Systems and Control Conference*, October 2013.

[22] A. Bafandeh and C. Vermillion. Real-time Altitude Optimization of Airborne Wind Energy Systems using Lyapunov-based Switched ES Control. *Proceedings of the American Control Conference*, 2016. Boston, MA.

[23] 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.

[24] P. Williams, B. Lansdorp, and W. Ockesl. Optimal Crosswind Towing and Power Generation with Tethered Kites. *Journal of Guidance, Control, and Dynamics*, 31(1):81–93, January 2008.

[25] A. U. Zgraggen, L. Fagiano, and M. Morari. On Real-time Optimization of Airborne Wind Energy Generators. In *52nd IEEE Conference on Decision and Control*, Dec 2013.

[26] L. Fagiano, A. U. Zgraggen, M. Khammash, and M. Morari. Automatic Control of Tethered Wings for Airborne Wind Energy: Design and Experimental Results. *Proceedings of European Control Conference*, 2013. Zurich, Switzerland.

[27] L. Fagiano, A. Zgraggen, M. Morari, and M. Khammash. Automatic Crosswind Flight of Tethered Wings for Airborne Wind Energy: Modeling, Control Design, and Experimental Results. *IEEE Transactions on Control Systems Technology*, 22(4):1433–1447, July 2014.

[28] A. U. Zgraggen, L. Fagiano, and M. Morari. Real-Time Optimization and Adaptation of the Crosswind Flight of Tethered Wings for Airborne Wind Energy. *IEEE Transactions on Control Systems Technology*, 23(2):434–448, March 2015.

[29] M. Cobb, N. Deodhar, and C. Vermillion. Lab-Scale Experimental Characterization and Dynamic Scaling Assessment for Closed-Loop Crosswind Flight of Airborne Wind Energy Systems. *Journal of Dynamic Systems, Measurement, and Control*, 140, July 2018.

[30] Makani. Makani kites: Airborne wind energy. https://x.company/makani/. Accessed: August 10, 2018.

[31] Altaeros. “Altaeros Energies Achieves Breakthrough in High Altitude Wind Power”. http://www.altaeros.com/, 2020. Accessed: April 1, 2020.

[32] KiteGen. Kitegen research. http://www.kitegen.com/en/. Accessed: August 10, 2018.

[33] Ampyx Power. The energy of tomorrow. https://www.ampyxpower.com/. Accessed: August 10, 2018.

[34] KiteMill. http://www.kitemill.com/. Accessed: August 10, 2018.

[35] Enerkite. Airborne wind energy. https://www.enerkite.de/en/. Accessed: August 10, 2018.

[36] N. Deodhar, A. Bafandeh, J. Deese, B. Smith, T. Muyimbwa, C. Vermillion, and P. Tkacik. Laboratory-Scale Flight Characterization of a Multitethered Aerostat for Wind Energy Generation. *AIAA Journal*, 55(6):1823–1832, 2017.

[37] P. Nikpoorparizi, N. Deodhar, and C. Vermillion. Modeling, Control Design, and Combined Plant/Controller Optimization for an Energy-Harvesting Tethered Wing. *IEEE Transactions on Control Systems Technology*, 26(4):1157–1169, 2017.

[38] N. Deodhar, J. Deese, and C. Vermillion. Experimentally Infused Plant and Controller Optimization Using Iterative Design of Experiments—Theoretical Framework and Airborne Wind Energy Case Study. *Journal of Dynamic Systems, Measurement, and Control*, 140(1), August 2017.

[39] K.B. Ariyur and M. Krstic. *Real-time Optimization by Extremum-seeking Control*. John Wiley and Sons, Hoboken, NJ, 2003.

[40] C.E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[41] C. Vermillion, B. Glass, and B. Szalai. Development and Full-Scale Experimental Validation of a Rapid Prototyping Environment for Plant and Control Design of Airborne Wind Energy Systems. *Proceedings of ASME Dynamic Systems and Control Conference*, 2014.

[42] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons Inc, 8 edition, 2012.

[43] W.J. Welch. Computer-aided Design of Experiments for Response Estimation. *Technometrics*, 26(3):217–224, 1984.

[44] F. Pukelsheim. *Optimal Design of Experiments*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2006.

[45] C. Williams and C.E. Rasmussen. *Gaussian Processes for Machine Learning*, volume 2. MIT press Cambridge, MA, 2006.

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

[47] P.A. Ioannou and J. Sun. *Robust Adaptive Control*, volume 1. PTR Prentice-Hall Upper Saddle River, NJ, 1996.

[48] I. Landau. *Adaptive Control: The Model Reference Approach*, volume 1. CRC Press, 1979.

[49] S. Nicosia and P. Tomei. Model Reference Adaptive Control Algorithms for Industrial Robots. *Automatica*, 20(5):635–644, 1984.

[50] L. Praly. Robust Model Reference Adaptive Controllers, Part I: Stability Analysis. In *The 23rd IEEE Conference on Decision and Control, 1984.*

[51] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic. A New Generation of Adaptive Controllers for Linear Systems. In *Proceedings of the 31st IEEE Conference on Decision and Control*, Dec 1992.

[52] M. Krstic, I. Kanellakopoulos, and P. Kokotovic. Nonlinear Design of Adaptive Controllers for Linear Systems. *IEEE Transactions on Automatic Control*, 39(4):738–752, 1994.

[53] M.G. Safonov and T.C. Tsao. The Unfalsified Control Concept and Learning. In *Proceedings of the 33rd IEEE Conference on Decision and Control, 1994.*

[54] F. B. Cabral and M. G. Safonov. A Falsification Perspective on Model Reference Adaptive Control. In *Proceedings of the 35th IEEE Conference on Decision and Control, 1996.*

[55] K.B. Ariyur and M. Krstic. Analysis and Design of Multivariable Extremum Seeking. *Proceedings of American Control Conference*, 2002. Anchorage, AK.

[56] R. Jin, W. Chen, and A. Sudjianto. On Sequential Sampling for Global Metamodeling in Engineering Design. In *Proceedings of International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2002. Montreal, Quebec, Canada.

[57] D. Jones, M. Schonlau, and W. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13:455–492, 1998.

[58] S. Haykin. *Kalman Filtering and Neural Networks*. Wiley, New York, 2003.

[59] L. Csató and M. Opper. Sparse On-line Gaussian Processes. *Neural computation*, 14(3):641–668, 2002.

[60] M. F. Huber. Recursive Gaussian Process Regression. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013.

[61] M. F. Huber. Recursive Gaussian Process: On-line Regression and Learning. *Pattern Recognition Letters*, 45:85–91, 2014.

[62] A.E. Bryson Jr and L.J. Henrikson. Estimation using Sampled Data Containing Sequentially Correlated Noise. *Journal of Spacecraft and Rockets*, 5(6):662–665, 1968.

APPENDIX

Definition of Gaussian distribution

A random varible $\mathbf{x} \in \mathbb{R}^n$ *has a Gaussian distribution with mean* $\mathbf{m} \in \mathbb{R}^n$ *and covariance* $\mathbf{P} \in \mathbb{R}^{n\times n}$ *if its probability density has the form*

$$N(\mathbf{x}|\mathbf{m},\mathbf{P}) \quad = \quad \frac{1}{(2\pi)^{n/2}|\mathbf{P}|^{1/2}} exp(\frac{-1}{2}(\mathbf{x}-\mathbf{m})^T\mathbf{P}^{-1}(\mathbf{x}-\mathbf{m})) \tag{92}$$

where $|\mathbf{P}|$ is the determinant of the matrix $\mathbf{P}$

Definition of joint distribution of Gaussian variables

If random variables $\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{y} \in \mathbb{R}^n$ *have the Gaussian probability distributions*

$$\mathbf{x} \sim N(\mathbf{m},\mathbf{P}), \tag{93}$$

$$\mathbf{y}|\mathbf{x} \sim N(\mathbf{Hx}+\mathbf{u},\mathbf{R}), \tag{94}$$

then the joint distribution of $\mathbf{x},\mathbf{y}$ *and the marginal distribution of* $\mathbf{y}$ *are given as*

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{Hx}+\mathbf{u} \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{PH}^T \\ \mathbf{HP} & \mathbf{HPH}^T+\mathbf{R} \end{pmatrix} \right), \tag{95}$$

$$y \sim N(\mathbf{Hx}+\mathbf{u},\mathbf{HPH}^T+\mathbf{R}). \tag{96}$$

Conditional distribution of Gaussian variables

If the random variables $\mathbf{x}$ *and* $\mathbf{y}$ *have the joint Gaussian probability distribution*

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B} \end{pmatrix} \right), \tag{97}$$

then the marginal and conditional distributions of **x** *and* **y** *are given as follows*

$$\mathbf{x} \sim N(\mathbf{a}, \mathbf{A}), \tag{98}$$

$$\mathbf{y} \sim N(\mathbf{b}, \mathbf{B}), \tag{99}$$

$$\mathbf{x}|\mathbf{y} \sim N(\mathbf{a} + \mathbf{C}^T\mathbf{B}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}^T\mathbf{B}^{-1}\mathbf{C}), \tag{100}$$

$$\mathbf{y}|\mathbf{x} \sim N(\mathbf{b} + \mathbf{C}^T\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^T\mathbf{A}^{-1}\mathbf{C}), \tag{101}$$