Iterative Learning-Based Path Optimization for Repetitive Path Planning, with Application to 3D Crosswind Flight of Airborne Wind Energy Systems

Mitchell Cobb[1], Kira Barton[2], Hosam Fathy[3], and Chris Vermillion[4]

***Abstract*—This paper presents an iterative learning approach for optimizing course geometry in repetitive path following applications. In particular, we focus on airborne wind energy (AWE) systems. Our proposed algorithm consists of two key features: First, a recursive least squares fit is used to construct an estimate of the behavior of the performance index. Second, an iteration-to-iteration path adaptation law is used to adjust the path shape in the direction of optimal performance. We propose two candidate update laws, both of which parallel the mathematical structure of common iterative learning control (ILC) update laws but replace the tracking-dependent terms with terms based on the performance index. We apply our formulation to the iterative crosswind path optimization of an AWE system, where the goal is to maximize the average power output over a figure-8 path. Using a physics based AWE system model, we demonstrate that the proposed adaptation strategy successfully achieves convergence to near-optimal figure-8 paths for a variety of initial conditions under both constant and real wind profiles.**

I. INTRODUCTION

Numerous engineering applications require the generation and repeated traversal of a path, where the choice of path can have a dramatic impact on performance. These include the control of redundant robotic manipulator paths for energy usage (see [1],[2]), optimization of bipedal walking gaits (see [3], [4]), trajectory optimization of automated racing vehicles (see [5], [6]), and flight path optimization for airborne wind energy (AWE) systems (see [7], [8]). In many applications of path optimization, the path is optimized *offline*. When the path is repeated, however, there exists an opportunity to *adapt* this path from one iteration to the next. Leveraging previous iterations' results to determine the best course of action at the current iteration can be especially useful in the presence of model uncertainties and slowly varying environmental disturbances.

This research was supported by National Science Foundation award number 1538369, titled "Collaborative Research: Self-Adjusting Periodic Optimal Control with Application to Energy-Harvesting Flight."

[1]Mitchell Cobb is a PhD candidate at North Carolina State University mcobb@ncsu.edu.

[2]Kira Barton is a Professor in the Department of Mechanical Engineering at the University of Michigan, Ann Arbor, MI 48109, USA bartonkl@umich.edu.

[3]Hosam Fathy is a Professor in the Department of Mechanical and Nuclear Engineering at the Pennsylvania State University, State College, PA 16802, USA hkf2@engr.psu.edu.

[4]Chris Vermillion is an Associate Professor in the Department of Mechanical and Aerospace Engineering at North Carolina State University, Raleigh, NC 27695, USA cvermil@ncsu.edu. He is also a technical advisor and equity stakeholder for Altaeros Energies, Inc.

Airborne wind energy (AWE) systems represent one technology for which dynamic models are far from perfect and iteration-to-iteration adaptation can be invaluable. AWE systems represent an alternative to traditional wind turbines, where the tower is replaced by tethers and a lifting body (either a rigid wing, kite, or aerostat) that either houses onboard turbine(s) or enables ground-based energy generation through cyclic spooling motion (spooling out under high tensions and in under low tensions). Fig. 1 shows a variety of AWE system designs that have been developed and deployed over the past decade. In addition to accessing altitudes far in excess of conventional wind turbines, AWE systems can execute repeated *crosswind* motions that increase the wind speed presented to the turbine (see [9]). Given the limitations of existing dynamic modeling tools, [10], [11], and [12] have investigated on-line adaptation of some attributes of the crosswind flight path. However, [10], [11] seek to optimize the flight path *location* whereas this work seeks to optimize the flight path *shape*. Additionally, [12] performs a model-based, offline optimization, whereas the work presented herein performs online optimization based on a relatively simple response surface (metamodel).

While the proposed path adaptation structure of this work follows an iterative learning control (ILC) update structure, there are several important differences. In contrast to ILC applications that focus exclusively on tracking (see [15] for an overview), the present work focuses on maximizing a performance index. Even when compared to more recent point-to-point ILC approaches (see [16]-[17]), which only penalize tracking at prescribed *waypoints* and therefore allow for the maximization (or minimization) of performance objectives *between* waypoints, the present work differs in two significant ways:

1) While the aforementioned references focus on performance between prescribed waypoints, the present work focuses on applying iterative learning to the parameters of the path itself.
2) In the present work, the path is specified entirely in space, not in time, thereby allowing the total iteration time to vary from one iteration to the next.

Motivated by the AWE application, this paper presents an iterative learning framework for adapting the parameters that define a repetitive path, with the objective of maximizing a performance index. The framework is validated through simulation on an AWE system model, where the performance index

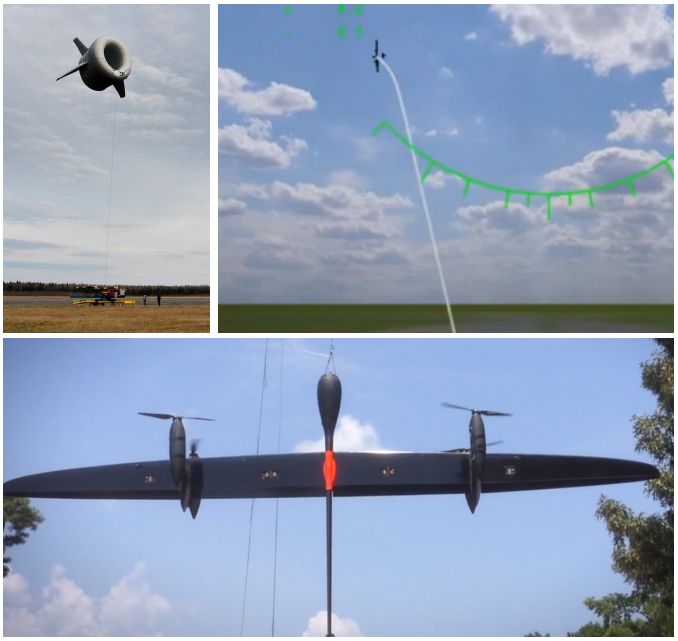

Fig. 1: Two examples of AWE system designs: Image credit Altaeros, Inc. [13] (top left), and WindLift, Inc. [14] (top right, bottom). Obtained with permission.

is driven by iteration-averaged power output. The proposed adaptation algorithm consists of two steps:

1) An *estimation* component that uses a recursive least squares (RLS) estimator to update an estimated model of the performance index vs. path parameters;
2) A *path parameter update law* whose structure parallels that of an iterative learning control (ILC) update. The update law also contains a *persistent excitation* component that ensures adequate exploration of the space of available path geometries.

Given that this work uses iterative learning to maximize an economic index (average power generation), one might draw parallels to the learning model predictive control (LMPC) of [5] and [6] attempt to optimize performance over a given "track". Although LMPC seeks to minimize an economic objective (time) in a repeated task (automated racing), it does not treat the optimization of the path itself. This makes sense in the autonomous racing example of [5] and [6], where the track is fixed. However, in applications such as the AWE system, there exists considerable opportunity to enhance performance by iteratively adapting the path itself.

In order to validate our approach on the AWE application, we consider a simplified physics-based model of an AWE system wherein the system is constrained to move on the surface of a sphere. The translational *kinematics* of this model are extremely similar to the unicycle model of [18], but the *dynamics* differ significantly. Both the translational and rotational accelerations are driven by aerodynamic forces, rather than modeled as static functions of the position and orientation, as in [18]. The model in this work consists of a fuselage, whose orientation is controlled by a rudder and dictates the direction of motion through a nonholonomic constraint, as well as a large main wing which is used to generate translational motion. These model simplifications enable us to work with a relatively simple lower level path following controller while preserving the fundamentals of the path *optimization* problem, which is the main focus of this work.

To apply our proposed path adaptation framework to the AWE application, we parameterize a figure-8 crosswind path in terms of basis parameters, which describe the width and height of the figure-8 path on the surface of the sphere. It is these basis parameters, which are directly related to the path geometry, that are adapted at each iteration.

In summary, the present paper makes several contributions relative to the existing literature:

- We describe an iterative learning framework for adapting the parameters that define a repetitive path, with the goal of maximizing an *economic* performance index. This differs sharply from traditional ILC work that focuses on reducing tracking error for a predefined reference trajectory. The focus on maximizing a performance index rather than merely tracking a reference requires additional components in the control formulation, beyond the ILC update law itself, as described in Section III-A.
- We present a physics-based AWE system model that accurately captures translational dynamics of the airborne system while imposing a unicycle constraint that simplifies the lower-level path following problem.
- We demonstrate the efficacy of the proposed path adaptation algorithm on the medium-fidelity model under both a constant and realistic wind profile based on real wind data.

This work represents an extension the authors' previous conference publication, [19]. However, it is differentiated from that work by several significant changes and additions, specifically:

- The dynamic model has been enhanced to reflect the 3D dynamics of the system. It is also provided in full detail.
- The lower-level controller now utilizes a pure-pursuit, path-following strategy that is capable of tracking a larger set of course geometries under variable wind conditions.
- We show that, under ideal conditions, the iterative update is guaranteed to converge.
- We validate the optimization strategy in a realistic wind profile, based on NREL data.

This paper is organized as follows: Section II describes the overall problem formulation, including a detailed description of AWE system model. Section III describes the general path geometry adaptation algorithm, initial convergence analysis, and application to the AWE system. This section describes both the upper-level path adaptation algorithm and lower-level path following algorithm for the AWE application. Section IV presents simulation results for the AWE application, demonstrating convergence to near-optimal figure-8 paths for a variety of initial conditions under both a constant wind profile and a realistic wind profile, based on data obtained from the National Renewable Energy Lab (NREL).

II. PROBLEM FORMULATION AND AWE SYSTEM MODEL

A. Generic Problem

Consider a mobile dynamic system composed of a plant and a lower-level closed loop path following controller, which are described, respectively, by:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), d_e(t)), \quad (1)$$

$$\mathbf{u}(t) = g(\mathbf{x}(t), \vec{p}(s)\,). \quad (2)$$

Here, $\mathbf{x} \in \mathbb{R}^N$ represents the system state, $\mathbf{u} \in \mathbb{R}^p$ represents the control input vector, $\vec{p}(s) \in \mathbb{R}^m$, represents the path to be followed in m spatial dimensions, and $d_e(t)$ represents a measurable external environmental signal. In this work, bold face font indicates a generic vector while the arrow specifically indicates a spatial vector in $\mathbb{R}^3$. The path is specified *only* in terms of the variable $s \in \mathbb{R}$, defined on some range $s \in [s_i, s_f]$, termed the "path variable". In this work we address sequences of closed paths, intended for continuous or discontinuous operation, where $\vec{p}(s_i) = \vec{p}(s_f)$.

We seek to find a path, $\vec{p}^*(s)$, that maximizes a performance index, $J(\vec{p}(s), \mathbf{x}(0), d_e(t))$, subject to the constraints that the system follows the prescribed path. In doing so, we assume that the lower-level controller has been fully defined, as in (2). This problem is stated mathematically in (3) - (4), where (3) prescribes a generic performance index and (4) requires the system track the prescribed path, $\vec{p}$:

$$\underset{\vec{p}(s)}{\text{maximize}} \quad J\big(\vec{p}(s), \mathbf{x}(0), d_e(t)\big) = \int_{T_i}^{T_f} l(\mathbf{x}(t), \vec{p}(s), d_e(t))dt \quad (3)$$

$$\text{subject to :} \quad \text{Equations} \quad (1) \quad \text{and} \quad (2),$$

$$\min_s \big\{ d(\vec{r}(t), \vec{p}(s)) \big\} \leq d_t \quad \forall\, t. \quad (4)$$

Here, $\mathbf{x}(0)$ is the initial state of the system, $d(\vec{r}(t), \vec{p})$ is the distance between the position, $\vec{r}(t) \subset \mathbf{x}$, and all points on the path $\vec{p}(s) \subset \mathbb{R}^3$, d_t is a distance tolerance for path tracking, and T_i and T_f are the times when the system begins and finishes the course, or path. In order to obtain a closed-form, iteration-to-iteration path adaptation law, it will be useful for us to approximate the hard path tracking constraint of (4) with a path tracking penalty, resulting in the following modified optimization problem:

$$\underset{\vec{p}(s)}{\text{maximize}}\ J\big(\vec{p}(s), \mathbf{x}(0), d_e(t)\big) = \quad (5)$$

$$\int_{T_i}^{T_f} l(\mathbf{x}(t), \vec{p}(s), d_e(t))dt - \frac{\kappa}{T_f} \int_0^{T_f} F(\vec{r}(t), \vec{p}(t), d_e(t))dt,$$

$$\text{subject to :} \quad \text{Equations} \quad (1) \quad \text{and} \quad (2), \quad (6)$$

where κ is a scalar weight and $F(\vec{r}(t), \vec{p}(t), d_e(t))$ is a non-negative, scalar-valued function that captures the difficulty of tracking the prescribed path. It is important to note that for the present application, where the external disturbance, $d_e(t)$, is wind, this external disturbance not only impacts lower-level path following ability (which demands a carefully designed lower-level controller that is robust to variations in wind) but also impacts energy generation (which demands a carefully designed controller for adapting the crosswind path parameters). This impact on energy generation is precisely the reason for the inclusion of d_e in the objective function. Note that the idealized, theoretical results presented later in section III-B assume that both the initial condition, $\mathbf{x}(0)$, and the environmental conditions, $d_e(t)$, are constant from iteration to iteration. However, we include a discussion of how critical design parameters can be modified to ensure robustness in the presence of iteration-varying initial conditions and environmental conditions.

B. Overview of the AWE Application

In this work, we will consider an AWE system comprised of rigid airfoils with on-board rotors as depicted in Fig. (2b). Flying the system in repetitive crosswind motions has been shown to dramatically enhance the wind speed presented to the on-board turbines (deemed the *apparent wind speed*), thereby resulting in substantially increased power production over stationary operation (see [9]). Specifically, the apparent wind velocity is given by the vector difference equation

$$\vec{v}_{app}(t) = \vec{v}_w(t) - \vec{v}(t), \quad (7)$$

where $\vec{v}_w$ is the velocity of the wind and $\vec{v}$ is the velocity of the lifting body. If the apparent wind speed is less than the turbines' rated wind speed, then the mean power output over one iteration (figure-8 "lap"), P, can be approximated as:

$$P = \frac{k}{T_f} \int_{T_i}^{T_f} |\vec{v}_{app}(t)|^3 dt, \quad (8)$$

where k is a lumped power coefficient.

C. 3D "Unifoil" Model

Successful crosswind flight for AWE systems hinges upon three requirements, namely:

1) Generating an efficient round-trip, or closed, crosswind path;
2) Controlling the lifting body's *attitude* (orientation) so that it maximizes the apparent wind speed;
3) Simultaneously controlling the lifting body's position and attitude so that it tracks this path.

In order to simplify the lower-level tracking control problem (related to requirements 2 and 3), while preserving fundamental attributes of the upper-level crosswind path optimization problem (which relates directly to requirement 1), we propose a model that we term the "unifoil" model. This model fully characterizes the longitudinal dynamics of the airborne system, which are critical to power production, while imposing a kinematic constraint (a "unicycle" constraint) whereby the AWE system moves in the direction that it is pointed. This allows us to solve the path following problem relatively easily, which in turn allows us to focus on validating the use of an iterative learning process to optimize that path.

This model is based on the design of a rigid wing system with on-board turbines, similar to the system shown in the bottom of Fig. (1) and shown schematically in Fig. (2b). The AWE system is characterized by five elements:

1) A single large wing;
2) A horizontal stabilizer, whose lift and drag properties are lumped with the main wing;
3) A vertical stabilizer (with rudder), positioned several meters behind and perpendicular to the wing, and;
4) An aerodynamically negligible fuselage that rigidly connects the wing and the rudder.
5) Counter-rotating on-board turbines.

The lower level controller actuates the main wing to generate translational motion from the aerodynamic forces on the airfoil. It actuates the rudder to generate a turning or yawing moment. The system's kinematics are simplified through the following constraints, which are reflective of a system with (i) stiff tethers, (ii) limited sideslip, and (iii) a high lift/weight ratio:

1) The distance from the origin is fixed, or equivalently, the system is constrained to lie on a sphere centered at the origin. This is also equivalent to assuming that the tether(s) are always perfectly taut.
2) The system only translates on the sphere in the direction that the fuselage is pointing, which is taken to be the direction of the body-fixed x-axis. This is the familiar no-slip kinematic constraint.
3) The body-fixed x-z plane (i.e., the longitudinal plane) is constrained to be perpendicular to the sphere. This assumption approximately holds when lift forces significantly exceed gravitational forces. In these cases, the airborne system requires minimal roll, relative to the sphere, in order to stay aloft.

It should be emphasized that this model does account for a non-zero *aerodynamic* sideslip angle (i.e., the angle between the apparent wind vector and the longitudinal plane); however, it does not allow for *geometric* slip (i.e., the airborne system only moves in the direction that it is pointing). Under these constraints, the system can be completely described by three variables, along with their associated derivatives. These are the azimuth angle, Θ, zenith angle, Φ, and the heading on the sphere, Γ, which is the complement of the velocity angle used in [20]. A schematic depiction of the system is given in Fig. (2). The first step in determining the aerodynamic forces on the wing and stabilizers is to describe the apparent wind vector from the ground-fixed coordinate frame, $\vec{v}_{app}^{\,G}(t)$, in the body-fixed coordinate frame.

1) Apparent Wind: The orientation of the model is described by the set of three Euler angles: roll, $\phi(t)$, pitch, $\theta(t)$, and yaw, $\psi(t)$. The rotation matrix, which relates a vector in the ground-fixed coordinate system to a vector in the body-fixed coordinate system, $\mathbf{R}_{G\to B}(\phi(t),\theta(t),\psi(t)) \in \mathbb{R}^{3\times 3}$, has the common form available in [21]. Therefore the apparent wind vector in the body-fixed coordinate frame, $\vec{v}_{app}^{\,B}(t)$ is given by $\vec{v}_{app}^{\,B}(t) = \mathbf{R}_{G\to B}(t)\vec{v}_{app}^{\,G}(t)$.

Turning is accomplished though lateral force on the vertical stabilizer, induced by deflection of the rudder, through a control input u_r. The lateral force imparted on the vertical stabilizer is a function of the effective angle of attack of the stabilizer, which is a function of the angle between the apparent wind and stabilizer, along with the rudder deflection:

$$\alpha_r(t) = \gamma_r(t) + k_r u_r(t). \tag{9}$$

The angle of attack of the main wing is governed by the angle of the apparent wind relative to the tangent plane to the sphere on which the AWE system is flying, γ_w, and the pitch angle of the AWE system relative to the sphere, u_w. The

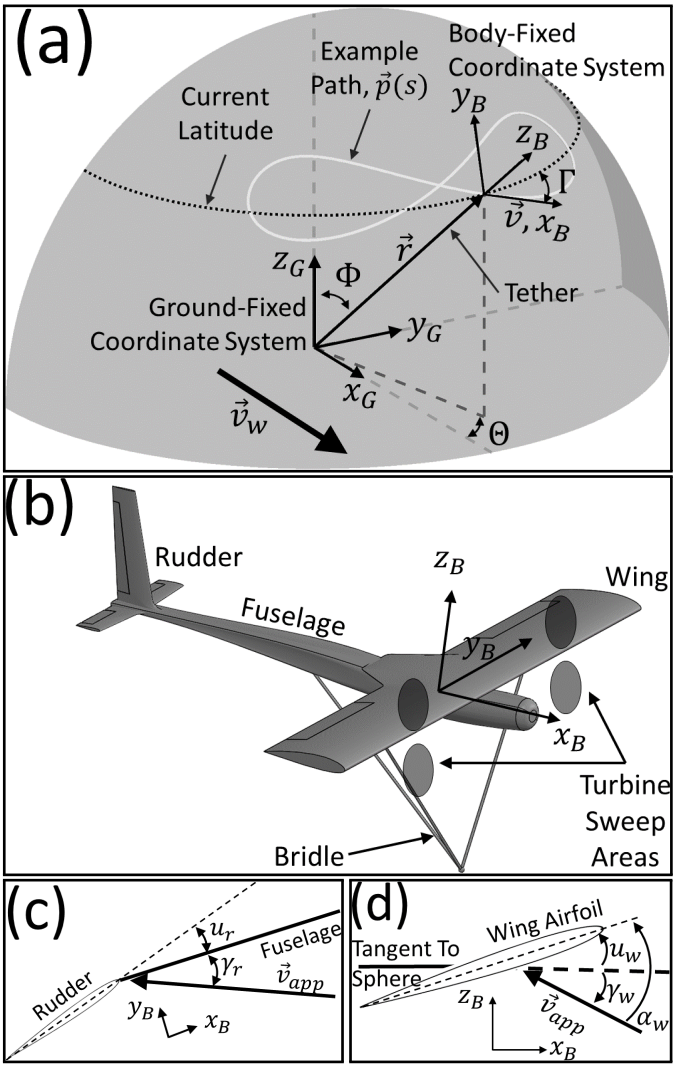

Fig. 2: Schematic of numerical AWE model. Subfigure (a) shows a wide-angle view of the system. It includes the polar ground-fixed coordinate system, (r,Θ,Φ), the Cartesian ground fixed coordinate system, (x_G, y_G, z_G), an example system path, $\vec{p}(s)$, the tether, the sphere on which the system operates, and the heading angle, Γ. Subfigure (b) shows a close up of the system including the wing, fuselage, horizontal stabilizers, and vertical stabilizer/rudder. body-fixed coordinate system, (x_B, y_B, z_B), on-board turbines, and the control inputs, u_r and u_w. Subfigure (c) shows the relationship between u_r, α_r, and γ_r, and subfigure (d) shows the relationship between u_w, α_w, and γ_w.

latter can ultimately be controlled via elevators. Ultimately, the wing's angle of attack is given by:

$$\alpha_w(t) = \gamma_w(t) + u_w(t). \tag{10}$$

The aforementioned angles are depicted relative to the body fixed reference frame at the bottom of Fig. (2).

The wing (which is rigidly attached to the fuselage and stabilizers) is allowed to rotate about its span, which is aligned with the body fixed y-axis. Therefore the angular deflection from the wing control signal, $u_w(t)$ occurs in the x-z body-fixed plane as depicted in Fig. (2b). Likewise, the rudder is allowed to rotate about its span, which is aligned with the body fixed z-axis. Therefore the angular deflection from the rudder control signal, $u_r(t)$ occurs in the x-y body-fixed plane as depicted in Fig. (2b). The additive terms, $\gamma_{w,r}(t)$, are the two angles of incidence between the apparent wind and the fuselage. They are calculated from the body-fixed components of the apparent wind vector as

$$\gamma_w(t) = \tan^{-1}\left(\frac{v^B_{app,z}(t)}{v^B_{app,x}(t)}\right) \tag{11}$$

$$\gamma_r(t) = \tan^{-1}\left(\frac{v^B_{app,y}(t)}{v^B_{app,x}(t)}\right). \tag{12}$$

2) Translational Forces: In light of the no-slip constraint, the total accelerations in the body-fixed z and y directions are equal to zero, and the total acceleration in the body-fixed x direction, a_B, is expressed as:

$$\begin{aligned} a_B =& g^B_x + \frac{\rho}{2m}||\vec{v}^B_{app}||^2 \\ &\Big(A^w_{ref}C^w_L(\alpha_w)\sin(\gamma_w) - A^w_{ref}C^w_D(\alpha_w)\cos(\gamma_w) \\ &+ A^r_{ref}k^r_{L1}\alpha_r\sin(\gamma_r) - A^r_{ref}\left(k^r_{D0} + k^r_{D2}\alpha_r^2\right)\cos(\gamma_r) \\ &- n_{turb}A^t_{ref}C^t_D\cos(\gamma_w)\Big) \end{aligned} \tag{13}$$

where the time dependence has been suppressed for notational simplicity. In this expression, g^B_x is the projection of the gravitational vector onto the body fixed x-axis, m is the total mass of the system, n_{turb} is the number of turbines, ρ is the density of air, $A^{w,r,t}_{ref}$ represents the reference area of the wing/horizontal stabilizer, rudder/vertical stabilizer, and turbine, respectively, $C^{w,t}_{D,L}$ represents the coefficients of lift and drag of the wing or turbine, respectively and $k^r_{L,D}$ are the coefficients to the linear and quadratic fits of the vertical stabilizer/rudder lift coefficient and drag coefficient polars. The coefficients of lift are a function of the angle of attack and are obtained from a lookup table generated in XFLR5 [22]. The specifications for the main wing and the stabilizer/rudder used in this work are available on [23] as airfoils NACA 2412 and HT05 respectively. Both were chosen for their extremely high lift-to-drag characteristics which are necessary to achieve significantly increased power production as well as nimble handling without full control of the roll, which is restricted by the third constraint.

To account for induced drag due to wingtip vorticies (which is unaccounted for in 2D XFLR5 predictions), the drag coefficients used in simulation have been corrected according to:

$$C^{w,r}_D(\alpha_{w,r}) = \min\{\widetilde{C}^{w,r}_D(\alpha_{w,r})\} + \frac{\left(C^{w,r}_L(\alpha_{w,r}) - C_{L0}\right)^2}{\pi\, e\, AR^{w,r}} \tag{14}$$

where $\widetilde{C}^{w,r}_D$ is the set of estimated drag coefficients produced by XFLR5's 2D analysis, C_{L0} is the coefficient of lift at the angle of attack where the estimated coefficient of drag is minimized, e is the Oswald efficiency, and $AR^{w,r}$ is the aspect ratio of the airfoil in question.

3) Rotational Moments: For the purposes of this work, the rudder couples with the nonholonomic no-slip constraint to produce rotational motion. The total rotational moment about the body fixed z-axis is calculated as:

$$M_z = \frac{1}{2}\rho||v^B_{app}||^2 l_f A^r_{ref}\left(k^r_{L0} + k^r_{L1}\alpha\right)\cos(\gamma_r) \tag{15}$$

where l_f, the length of the fuselage, determines the moment arm. The quantities $k^r_{L,0}$ and $k^r_{L,1}$ are coefficients of the linear fit of the lift coefficients $C^r_L(\alpha_r)$ obtained from XFLR. Note that the moment induced by aerodynamic drag on the vertical is assumed to be negligible. Additionally, because the vertical stabilizer and rudder are symmetric, k_{L0} is zero.

4) Position and Orientation in the Ground-Fixed Coordinate System: In order to translate these body-fixed forces and moments into a set of ground-fixed coordinates and orientations, we employ the same mathematical structure as [18], specifically:

$$\dot{\Theta}(t) = \frac{v^B_x(t)}{r\cos\left(\Phi(t)\right)}\sin\left(\frac{\pi}{2} - \Gamma(t)\right) \tag{16}$$

$$\dot{\Phi}(t) = -\frac{v^B_x(t)}{r}\cos\left(\frac{\pi}{2} - \Gamma(t)\right) \tag{17}$$

$$\ddot{\Gamma}(t) = -\frac{M_z}{J} \tag{18}$$

where the velocity in the body-fixed x-direction, $v^B_x(t)$, is given by integrating equation (13). The Euler angles are immediately derived from the Cartesian representation of the body-fixed unit vectors as detailed in [24].

5) Plant Parameter Values Used In Simulation: Table I provides the values of the plant parameters used in this work.

III. CONTROL DESIGN FRAMEWORK

In this section, we propose two iterative path adaptation laws for repetitive path following, then proceed to show how these adaptation laws can be tailored to the AWE application. Because the path adaptation laws focus on *optimizing* the path, not *tracking* the path, either one must be accompanied by a lower-level path tracking controller. This section also provides details of how this lower-level tracking controller has been designed for the AWE application.

Var.	Description	Value	Units
ρ	Air density	1.225	kg·m^{-3}
m	Total system mass	100	kg
J	Rotational inertia about z_B	1062	kg· m^2
r_0	Tether length	100	m
L_w	Chord length of main wing	1	m
L_r	Chord length of vert. stabilizer + rudder	1	m
S_w	Span of main wing	7	m
S_r	Span of vertical stabilizer + rudder	1.5	m
A^w_{ref}	Ref. area of main wing	7	m^2
A^r_{ref}	Ref. area of vert. stabilizer + rudder	1.5	m^2
-	Diameter of single turbine	0.25	m
A^t_{ref}	Ref. area of single turbine	0.049	m^2
n_{turb}	Number of turbines	4	-
AR^w	Aspect ratio of main wing	5	-
AR^r	Aspect ratio of vert. stabilizer + rudder	1.5	-
l_f	Length of fuselage	7	m
v_w	Constant wind speed	7	m·s^{-1}

TABLE I: Parameter values used in simulation results.

A. General Iterative Path Adaptation Structure

The purpose of the path adaptation algorithm is to *use previous iterations' paths and corresponding performances to adjust future paths to achieve convergence to an optimal path*. In applications where the next iteration starts at the exact time that the current iteration ends, the optimization and path adaptation must take place *very* quickly, so that the new path geometry is available at the beginning of each iteration. In order to accomplish this, we parameterize our path in terms of a small set of *basis parameters*, $\mathbf{b} \in \mathbb{R}^n$. That is, we replace $\vec{p}(s)$ in equations 1, 4, and 5, with $\vec{p}(s, \mathbf{b})$. We then apply one of the following iterative adaptation laws:

Option 1: Error-based adaptation

$$\mathbf{b}_{j+1} = K_b \mathbf{b}_j + K_e\big(\hat{\mathbf{b}}^*_j - \mathbf{b}_j\big) + \mathbf{p}_j, \tag{19}$$

Option 2: Gradient-based adaptation

$$\mathbf{b}_{j+1} = K_b \mathbf{b}_j + K_e \nabla \hat{J}(\mathbf{b}_j) + \mathbf{p}_j. \tag{20}$$

In both update laws, $\mathbf{b}_j \in \mathbb{R}^n$ represents the basis parameters at iteration j, and matrices $K_b \in \mathbb{R}^{n\times n}$ and $K_e \in \mathbb{R}^{n\times n}$ represent learning gains.

In (19), referred to as the error-based update law, $\hat{\mathbf{b}}^*_j$ represents the *estimated* optimal basis parameters at the current iteration. In (20), referred to as the gradient-based update law, $\nabla \hat{J}(\mathbf{b}_j)$ represents the gradient of the *estimated* response surface, $\hat{J}$, at the current basis parameters, $\mathbf{b}_j$, of iteration j.

With the exception of the term $\mathbf{p}_j$, the structure of the error-based update law, (19), parallels the structure of traditional ILC, with $\mathbf{b}_j$ taking the place of the control input sequence and $\hat{\mathbf{b}}^*_j - \mathbf{b}_j$ taking the place of the error term. When taking $K_b = I$, the first two terms of the adaptation law reflect movement in the design space, towards the perceived optimizer, $\hat{\mathbf{b}}^*_j$.

Likewise, the gradient-based update law of (20), parallels the structure of gradient-based ILC [25], with $\mathbf{b}_j$ taking the place of the control input sequence and $\nabla \hat{J}(\mathbf{b}_j)$ taking the place of the gradient of a (squared) tracking error cost function with respect to the control input sequence. When taking $K_b = I$, the first two terms of the adaptation law reflect movement in the direction of perceived increasing performance index.

The term $\mathbf{p}_j \in \mathbb{R}^n$ in both update laws is a *persistent excitation term* that is designed to ensure adequate exploration of the domain of available basis parameters. This term ultimately ensures convergence of $\mathbf{b}_j$ to a region around the true optimal basis parameters, $\mathbf{b}^*$.

By restricting the set of possible path geometries through this parameterization of the generic path, $\vec{p}(s)$, we may sacrifice the global optimality of the final result. However, with a proper choice of parameterization, this loss of optimality may be negligible. Furthermore, by reducing the design space through parameterization, the problem gains some structure, and the optimization becomes feasible. In the case of the parameterization chosen in this work (detailed later), it also becomes computationally lightweight.

To implement the course geometry adaptation strategy, three mathematical operations must take place at each iteration:

1) The estimated response surface, $\hat{J}(\mathbf{b}_j)$, must be updated based on performance at iteration j.
2) The estimated optimal basis parameters, $\hat{\mathbf{b}}^*_j$, or the estimated gradient of the response surface, $\nabla \hat{J}(\mathbf{b}_j)$, must be calculated from the estimated response surface.
3) The path geometry for the next iteration, $\vec{p}_{j+1}(s)$, must be updated according to the expression for $\vec{p}(s)$ and the appropriate choice of (19) or (20).

We now consider how to quickly and efficiently perform the first two operations.

1) Estimating the Response Surface: To estimate $\hat{J}(\mathbf{b}_j)$, we model the performance index as the inner product of a regressor vector, $\mathbf{h}(\mathbf{b}) \in \mathbb{R}^q$, and a coefficient vector, $\boldsymbol{\beta} \in \mathbb{R}^q$, as follows:

$$\hat{J}(\mathbf{b}_j) = \mathbf{h}(\mathbf{b_j})^T \boldsymbol{\beta}_j. \tag{21}$$

Here, $\hat{J}(\mathbf{b}_j)$ represents an approximation of the performance index, and the regressor vector structure, $\mathbf{h}(\mathbf{b}_j)$, is selected to encode the anticipated dependency of J on the basis parameters (e.g., if we expect that J is quadratic with respect to the basis parameters, then terms $\mathbf{h}(\mathbf{b}_j)$ should include the squares of basis parameters). The coefficients to the estimated response surface, $\boldsymbol{\beta}$ are then identified at each iteration, j, using recursive least squares (RLS), with an exponential forgetting factor, λ, as follows:

$$\begin{aligned} V_j &= \frac{1}{\lambda}\left(V_{j-1} - \frac{V_{j-1}\mathbf{h}(\mathbf{b}_j)\mathbf{h}(\mathbf{b}_j)^T V_{j-1}}{1 + \mathbf{h}(\mathbf{b}_j)^T V_{j-1}\mathbf{h}(\mathbf{b}_j)}\right), \quad \lambda \le 1 \\ \beta_j &= \beta_{j-1} + V_j \mathbf{h}(\mathbf{b}_j)\big(J(\mathbf{b}_j) - \mathbf{h}(\mathbf{b}_j)^T \beta_{j-1}\big) \end{aligned} \tag{22}$$

where V_j is the inverse of the weighted sample covariance matrix. In this work, V_j and β_j are initialized by first testing five user-specified points in design space and then solving the least squares problem.

Given a parameter estimate, $\boldsymbol{\beta}_j$, the corresponding optimizer or gradient of $\mathbf{h}(\mathbf{b}_j)^T\boldsymbol{\beta}_j$, denoted by $\hat{\mathbf{b}}^*_j$, or $\nabla \hat{J}(\mathbf{b}_j)$, is computed either analytically or numerically.

Note that this method of parameterizing the response surface implicitly assumes that the initial condition, $\mathbf{x}(0)$, and environmental conditions, $d_e(t)$ are constant from iteration to iteration, as mentioned earlier. This realization further motivates the introduction of the forgetting factor, λ, which

heavily weights recently acquired data in the estimate, $\boldsymbol{\beta}_j$ thus ensuring that the estimated response surface is weighted towards data acquired using similar course geometries *and* under similar wind conditions. We provide guidelines for tuning λ later in the paper.

2) Design of Persistent Excitation Term: The term $\mathbf{p}_j$ in (19) and (20) is used to ensure adequate *exploration* of the design space, such that the estimated response surface, $\hat{J}$, ultimately converges to within a finite error of the true response surface. Specifically, to guarantee convergence of the RLS estimator, we must choose $\mathbf{p}_j$ such that the following *uniform persistent excitation* condition holds [26]:

Definition 1. *(Uniform persistent excitation) The signal $\hat{\mathbf{b}}$ with $\mathbf{b}_i \in \hat{\mathbf{b}}$, is uniformly persistently exciting if there exists an integer $T > 0$ such that:*

$$\sum_{i=k}^{k+T} (\mathbf{h}(\mathbf{b}_i))(\mathbf{h}(\mathbf{b}_i))^T \succ 0, \tag{23}$$

for all k.

Here, the symbol $\succ$ indicates that the left hand side is a positive definite matrix. In the update law of (19) and (20), $\mathbf{p}_j$ is the only term that can be freely specified at each iteration. Thus, $\mathbf{p}_j$ must be carefully designed to ensure that the above persistent excitation condition holds. In section III-D3 we will describe how $\mathbf{p}_j$ has been chosen for the AWE application.

B. Convergence Analysis - Ideal Case

In this subsection, we present initial convergence analyses, based on the basis parameter update laws of (19) and (20). Our analysis provides sufficient conditions under which $\mathbf{b}_j$ converges to a finite set containing $\mathbf{b}^*$. For simplicity, we consider the case where the learning gains take the form $K_b = I$, $K_e = k_e I$.

In performing our analysis, we make the following assumptions regarding the structure of $\mathbf{h}(\mathbf{b})$ and $\mathbf{p}_j$:

- *Assumption 1:* The initial condition, $\mathbf{x}(0)$, and external disturbance, $d_e(t)$, are iteration-invariant, and the only uncertainties are parametric:

$$J(\mathbf{b}, \mathbf{x}(0), d_e(t)) = J(\mathbf{b}) = \mathbf{h}(\mathbf{b})^T \boldsymbol{\beta}^* \text{ for some } \boldsymbol{\beta}^*. \tag{24}$$

- *Assumption 2:* The performance index, $J(\mathbf{b})$ is convex and differentiable everywhere, possesses a unique maximizer, and the gradient is Lipschitz continuous with constant L, that is:

$$\|\nabla J(\mathbf{b}_{j+1}) - \nabla J(\mathbf{b}_j)\| \leq L \|\mathbf{b}_{j+1} - \mathbf{b}_j\|. \tag{25}$$

- *Assumption 3:* The excitation signal at each iteration, $\mathbf{p}_j$, is chosen such that there exists an integer $T > 0$ for which:

$$\sum_{i=k}^{k+T} [\mathbf{h}(\xi + \sum_{k=0}^{i} (\boldsymbol{\Delta}_k + \mathbf{p}_{j+k}))][\mathbf{h}(\xi + \sum_{k=0}^{i} (\boldsymbol{\Delta}_k + \mathbf{p}_{j+k}))]^T \succ 0, \tag{26}$$

$\forall\, \xi \in \mathbb{R}^{n_b},\ \boldsymbol{\Delta}_k \in \mathbb{R}^{n_b} : \|\boldsymbol{\Delta}_k\| \leq \Delta_{\max}, k = 0 \ldots T.$

Assumption 1 guarantees that the regressor vector structure ($h(\mathbf{b})$) characterizes the actual performance index. Assumption 2 bounds the iteration-to-iteration variation in the performance index. Satisfaction of Assumption 3 ensures the necessary persistent excitation, which guarantees that $\boldsymbol{\beta}$ converges by the definition of persistent excitation.

Under the aforementioned assumptions, it is possible to guarantee convergence of $\hat{J}(\mathbf{b}_j)$ to $J(\mathbf{b})$ and $\mathbf{b}_j$ to a finite set containing $\mathbf{b}^*$, through the following propositions, which correspond to the error-based and gradient-based adaptation laws, respectively.

Proposition 1. *(Convergence of error-based iterative path adaptation law) Suppose that Assumptions 1, 2, and 3 are satisfied, and that $\|\mathbf{p}_j\| \leq p_{\max}, \forall j \geq 0$. Then under the basis parameter update law of (19) with $K_b = I$, $K_e = k_e I$, and $k_e < 1$, the following results hold:*

- $\lim_{j\to\infty} \|\mathbf{b}_j^* - \mathbf{b}_j\| = 0.$
- *The set $B = \{\mathbf{b} : \|\mathbf{b}^* - \mathbf{b}\| \leq \frac{p_{max}}{k_e}\}$ is attractive, (i.e., $\mathbf{b}$ converges to a ball of radius $\frac{p_{max}}{k_e}$ around $\mathbf{b}^*$).*

Proof. Defining $\delta_i \triangleq k_e(\hat{\mathbf{b}_j}^* - \mathbf{b}_j)$, $\mathbf{b}_{j+1}$ is related to $\mathbf{b}_j$ by:

$$\mathbf{b}_{j+i} = \mathbf{b}_j + \sum_{k=0}^{i} (\delta_{j+k} + \mathbf{p}_{j+k}). \tag{27}$$

Taking $\xi = \mathbf{b}_j$ and $\boldsymbol{\Delta}_k = \delta_{j+k}$ for $k = 0 \ldots i$, it follows from Assumption 3 that there exists $T > 0$ for which equation (23) holds for all j. Thus, uniform persistent excitation is achieved, which by Assumption 3 guarantees that β_j converges. This fact, in combination with Assumption 1 guarantees that:

$$\lim_{j\to\infty} \|\boldsymbol{\beta}_j - \boldsymbol{\beta}^*\| = 0. \tag{28}$$

According to Assumption 2, there exists a unique maximizer, $\mathbf{b}^*$, that can be computed from $\boldsymbol{\beta}^*$; therefore, it immediately follows that:

$$\lim_{j\to\infty} \|\mathbf{b}_j^* - \mathbf{b}\| = 0. \tag{29}$$

Next, note that the update law of (19) can be rewritten as:

$$\mathbf{b}_{j+1} = (1 - k_e)\mathbf{b}_j + k_e(\mathbf{b}^* + \epsilon_j) + \mathbf{p}_j, \tag{30}$$

where $\epsilon_j \triangleq \hat{\mathbf{b}}_j^* - \mathbf{b}^*$. This can be represented in the z domain by:

$$\mathbf{b}(z) = \frac{k_e}{z + k_e - 1}\big(\mathbf{b}^*(z) + \epsilon(z)\big) + \frac{1}{z + k_e - 1}\mathbf{p}(z). \tag{31}$$

Given that $\lim_{j\to\infty} \epsilon_j = 0$, $\mathbf{b}^*$ is a constant, and the poles in (31) are located at $z = 1 - k_e$, (which for $0 < k_e < 1$ indicates input-output stable and overdamped dynamics), the steady state-dynamics satisfy:

$$\|\mathbf{b}_{ss} - \mathbf{b}^*\| \leq \frac{p_{max}}{k_e}. \tag{32}$$

Thus, $\mathbf{b}$ converges to a set $B = \{\mathbf{b} : \|\mathbf{b}^* - \mathbf{b}\| \leq \frac{p_{max}}{k_e}\}$ at steady state. □

Proposition 2. *(Convergence of gradient-based iterative path adaptation law) Suppose that Assumptions 1, 2, and 3 are satisfied, and that $\|\mathbf{p}_j\| \leq p_{\max}, \forall j \geq 0$. Then under the basis*

parameter update law of (20) with $K_b = I$, $K_e = k_e I$, and $0 < k_e < \frac{2}{L}$, the following results hold:

- $\lim_{j\to\infty} \|\hat{J}(\mathbf{b}_j) - J(\mathbf{b})\| = 0$.
- *The set* $B = \{\mathbf{b} : \|\nabla J(\mathbf{b})\| \leq \frac{1}{L_{min}} \frac{p_{max}}{\frac{2}{L} - k_e}\}$ *is attractive.*

Proof. By the same logic as the proof of Proposition 1, if we define $\delta_i \triangleq k_e I \nabla \hat{J}(\mathbf{b}_j)$, $\mathbf{b}_{j+i}$ then it follows from Assumption 3 that uniform, persistent excitation is achieved, and (28) holds in this case as well. Furthermore, according to Assumption 1, the only uncertainties are parametric, therefore it immediately follows that:

$$\lim_{j\to\infty} \hat{J}(\mathbf{b}_j) - J(\mathbf{b}) = 0 \quad \forall\ \mathbf{b}. \tag{33}$$

Which implies from continuity that:

$$\lim_{j\to\infty} \|\nabla \hat{J}(\mathbf{b}_j) - \nabla J(\mathbf{b})\| = 0. \tag{34}$$

In fact, Assumption 1 guarantees that $\nabla \hat{J}(\mathbf{b}_j) = \nabla J(\mathbf{b})$ for some finite value of j. Assumption 2 provides a bound on the change in performance index between two successive iterations, $\Delta J \equiv J(\mathbf{b}_{j+1}) - J(\mathbf{b}_j)$:

$$\Delta J \geq \nabla J(\mathbf{b}_j)^T (\mathbf{b}_{j+1} - \mathbf{b}_j) - \frac{L}{2} \|\mathbf{b}_{j+1} - \mathbf{b}_j\|^2. \tag{35}$$

Substituting the update law of (20) for $\mathbf{b}_{j+1}$, with $K_b = I$ and $K_e = k_e I$, gives:

$$\Delta J \geq \nabla J(\mathbf{b}_j)^T (k_e \nabla \hat{J}(\mathbf{b}_j) + \mathbf{p}_j) - \frac{L}{2} \|k_e \nabla \hat{J}(\mathbf{b}_j) + \mathbf{p}_j\|^2. \tag{36}$$

Noting from before that $\nabla \hat{J}(\mathbf{b}_j) = \nabla J(\mathbf{b}_j)$ for a finite value of j, then for sufficiently large j, this can be algebraically rearranged to obtain:

$$\begin{aligned} \Delta J \geq \frac{L}{2} \Bigg(& k_e \left(\frac{2}{L} - K_e \right) \|\nabla J(\mathbf{b}_j)\|^2 \\ & + 2 \left(\frac{1}{L} - k_e \right) \mathbf{p}_j^T \nabla J(\mathbf{b}_j) - \mathbf{p}_j^T \mathbf{p}_j \Bigg). \end{aligned} \tag{37}$$

This is lower bounded once again by using the largest magnitude of the excitation signal, p_{max}:

$$\begin{aligned} \Delta J \geq \frac{L}{2} \Bigg(& k_e \left(\frac{2}{L} - k_e \right) \|\nabla J\|^2 \\ & + 2 \left(\frac{1}{L} - k_e \right) p_{max} \|\nabla J\| - p_{max}^2 \Bigg). \end{aligned} \tag{38}$$

Where the dependence of J on $\mathbf{b}$ has been suppressed for succinctness. Sufficient conditions on k_e and $\|\nabla J\|$ for the quadratic expression to be greater than or equal to zero are:

$$0 < k_e < \frac{2}{L} \quad \text{and} \tag{39}$$

$$\|\nabla J\| \geq \frac{p_{max}}{\frac{2}{L} - k_e}. \tag{40}$$

Defining $B = \{\mathbf{b} : \|\nabla J(\mathbf{b})\| \leq \frac{p_{max}}{\frac{2}{L} - k_e}\}$, J' as the maximum scalar such that $B \subset \{\mathbf{b} : J(\mathbf{b}) \geq J'\}$, and $S = \{\mathbf{b} : J(\mathbf{b}) \geq J'\}$, then S is invariant and $\Delta J > 0$ whenever $\mathbf{b} \notin S$. Thus the performance index will be non-decreasing (i.e. $\Delta J \geq 0$) whenever the true response surface has been identified, the learning gain is chosen appropriately and $\mathbf{b} \notin S$. □

C. Convergence Analysis - Discussion

As mentioned previously, the convergence analysis given above is based on a set of three assumptions. The most severe deviations from the ideal case occur because, in real-world systems, Assumption 1 will be violated to some extent.

In the application examined in this work, inconsistent initial conditions arise from a combination of two sources. The first is the optimization itself. By varying the shape of the path between iterations, we change (at least) the heading angle, Γ, at which the system finishes iteration j. This is then the heading angle at which the system starts iteration $j + 1$. Second, the external disturbance (the wind speed) is varying from iteration to iteration. Because the wind speed determines the system's speed, it influences the state of the system at the end of iteration j, which is then the initial state at iteration $j + 1$. These iteration-to-iteration variations must be kept manageable in size through careful tuning of the learning gains.

Furthermore, in this application there is no reason to expect that the true, global response surface satisfies Assumption 1. Rather, our estimated response surface should be viewed as a *local* approximation.

In order to increase the robustness of the optimization against the effects of violating these assumptions, we suggest three ways in which the optimization can be adjusted:

- First, we implement a trust region to limit the amount that the basis parameters are allowed to vary between iterations.
- Second, we reduce the learning gain, K_e.
- Third, we set the forgetting factor, λ to a value less than 1.

The first and second adjustments are helpful in mitigating the effects of both the inconsistent initial conditions and the local nature of the response surface estimation. The last adjustment is helpful in mitigating the effects of the inconsistent external disturbance. The idea here is that by heavily weighting the more recent data in the response surface estimation, we are more likely to fit our response surface to data measured under a similar external disturbance.

D. Iterative Path Adaptation Framework for the AWE Application

The overall structure of the AWE control system, including both the iteration domain path adaptation (which generates and optimizes the figure-8 path for the next iteration) and time-domain wing and rudder angle controllers (which allow the AWE system to follow the figure-8 path, while controlling the angle of attack to maximize instantaneous power output), is shown in Fig. 3.

1) Path Parameterization: In this work, we parameterize the path in terms of two basis parameters, the azimuthal sweep angle of the path, termed the width, W, and the zenith sweep

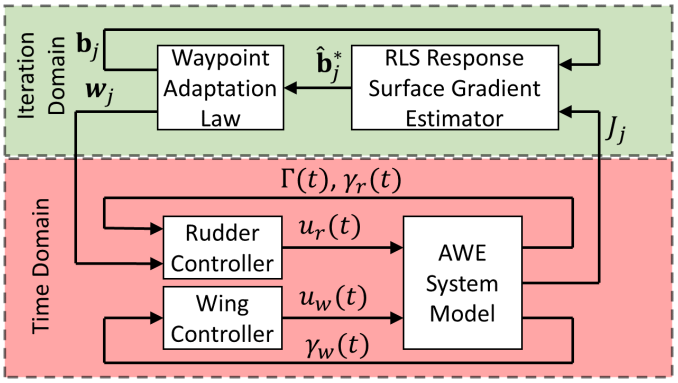

Fig. 3: Block diagram showing path adaptation performed between iterations (top) and the lower-level controllers used to follow the path and maximize instantaneous power (bottom).

angle, termed the height, H. Thus, the vector of iteration-varying basis parameters is $\mathbf{b}_j = [\ W_j \quad H_j\ ]^T$. The shape of the path, $\vec{p}_j(s, \mathbf{b}_j) = (r_j(s, \mathbf{b}_j), \Theta_j(s, \mathbf{b}_j), \Phi_j(s, \mathbf{b}_j))$, is then calculated from a parametric form of the figure-eight known as the Lemniscate of Gerono (see [27]), where:

$$r_j(s, \mathbf{b}_j) = r_0 \quad \forall j \tag{41}$$

$$\Theta_j(s, \mathbf{b}_j) = W_j \cos\left((2s + \frac{3}{2})\pi\right), \tag{42}$$

$$\Phi_j(s, \mathbf{b}_j) = -H_j \sin\left((2s + \frac{3}{2})\pi\right) \cos\left((2s + \frac{3}{2})\pi\right) + \Phi_0. \tag{43}$$

where $s \in \mathbb{S}$ is an angular path variable and the manifold of all path variables, $\mathbb{S}$, is a circle with the points $s = 0$ and $s = 1$ identified as equal. The quantity Φ_0 is the mean zenith angle of the course.

2) Performance Metric: For the AWE application, the ultimate objective is to *maximize power output in a manner that can be repeated from one iteration to the next*. This means that the performance index, J, should be chosen primarily to reflect average power *augmentation* as compared to stationary flight, but should also disallow figure-8 paths that cannot be followed adequately (which will lead to situations where the figure-8 paths cannot be repeated because they do not finish at the same initial condition that they start at). To capture these desires, the following two-term performance index is chosen and used in simulation:

$$J_j = \frac{\int_0^{T_{f,j}} v_{app,j}^3(t)dt}{\int_0^{T_{f,j}} v_{w,j}^3(t)dt} - \frac{K_t}{T_f}\int_0^{T_{f,j}} \left(\dot{\Gamma}(t)\right)^2 dt \tag{44}$$

The first term, referred to as the energy augmentation ratio (EAR), expresses the energy generation augmentation resulting from crosswind flight. The numerator is proportional to the energy produced by the system and the denominator of the first term is proportional to the energy that would have been produced by the system under stationary (non-crosswind) flight. The variable K_t is a scalar weighting parameter. The second term penalizes courses that require large turning actions, which ultimately require large aerodynamic forces to execute turns at high speeds (since centripetal acceleration is equal to the product of angular rate and velocity). Because the model used for our simulations has been intentionally simplified to facilitate path following, it is very important that we consider ease of this path following problem, and the second term in the performance index satisfies that goal.

3) Basis Parameter Adaptation: In order to implement a basis parameter update law, we first require an estimate of the response surface at each iteration, $\hat{J}(\mathbf{b}_j)$. This estimate is updated at each iteration using the RLS estimator of (22). For the purpose of constructing the RLS estimator for the AWE system, the estimated performance index, $\hat{J}$, is parameterized as $\hat{J} = \mathbf{h}(\mathbf{b}_j)^T \boldsymbol{\beta}$, where:

$$\mathbf{h}(\mathbf{b_j}) = [\ 1 \quad W_j \quad W_j^2 \quad H_j \quad H_j^2\ ]^T, \tag{45}$$

$$\boldsymbol{\beta_j} = [\ \beta_{0,j} \quad \beta_{1,j} \quad \beta_{2,j} \quad \beta_{3,j} \quad \beta_{4,j}\ ]^T. \tag{46}$$

The excitation signal is designed as a uniform random number with a user-specified range:

$$\mathbf{p}_j = [\ 2A_W(N_{R,j} - \tfrac{1}{2}) \quad 2A_H(N_{R,j} - \tfrac{1}{2})\ ]^T, \tag{47}$$

where A_W, A_H, are the user-refined ranges and $N_{R,j}$ is the number generated from a uniform random number generator which satisfies $0 \leq N_{R,j} \leq 1\ \forall\ j$. The orthogonality of the random number generation function achieves the required uniform persistent excitation.

The parameterization of (45) represents a good local but inaccurate global characterization of J, rendering the gradient-based update law of (20) a more appropriate choice in this case.

From $\boldsymbol{\beta}_j$, and $\mathbf{h}(\mathbf{b_j})$, the gradient of the estimated performance index at the current set of basis parameters, $\nabla\hat{J}_j$, can be calculated in a very straightforward fashion:

$$\nabla\hat{J}_j = [\ \beta_{1,j} + 2\beta_{2,j}W_j \quad \beta_{3,j} + 2\beta_{4,j}H_j\ ]^T \tag{48}$$

This estimated gradient is then used in a slightly modified version of the path adaptation law of (20):

$$\mathbf{b}_{j+1} = \mathbf{b}_j + K_e(\nabla\hat{J}_j, \Delta\mathbf{b}_{max})\nabla\hat{J}(\mathbf{b}_j) + \mathbf{p}_j. \tag{49}$$

Where the learning gain matrix, $K_e \in \mathbb{R}^{2\times 2}$ has now been adapted to vary from iteration to iteration in order to implement a trust region. Specifically, it now depends on estimated gradient, $\nabla\hat{J}_j$, as well as the elements of the user-defined trust region, $\Delta\mathbf{b} = [\ \Delta W_{max} \quad \Delta H_{max}\ ]^T$ according to:

$$K_e(\nabla\hat{J}_j, \Delta\mathbf{b}_{max}) = \min\left\{1, \frac{\Delta W_{max}}{|\beta_{1,j} + 2\beta_{2,j}W_j|}, \frac{\Delta H_{max}}{|\beta_{3,j} + 2\beta_{4,j}H_j|}\right\} I \tag{50}$$

E. Lower-Level Rudder (Path Following) and Wing Angle Controllers for the AWE Application

Lower-level controllers must adjust the rudder angle, u_r, in order to follow the path and must adjust the wing angle, u_w, in order to do so efficiently. The wing angle, u_w, is chosen to maximize the approximate net longitudinal acceleration given in equation (13). In order to solve for the wing angle that maximizes the longitudinal acceleration, u_w^*, it is necessary to approximate the lookup tables of equation (13) with closed form analytical functions. This is accomplished by fitting a

linear regression to the linear region of the $C_L^{w,r}$ curves and by fitting a quadratic regression to the quadratic region of the $C_D^{w,r}$ curves. Determining the range over which these lookup tables are considered linear or quadratic is left up to the user. These regressions have the form:

$$\hat{C}_L^{w,r}(\alpha_{w,r}) = k_{L1}^{w,r}\alpha_{w,r} + k_{L0}^{w,r} \tag{51}$$
$$\hat{C}_D^{w,r}(\alpha_{w,r}) = k_{D2}^{w,r}\alpha_{w,r}^2 + k_{D1}^{w,r}\alpha_{w,r} + k_{D0}^{w,r}. \tag{52}$$

We can now substitute these equations into equation (13), set the partial derivative with respect to α_w equal to zero and solve for the optimal angle of attack, $\alpha_w^*(t)$,

$$\alpha_w^*(t) = \frac{1}{2k_{D2}^w}\left(k_{L1}^w \tan\big(\gamma_w(t)\big) - k_{D1}^w\right). \tag{53}$$

Through the direct relationship between $\alpha_w(t)$ and $u_w(t)$ in equation (10), the optimal control input is then, $u_w^*(t) = \alpha_w^*(t) - \gamma_w(t)$.

Path following is accomplished by first calculating a heading setpoint, Γ_{sp}, at each instant, then using a simple feedback linearizing model reference controller to regulate the hull's orientation to that setpoint. The heading setpoint, Γ_{sp}, is determined at every instant through a pure pursuit path following approach similar to [28] that consists of three steps:

1) Compute the local projection of the current position onto the path by finding the path variable, s^*, that minimizes the great-circle distance from the current position, $\vec{r}_j(t)$, to the path, $\vec{p}_j(s, \mathbf{b}_j)$. That is, we solve the following minimization problem at every instant, t:

$$s_j^*(t) = \arg\min_{s\in\mathbb{S}^+}\{d_{Sphere}(\vec{r}_j(t), \vec{p}_j(s, \mathbf{b}_j))\}, \tag{54}$$

where d_{Sphere} denotes the great-circle, or spherical distance between the ends of the two input vectors.
Because a closed-form solution to this problem for our path geometry does not exist, we approximate the solution numerically by using the golden section algorithm [29] for our 1d minimization. In order to avoid problems originating from the self-intersecting nature of the path, this search is only performed over a small subset of possible path variables, $\mathbb{S}^+ \subset \mathbb{S}$, termed the "searched subset". The searched subset, is given by the closed subset, $[s_j^*(t-1), s_j^*(t-1) + s_s]$ where $s_j^*(t-1)$ is the path parameter found from the projection operation during the previous time step, and s_s is a user-defined parameter defining the maximum search-ahead distance. It is also worth pointing out that this formulation inherently enforces a non-decreasing constraint on the target point location, expressed by $s_c(t)$ and $\vec{r}_{c,j}(t)$.
2) Add a user-defined constant, c, to the local projection point, $s_j^*(t)$, to obtain the target point at the current instant, $s_{c,j}(t)$,

$$s_{c,j}(t) = s_j^*(t) + c. \tag{55}$$

3) Calculate the spherical heading between the current position, $\vec{r}_j(t) = (r_0, \Theta_j(t), \Phi_j(t))$, and the target, $\vec{r}_{c,j}(t) = (r_0, \Theta_{c,j}, \Phi_{c,j})$, calculated by evaluating equations (41) - (43) at $s_{c,j}(t)$

$$\Gamma_{sp} = \frac{\pi}{2} - \operatorname{atan}\left(\frac{\sin(\Theta_{c,j} - \Theta_j)}{\cos(\Phi'_{c,j})\tan(\Phi'_j) - \sin(\Phi'_{c,j})\cos(\Theta_{c,j} - \Theta_j)}\right) \tag{56}$$

where Φ' indicates the complement of a zenith angle, also known as elevation or latitude.

The rudder controller has been designed as a model reference, feedback linearizing controller. Specifically, the second-order, relative degree 2 reference model describing desirable heading angle tracking performance is given by:

$$\frac{\Gamma_m(s)}{\Gamma_{sp}(s)} = \frac{1}{(\tau s + 1)^2} \tag{57}$$

where $\Gamma_m(t)$ is the reference model output, which is designed to follow $\Gamma_{sp}(t)$ with time constants of τ. By substituting the approximated aerodynamics of equation (51) into the rotational dynamic equation (15) we can obtain an approximate model of the rotational plant dynamics that is then inverted to obtain our control law:

$$u_{r,cmd}(t) = -\frac{J}{M_z(\gamma_r)}\Big(\frac{1}{\tau^2}(\Gamma_{sp} - \Gamma) - \frac{2}{\tau}\dot{\Gamma}\Big) - \gamma_r. \tag{58}$$

Here, the term $M_z(\gamma_r)$ is given by (15). To ensure that we do not attempt unrealistic rudder deflections, we saturate the commanded rudder angle to a range of $\pm 30°$. Thus, the final command to the rudder is given as $u_r = \max\{\min\{u_{r,cmd}(t), 30°\}, -30°\}$.

F. Controller Parameter Values Used In Simulation

Values for controller parameters used in the simulation results of Section IV are provided in Table II.

Variable	Description	Value	Units
μ	Forgetting factor	0.98	-
A_W	Width excitation amplitude	0.75	deg
A_H	Height excitation amplitude	0.2	deg
ΔW_{max}	Width trust region	3	deg
ΔH_{max}	Height trust region	1	deg
τ	Reference model time constant	0.03	s
Φ_0	Mean path elevation angle	40	deg
k_e	Learning gain	5	-
K_t	Performance index tracking weight	20	-
c	Carrot lead path distance	0.02	-
s_s	Path search-ahead distance	0.01	-

TABLE II: Parameter values used in simulation results.

IV. SIMULATION RESULTS AND DISCUSSION

This control strategy was tested under two wind conditions. The first, termed the *constant* wind profile was a spatially and temporally constant wind speed. The second, termed the *variable* wind profile was a spatially uniform but temporally varying wind speed based on data obtained from the NREL National Wind Technology Center [30].

A. Constant Wind Profile

In order for this ILC-based path optimization to be successful under a constant wind profile, it should produce paths that converge to a fixed geometry. Furthermore, the performance index, J_j, should increase as the iteration number increases, as should the average power augmentation. Fig. (4) shows the basis parameters over one optimization for a variety of initial conditions. This figure demonstrates that the figure-8 path converges for a variety of initial conditions. Fig. (5) shows some examples of the initial and final three-dimensional shapes resulting from the optimization starting from initial course geometry 2.

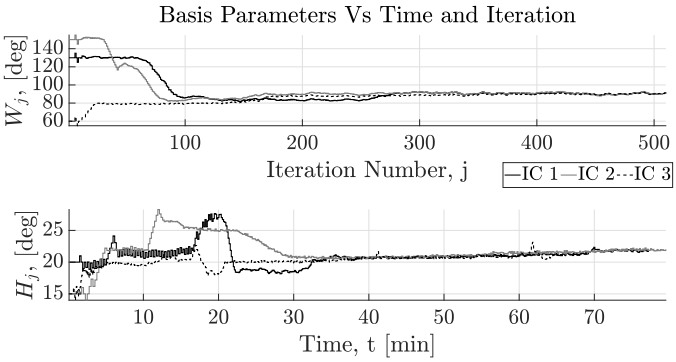

Fig. 4: Evolution of basis parameters for a variety of initial conditions under a constant wind profile.

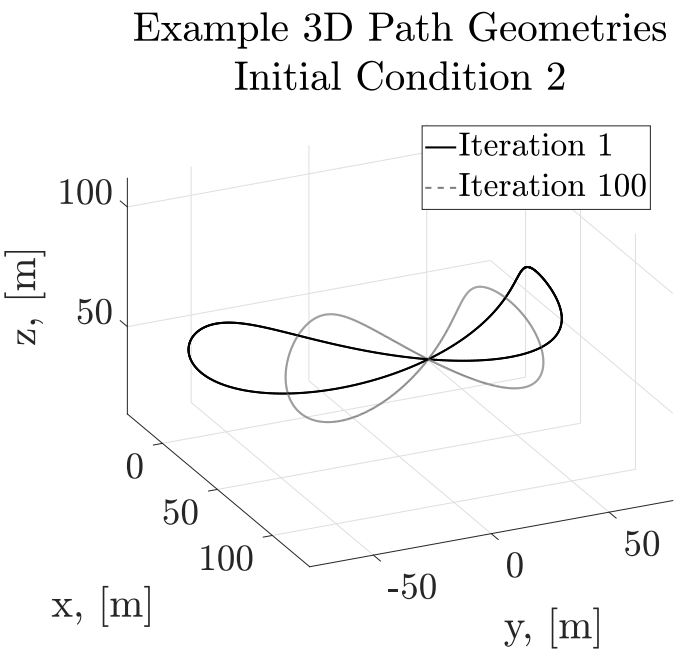

Fig. 5: Example initial and final path geometries for initial condition 2.

Additionally, the path optimization should result in paths that present the turbines with more wind from iteration to iteration. In order to examine the amount of wind presented to the turbines from iteration to iteration, Fig. 6 shows the cubed ratios of the apparent wind speed to the true wind speed.

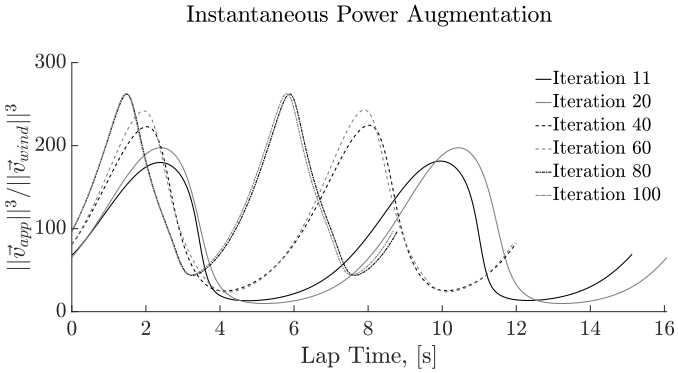

Fig. 6: Evolution of apparent wind speed (relative to the constant, true wind speed) as a function of time, for several iterations of the optimization.

By examining the performance index as a function of the iteration number, as shown in Fig. (7), we can see that the majority of the improvement in the performance index occurs in the first 50 iterations. This evidence further suggests that the response surface $J(\mathbf{b})$ is relatively flat in the vicinity of the optimum, J^*, at least for the constant wind speed chosen here.

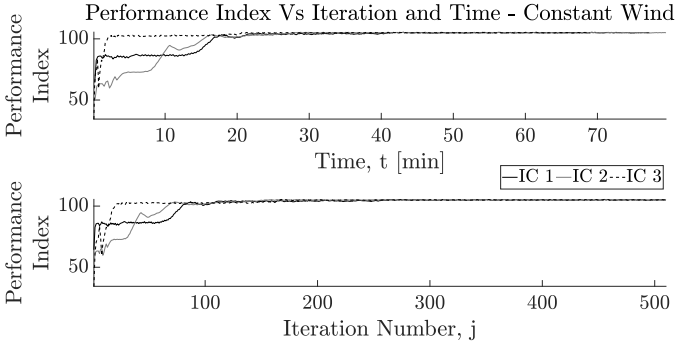

Fig. 7: Evolution of performance index for a variety of initial conditions under a constant wind profile.

In addition, Fig. 8 shows the energy augmentation ratio (EAR) given by the first term of equation (44), plotted against iteration number, j. This plot demonstrates that this optimization results in course geometries that produce significantly more power than the initial course geometry.

B. Realistic Wind Profile

In order for this ILC-based path optimization to be successful under a variable wind profile, it should produce course geometries that show significantly increased energy generation when compared to a fixed, constant course geometry. Fig. (9) shows the wind speed profile used in this work. The instantaneous wind speed in simulation is based on data obtained from the NREL National Wind Technology Center [30]. This data is sampled once every minute, but linear interpolation was used to determine intermediate values. The data selected for simulation begins on August 24, 2001, which is the beginning of the data set, and runs for a duration of two hours.

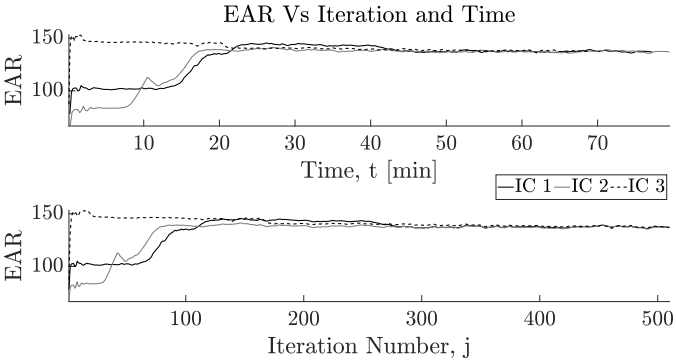

Fig. 8: Evolution of energy augmentation ratio (EAR) for a variety of initial conditions under a constant wind profile.

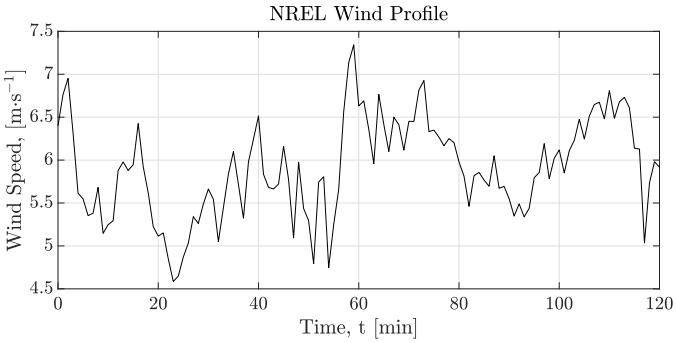

Fig. 9: Wind speed data obtained from NREL National Wind Technology Center.

The increased performance index for a variety of initial course geometries is illustrated by Fig. (10)-(12) and the increased energy generation is illustrated by Fig. (13)-(14) which show the energy augmentation ratio (EAR) as a function of both the iteration number and the associated end time of each iteration.

The performance index and energy generation ratio are compared against a simulation wherein the course was constant or fixed, and equal to the initial course geometry, over the entire duration of the simulation. Because the increased energy generation results from completing the course faster, we can see that the optimization achieves more iterations in the same amount of time as the baseline.

V. CONCLUSIONS

This paper presented an iterative learning approach for optimizing the course geometry in repetitive path-following applications. Unlike offline path optimization algorithms, the approach described herein allows for adjustment of the path from one iteration to the next, which is essential for performance-critical applications where modeling uncertainty is high. This paper demonstrated the efficacy of the proposed path adaptation approach for one such performance-critical application, namely airborne wind energy (AWE) systems executing crosswind flight.

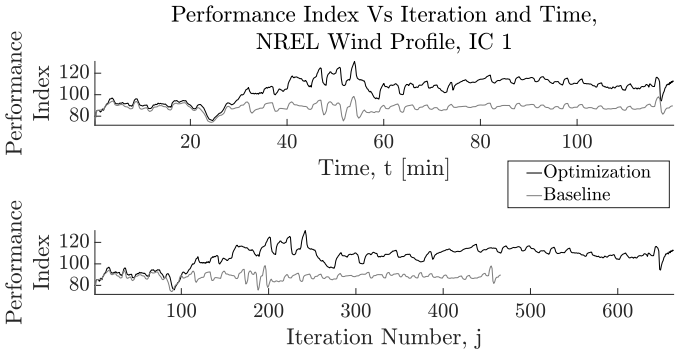

Fig. 10: Performance Index, J_j plotted against both time, t, (top) and iteration number, j, (bottom), for the optimization under the variable wind profile of Fig. (9), beginning from initial course geometry 1, where $\mathbf{b}_0 = [\,20^\circ \;\; 130^\circ\,]$.

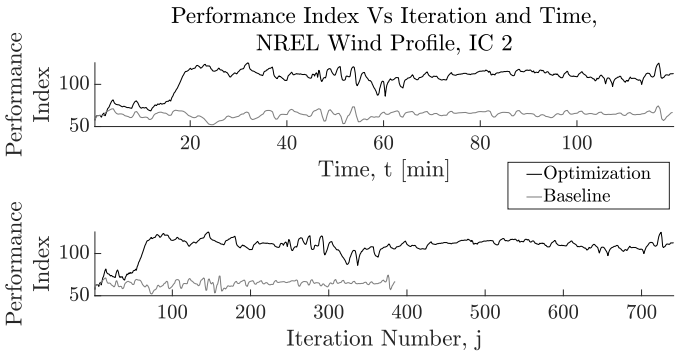

Fig. 11: Performance Index, J_j plotted against both time, t, (top) and iteration number, j, (bottom), for the optimization under the variable wind profile of Fig. (9), beginning from initial course geometry 2, where $\mathbf{b}_0 = [\,15^\circ \;\; 150^\circ\,]$.

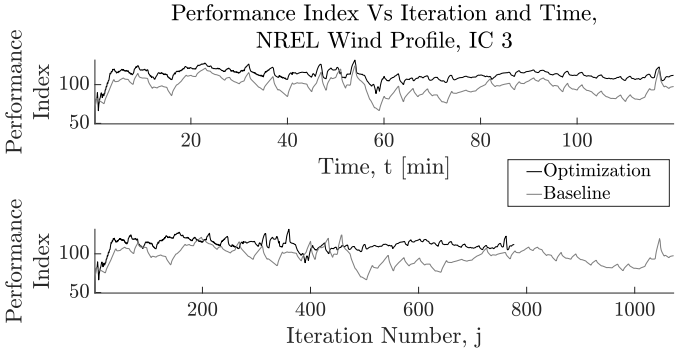

Fig. 12: Performance Index, J_j plotted against both time, t, (top) and iteration number, j, (bottom), for the optimization under the variable wind profile of Fig. (9), beginning from initial course geometry 3, where $\mathbf{b}_0 = [\,15^\circ \;\; 60^\circ\,]$.

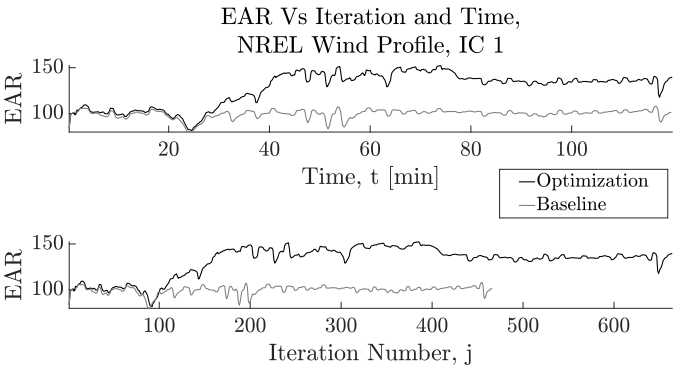

Fig. 13: Energy augmentation ratio (EAR), plotted against both time, t, (top) and iteration number, j, (bottom), for the optimization under the variable wind profile of Fig. (9), beginning from initial course geometry 1, where $\mathbf{b}_0 = [\,20^\circ \;\; 130^\circ\,]$.

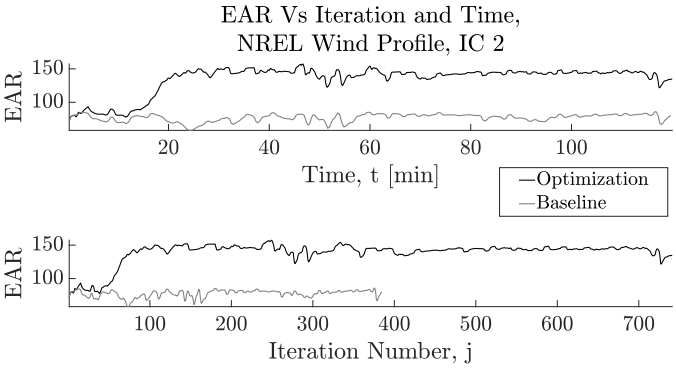

Fig. 14: Energy augmentation ratio (EAR), plotted against both time, t, (top) and iteration number, j, (bottom), for the optimization under the variable wind profile of Fig. (9), beginning from initial course geometry 2, where $\mathbf{b}_0 = [\,15^\circ \;\; 150^\circ\,]$.

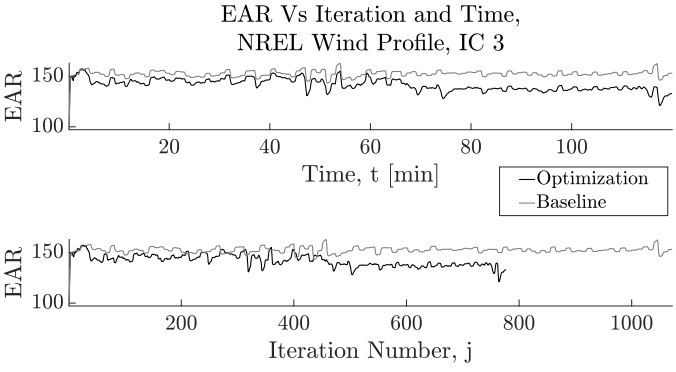

Fig. 15: Energy augmentation ratio (EAR), plotted against both time, t, (top) and iteration number, j, (bottom), for the optimization under the variable wind profile of Fig. (9), beginning from initial course geometry 3, where $\mathbf{b}_0 = [\,15^\circ \;\; 60^\circ\,]$.

ACKNOWLEDGMENT

This research was supported by National Science Foundation award number 1538369, titled "Collaborative Research: Self-Adjusting Periodic Optimal Control with Application to Energy-Harvesting Flight" and National Science Foundation award number 1727779, titled "Collaborative Research: An Economic Iterative Learning Control Framework with Application to Airborne Wind Energy Harvesting."

REFERENCES

[1] T. Chettibi, H. E. Lehtihet, M. H. S. H., 2004. "Minimum cost trajectory planning for industrial robots". *European Journal of Mechanics,* **23**(4), pp. 703–715.

[2] S. Seereeram, J. T. W., 1995. "A global approach to path planning for redundant manipulators". *IEEE Transactions on Robotics and Automation,* **11**(1), pp. 152–160.

[3] J. Holm, M. S., 2008. "Kinetic energy shaping for gait regulation of underactuated bipeds". *Proceedings of the IEEE International Conference on Control Applications*. San Antonio, TX.

[4] F. Asano, M. Yamakita, N. K. Z. L., 2004. "A novel gait generation for biped walking robots based on mechanical engineering constraint". *IEEE Transactions on Robotics and Automation,* **20**(3), pp. 565–573.

[5] U. Rosolina, A. Carvhalo, F. B., 2017. "Autonomous racing using learning model predictive control". *Proceedings of the 2017 IFAC World Congress*. Toulouse, France.

[6] M. Brunner, R. Ugo, J. G. F. B., 2017. "Repetitive learning model predictive control: An autonomous racing example". *Proceedings of the 56th Conference on Decision and Control*. Melbourne, Australia.

[7] P. Williams, B. Lansdorp, W. O., 2008. "Nonlinear control and estimation of a tethered kite in changing wind conditions". *AIAA Journal of Guidance, Control, and Dynamics,* **31**(3), pp. 973–978.

[8] G. Horn, S. Gros, M. D., 2013. "Numerical trajectory optimization for airborne wind energy systems described by high fidelity aircraft models". *Airborne Wind Energy*, pp. 205–218.

[9] Loyd, M., 1980. "Crosswind kite power". *Journal of Energy,* **4**(3), pp. 106–111.

[10] A. Zgraggen, L. Fagiano, M. M., 2013. "On real-time optimization of airborne wind energy generators". *Proceedings of the IEEE Conference on Decision and Control*. Florence, Italy.

[11] A. Zgraggen, L. Fagiano, M. M., 2015. "Real-time optimization and adaptation of the flight of tethered wings for airborne wind energy". *IEEE Transactions on Control Systems Technology,* **23**(2), pp. 434–448.

[12] M. Kehs, C. Vermillion, H. F., 2017. "Online energy maximization of an airborne wind energy generator in periodic flight". *IEEE Transactions on Control Systems Technology*(99), pp. 1–11.

[13] Altaeros Energies website. http://www.altaeros.com/.

[14] Windlift website. http://www.windlift.com/.

[15] D. Bristow, M. Tharayil, A. A., 2006. "A survey of iterative learning control". *IEEE Control Systems Magazine,* **26**(3), pp. 96–114.

[16] C. Freeman, Y. T., 2012. "Iterative learning control with mixed constraints for point-to-point tracking". *IEEE Transactions on Control Systems Technology,* **21**(3), pp. 604–616.

[17] I. Lim, K. B., 2013. "Pareto optimization-based iterative learning control". *Proceedings of the American Control Conference*. Washington, D.C.

[18] T. Wood, H. Hesse, R. S., 2017. "Predictive control of autonomous kites in tow test experiments". *IEEE Control Systems Letters,* **1**, pp. 110–115.

[19] M. Cobb, K. Barton, H. F. C. V., 2018. "Iterative learning-based waypoint optimization for repetitive path planning, with application to airborne wind energy systems". *Proceedings of the IEEE Conference on Decision and Control*. Melbourne, Australia.

[20] L. Fagiano, A. Zgraggen, M. M. M. K., 2014. "Automatic crosswind flight of tethered wings for airborne wing energy: Modeling, control design, and experimental results". *IEEE Transactions on Control Systems Technology,* **22**(4), pp. 1433–1447.

[21] Napolitano, M., 2011. *Aircraft Dynamics: From Modeling to Simulation*.

[22] Xflr5 software. http://www.xflr5.com/xflr5.htm.

[23] airfoiltools.com. .

[24] Computing euler angles from a rotation matrix. http://www.gregslabaugh.net/publications/euler.pdf.

[25] Moore, K., 1993. *Iterative Learning Control: An Overview. In: Iterative Learning Control for Deterministic Systems. Advances in Industrial Control*.

[26] M. Green, J. M., 1986. "Persistence of excitation in linear systems". *Systems and Control Letters,* **7**, pp. 351–360.

[27] J. Delgado, J. M. P., 2007. "Progressive iterative approximation and bases with the fastest convergence rates". *Computer Aided Geometric Design,* **24**(1), pp. 10–18.

[28] Hogg, R. W., Rankin, A. L., Roumeliotis, S. I., McHenry, M. C., Helmick, D. M., Bergh, C. F., and Matthies, L., 2002. "Algorithms and sensors for small robot path following". In Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Vol. 4, pp. 3850–3857 vol.4.

[29] Kiefer, J., 1953. "Sequential minimax search for a maximum". *Proceedings of the American Mathematical Society,* **4**(3), pp. 502–506.

[30] D. Jager, A. A., 2018. "NREL national wind technology center (NWTC): M2 Tower". *Proceedings of the IEEE Conference on Decision and Control*. Boulder, CO, http://dx.doi.org/10.5439/1052222.