Flexible-Time Economic Iterative Learning Control: A Case Study in Airborne Wind Energy

Mitchell Cobb[1], Maxwell Wu[2], Kira Barton[3], and Chris Vermillion[4]

Abstract— **This paper applies a unique variant of point-to-point iterative learning control, wherein total iteration times are made flexible in order to truly maximize an economic performance index, to the problem of crosswind flight control in an airborne wind energy system. The mathematical formulation relies on a lifted system representation that is specified without explicit reference to time, along with relaxed waypoint following constraints (termed *region-to-region* constraints) that ultimately allow for flexible waypoint arrival times. The approach is applied to a simplified 2D model of an airborne wind energy system that executes repetitive crosswind motions with the ultimate (economic) goal of maximizing its net energy output.**

I. Introduction

Many engineered systems, including assembly lines, pick-and-place robots, active exoskeletons, and the airborne wind energy (AWE) systems considered in the present paper, involve repetitive control. Traditional repetitive control and iterative learning control (ILC) methodologies have focused on utilizing information from previous iterations to improve *tracking* performance, ultimately driving transient errors to zero. While this is an important goal in many applications (especially manufacturing), there are a host of systems for which additional criteria, such as energy generation (or consumption), time optimality, or monetary cost are critical.

In response to the need for ILC strategies that address additional performance concerns beyond tracking, which we term *economic* objectives, a variant of ILC known as *point-to-point* ILC has emerged in the past decade (see [1], [2], [3], [4]). Point-to-point ILC only enforces tracking at particular *waypoints* (traditionally specified in both space/state variables and time), while allowing for economic objectives to be considered between waypoints. A variant of point-to-point ILC, termed *region-to-region* ILC, relaxes the equality constraints at the waypoints. Existing point-to-point and region-to-region ILC strategies typically fix both the time and spatial/state variable associated with each waypoint. Even the limited point-to-point approaches that allow for flexibility in the waypoint arrival times (see [5]) do not allow for flexibility in total iteration time; this represents a severe limitation, since any acceleration of one segment must be compensated with deceleration of another. Finally, while the recent technique of uncertain trial duration-ILC [6] does allow for flexible iteration times, existing results in uncertain trial duration ILC focus exclusively on reference tracking and do not incorporate additional economic objectives. The lack of a strategy that simultaneously addresses economic metrics and allows for flexible iteration times not only precludes consideration of time-optimal ILC problems; it precludes the consideration of *any* iterative problem for which performance is linked to total iteration time.

One emerging repetitive control application for which we seek to maximize an economic metric is the crosswind flight control of airborne wind energy (AWE) systems. AWE systems, examples of which are pictured in Fig. 1, replace conventional towers with tethers and a *lifting body* (typically a wing or aerostat), dramatically reducing material costs and enabling high-altitude operation. When a high lift/drag wing is used as the lifting body, it has been shown (first in a simplified 2D quasi-static analysis in [7], and in significant follow-on work involving higher-fidelity simulations/experiments in [8], [9], [10], [11], [12], and [13]) that dramatically increased energy production can be attained through the execution of figure-8 or circular patterns perpendicular to the direction of the prevailing wind. The real-time optimization of these crosswind flight patterns represents a repetitive control application for which the performance objective is an economic one – average net power output over the course of a lap. In fact, in previous work (see [14], [15], [16]), the authors have developed iterative learning strategies that modify the waypoints that define the figure-8 path after each lap. However, the underlying waypoint-following control laws utilized standard tracking control approaches. In the present work, we focus on tracking a prescribed set of waypoints that define a figure-8 path in a way that maximizes average lap velocity, which is correlated with the maximization of average power output.

To address the aforementioned AWE application and similarly framed path following problems with economic objectives, we formulate a flexible-time ILC control strategy that enforces waypoint following constraints while maximizing (or minimizing) an economic objective and allowing for flexible iteration times. The first step in this process, performed at each iteration, involves a path linearization of the original system dynamics along a path position, s,

This research was supported by National Science Foundation award number 1727779, entitled “Collaborative Research: An Economic Iterative Learning Control Framework with Application to Airborne Wind Energy Harvesting.”

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

[2]Maxwell Wu is a Ph.D. student at the University of Michigan maxwu@umich.edu.

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

that varies from 0 to 1 along the prescribed path. The resulting path linearized model is used to derive a lifted system representation with respect to the path position, where waypoint arrival times are implicit in the entries of the lifted matrix but not explicitly defined. The objective function to be maximized or minimized (lap-averaged flight velocity as a surrogate for power output in the AWE application) is linearized, enabling the updated control sequence to be computed as the solution to a linear program (LP). In the LP formulation, waypoint arrival constraints are cast as *inequality constraints*, which forces the path to fall within a specified region of the waypoints but does not require them to be matched exactly. Because waypoint arrival times are *implicit* in the derived lifted system matrix, the imposition of *inequality* constraints on waypoints is critical in allowing the iteration time to vary from one repetition to the next. In particular, the relaxation of equality constraints allows the system to overshoot the waypoints by some amount based on the derived lifted system representation. When the new control sequence is executed at the subsequent iteration, instead of overshooting the waypoints (and with the help of time-domain feedback), the system hits the waypoints and reduces its iteration (lap) time. Because of the computationally simple LP formulation of the ILC update, the strategy is well-suited to both discrete operation (where a pause between iterations exists) and continuous operation (where one iteration begins as soon as the previous one ends, requiring the ILC update to be performed very quickly).

The paper is organized as follows. In Section II, we lay out a general flexible-time ILC methodology for maximizing (or minimizing) an economic objective over a path that is described by waypoints. In Section III, we present the specific dynamic model and problem formulation for the AWE system. In Section IV, we present simulation results that show the effectiveness of the flexible-time ILC law on a simplified AWE system model. Conclusions are presented in Section V.

II. Flexible-Time ILC Methodology

A. Problem Formulation and General Approach

In this work, we consider waypoint-following problems with economic objectives, which take on the following form in continuous time:

$$\underset{u}{\text{maximize}} \quad J(u;x(0)) = \int_0^{t_f} h(x(t),u(t))dt, \qquad (1)$$

$$\text{subject to:} \quad \dot{x}(t) = f(x(t),u(t)) \qquad (2)$$

$$\min_t \{d(\vec{r}(t),\vec{p}(s_i))\} \leq d_t \forall s_i \in \mathbb{S}_w, \qquad (3)$$

Here, $J(u;x(0)) \in \mathbb{R}$ is the performance index (to be maximized in the above formulation), which depends on the control trajectory u and initial condition $x(0)$. Constraint (2) represents the dynamic model of the system, where $x(t) \in \mathbb{R}^n$ is the state vector, and $u(t) \in \mathbb{R}^m$ is the control input vector.

Equation (3) captures the need to track specific points along the path, termed *waypoints*. Here, $d(\cdot,\cdot)$ is the distance between the two input vectors, $\vec{r}(t) \in \mathbb{R}^l$ is the spatial position of the system in l dimensions, $\vec{p}(s) \in \mathbb{R}^l$ is the user-specified vector that describes the shape of the desired path. This path-shape vector is parameterized with respect to $s \in \mathbb{R}$. The value $d_t \in \mathbb{R}$ is the user-specified distance tolerance that describes a ball around each waypoint wherein the system is considered to have successfully reached the waypoint. Lastly, $\mathbb{S}_w$ is the user-defined set of values of s that define the set of waypoints.

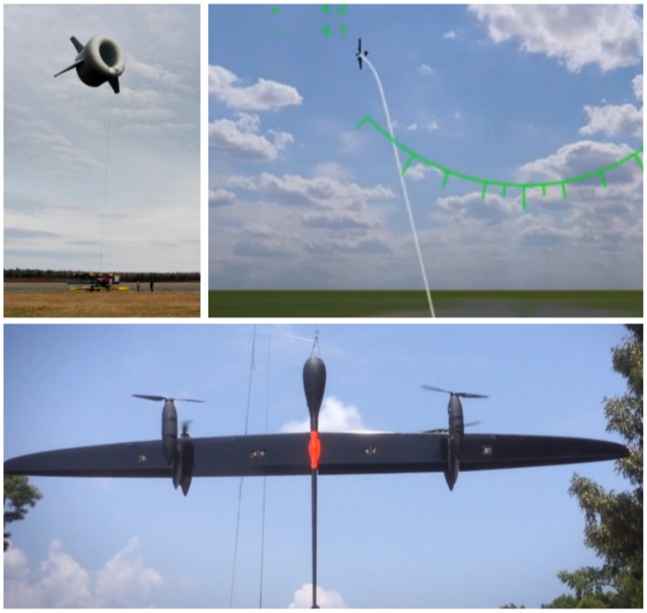

Fig. 1: Two examples of AWE system designs: Image credit Altaeros, Inc. [17] (top left), and Windlift, LLC [18] (top right, bottom). Obtained with permission.

In this work, we seek to leverage an ILC formulation that improves upon the value of $J(u;x(0))$ from one iteration to the next while allowing waypoint arrival times and total iteration duration to vary. The proposed formulation for accomplishing this is shown in Fig. 2. In this diagram, t represents time, whereas s represents a *path position*, which varies from 0 to 1 along the path. At the completion of each iteration, the ILC update receives the previous iteration's state and control trajectories, denoted by $x^k(t)$ and $u^k(t)$, respectively. The ILC update produces an updated control sequence and corresponding path, parameterized entirely in the spatial domain and denoted by u_{ff}^{k+1} and $\vec{p}(s)$, respectively. At each time instant, t, the time-domain controller must pick off the appropriate element of $\vec{p}(s)$, which is accomplished by the path-projection lookup table. Finally, a path-following feedback controller is included in the time-domain component of the control system in order to achieve closed-loop stable tracking of the prescribed path.

B. Flexible-Time ILC Update - Details

The flexible-time ILC update consists of three key steps:

1) The continuous-time nonlinear model is linearized about the state and control input sequences from the previous

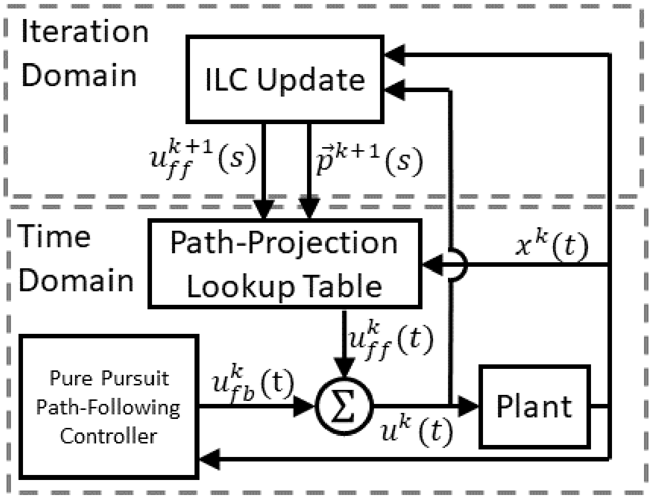

Fig. 2: Feedforward + feedback time-independent economic ILC control structure used in this work.

iteration. These sequences are parameterized with respect to the corresponding path position (s), and the resulting linearized model is a linear and *path-varying*.

2) The continuous model is discretized in the path domain, using a zero order hold, leading to a lifted system representation in the spatial domain.
3) The performance index of (1) and waypoint inequality constraints of (3) are approximated linearly, giving rise to an LP that can easily be solved with minimal computational burden.

Details of each of these three steps are provided below.

1) Path Linearization of the Dynamic Model: The goal of this step in the ILC update is twofold. First, we would like to re-parameterize the dynamic model of (2) with respect to our position along the path. Second, we would like to linearize that model around a sequence of points parameterized with respect to the path position. To do so, we must introduce measures for position along a path, which we adopt from the field of contour tracking in manufacturing processes [19].

To begin, we define the quantities *path variable* and *path position*:

Definition. *Suppose that a path in space is described by $\vec{p}(\phi)$, where $\phi \in [0 \quad 1]$ is referred to as a* path variable. *Then the corresponding* path position *is given by:*

$$s(t) = arg \min_{\phi} \{d(\vec{r}(t), \vec{p}(\phi))\}. \tag{4}$$

Thus, the path position, $s(t)$, is equal to the value of the path variable, ϕ that minimizes the Euclidean distance from our current position in space, $\vec{r}(t)$ to the path described by $\vec{p}(\phi)$.

The time-domain dynamic model of (2) can be expressed in path domain through the relationship:

$$\frac{dx}{ds} = f(x,u)\left(\frac{ds}{dt}\right)^{-1}, \tag{5}$$

where $\left(\frac{ds}{dt}\right)^{-1}$ is inverse of the rate of change of the position of system along the path. An analytical expression for this term is provided by a corollary of the implicit function theorem and given by:

$$\frac{ds}{dt} = -\frac{\partial_\phi \partial_t d(t,\phi)}{\partial_\phi^2 d(t,\phi)} \tag{6}$$

where $\partial_{\phi,t}$ denotes the partial derivative with respect to ϕ or t, and ∂^2 is the corresponding second partial derivative. If we then define the vector-valued function $g(x,u)$, where:

$$g(x,u) \triangleq -f(x,u)\frac{\partial_\phi^2 d(t,\phi)}{\partial_\phi \partial_t d(t,\phi)}, \tag{7}$$

then (5) can be written concisely as $x' = g(x,u)$ where $'$ denotes the derivative with respect to path position, s. Note that this expression $g(x,u)$ implicitly depends on the choice of path shape, $\vec{p}(\phi)$. Defining a nominal sequence of state vector and control inputs parameterized with respect to the path position as $x_0 \triangleq x|_{\substack{x=x_0(s)\\u=u_0(s)}}$ and $u_0 \triangleq u|_{\substack{x=x_0(s)\\u=u_0(s)}}$, we can then calculate an approximation of our path-parameterized dynamic model by linearizing $g(x,u)$. The resulting linearized model is given by:

$$\delta x'(s) = A(s)\delta x(s) + B(s)\delta u(s), \tag{8}$$

where:

$$\begin{aligned} \delta x'(s) &\triangleq x'(s) - g(x_0(s), u_0(s)), & (9)\\ \delta x(s) &\triangleq x(s) - x_0(s), & (10)\\ \delta u(s) &\triangleq u(s) - u_0(s), & (11) \end{aligned}$$

and the path-varying matrices $A(s)$ and $B(s)$ are defined as:

$$A(s) \triangleq \big(\nabla_x g(x,u)\big)|_{\substack{x=x_0(s)\\u=u_0(s)}} \quad B(s) \triangleq \big(\nabla_u g(x,u)\big)|_{\substack{x=x_0(s)\\u=u_0(s)}}. \tag{12}$$

2) Discretization of the Path-Parameterized Linearized Model: To obtain a lifted system representation that is useful for an ILC update, the model of equation (8) is discretized spatially using a zero order hold approximation. The sequences of discrete path variable matrices, $\{A_1, A_2, ..., A_N\}$ and $\{B_1, B_2, ..., B_N\}$, where N is the number of discrete steps along the path, are then given by:

$$A_q = e^{A(s_q)\Delta_s}, \tag{13}$$

$$B_q = \int_0^{\Delta_s} e^{A(s_q)\tau_s} B\big(s_q\big) d\tau_s, \tag{14}$$

where Δ_s is the user-specified path discretization level and s_q represents the path position at step q. The result is a discrete-path approximation given by:

$$\delta x_{q+1} = A_q \delta x_q + B_q \delta u_q, \tag{15}$$

where q refers to a discrete index along the path.

Given these discrete-path dynamics, it is now straightforward to derive a linear lifted system representation that relates the full sequence of control inputs to the full sequence of resulting states. To do this, we define these sequences as:

$$\begin{aligned} \delta \underline{x} &\triangleq [\ \delta x_0^T \ \ \dots \ \ \delta x_N^T\ ]^T, & (16)\\ \delta \underline{u} &\triangleq [\ \delta u_0^T \ \ \dots \ \ \delta u_N^T\ ]^T. & (17) \end{aligned}$$

Using this notation, with $\delta \underline{x}_0 = 0$, $\delta \underline{x}$ and $\delta \underline{u}$ are related by:

$$\delta \underline{x} = G \delta \underline{u}, \tag{18}$$

where the $n \times m$ block elements in G are given by:

$$G_{i,j} = \begin{cases} 0^{n \times m}, & i < j \\ B_i, & i = j \\ A_i A_{i-1} \ldots A_j B_i, & \text{otherwise.} \end{cases} \tag{19}$$

3) Economic ILC Update as a Linear Program: In this work, the feedforward control sequence, $\underline{u}_{ff}^{k}(s)$, is updated according to the following law:

$$\underline{u}_{ff}^{k+1}(s) = \underline{u}_{ff}^{k}(s) + \delta \underline{u}^{*}(s), \tag{20}$$

where $\delta \underline{u}^{*}(s)$ serves the role of a *correction* to the previous iteration's feedforward control sequence and is based on the solution to a linear program (LP). To compute $\delta \underline{u}^{*}(s)$, the objective function is linearized with respect to both the control and state sequences, resulting in the approximation:

$$\delta J(\delta \underline{x}^{k}) = \Psi_x \delta \underline{x} + \Psi_u \delta \underline{u}, \tag{21}$$

where:

$$\Psi_x = \nabla_{\underline{x}} J(\delta \underline{x}^{k}), \tag{22}$$

$$\Psi_u = \nabla_{\underline{u}} J(\delta \underline{u}^{k}). \tag{23}$$

The optimized correction in the control sequence is then computed according to the following LP:

$$\delta \underline{u}^{*} = \arg \max_{\delta \underline{u}} \quad \Psi_x \delta \underline{x}^{k} + \Psi_u \delta \underline{u}^{k} \tag{24}$$

$$\text{subject to:} \quad -\underline{\Delta}_{tol} \leq \Psi_w \delta \underline{x} \leq \underline{\Delta}_{tol} \tag{25}$$

$$\underline{b}_{lo} \leq \delta \underline{u} \leq \underline{b}_{hi}, \tag{26}$$

where Ψ_w is a diagonal matrix with ones in diagonal entries corresponding to values of s where waypoint tracking inequality constraints are to be imposed and zeros everywhere else.

Remark. *The imposition of inequality constraints, rather than equality constraints, for waypoint tracking, is significant. Because the lifted system representation depends upon the rate at which the path is traversed ($\frac{ds}{dt}$), the lifted system matrix implicitly (but not explicitly) depends on waypoint arrival times. In order to render waypoint arrival times flexible, the waypoints themselves must be made flexible. Furthermore, because it is well known that an LP will, in all but limiting special cases, result in a boundary solution, saturation constraints on $\delta \underline{u}$ are important. The result of this implementation is an ILC update that "nudges" the control sequence in the economically optimal direction at each iteration.*

C. Time Domain Path Projection and Path-Following Feedback Controller

Because $\underline{u}_{ff}^{k}$ is parameterized with respect to the path variable, s, not time, it is necessary to introduce a path projection that picks off the relevant element of $\underline{u}_{ff}^{k}$ at every time instant, t. This is done in two stages:

1) First, the closest point along the path provided by the ILC update, denoted by $\vec{p}^{\,k}(s)$, is determined.
2) Secondly, the corresponding value of $u_{ff}^{k}(s)$ is determined via interpolation.

Because the dynamics of path following operations typically involve pure integrators (leading to an open-loop system that is input-output unstable in the time domain), a stabilizing path-following controller becomes necessary. For the AWE system considered in this work (detailed in the next section), a pure pursuit path following controller is used to accomplish this goal.

III. AWE-Specific Problem

As noted in the introduction, AWE systems whose lifting bodies possess high lift/drag ratios can execute crosswind flight patterns that dramatically enhance power capture [7]. This arises from the fact that the wind presented to the airborne turbine(s) results from a combination of the system's motion and the naturally occurring wind. In fact, it is the vector difference of the true wind velocity vector and the wing's velocity, known as the *apparent wind velocity* and defined as $\vec{v}_{app} \triangleq \vec{v}_{wind} - \vec{v}$, that dictates the power production of the system. Fig. 3 illustrates this concept in a simple 2D quasi-static analysis. This concept has been shown through numerical models and experiments (see [8], [9], [10], [11], [12], and [13]) to translate to 3D flight. Because power production is proportional to apparent wind speed *cubed*, even small increases in apparent wind speed can result in significant increases in power.

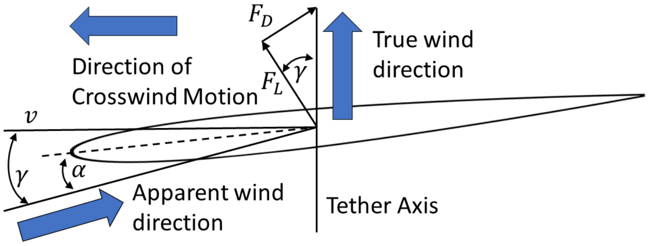

Fig. 3: Summary of the key forces and velocities at play in crosswind flight, based on a simplified 2D quasi-static analysis. In particular, crosswind flight with a high lift/drag (L/D) ratio results in a large ratio of crosswind flight speed (v) to wind speed (v_{wind}), leading to a large apparent wind speed (v_{app}).

In practice, crosswind flight patterns are executed through repetitive circular or figure-8 motions, as illustrated in Fig. 4. Given the correlation between crosswind flight speed and power production, there exists a significant interest in maximizing flight velocity over the course of a lap. The velocity maximization objective must be balanced against the need to achieve stable, well-controlled flight, which is encoded in the requirement that the system reach a specified set of waypoints. With fixed waypoints, increasing flight velocity typically requires reduced lap time, thereby requiring a flexible-time formulation. Thus, we have a system wherein we would like to reach specific waypoints, but in between those waypoints, we would like to maximize an economic metric (average flight velocity, which acts as a surrogate for energy generation). Furthermore, we have the opportunity to leverage information from each past figure-8 cycle ("lap") to inform the actions at the next lap. This conforms precisely to the economic ILC formulation considered in this paper.

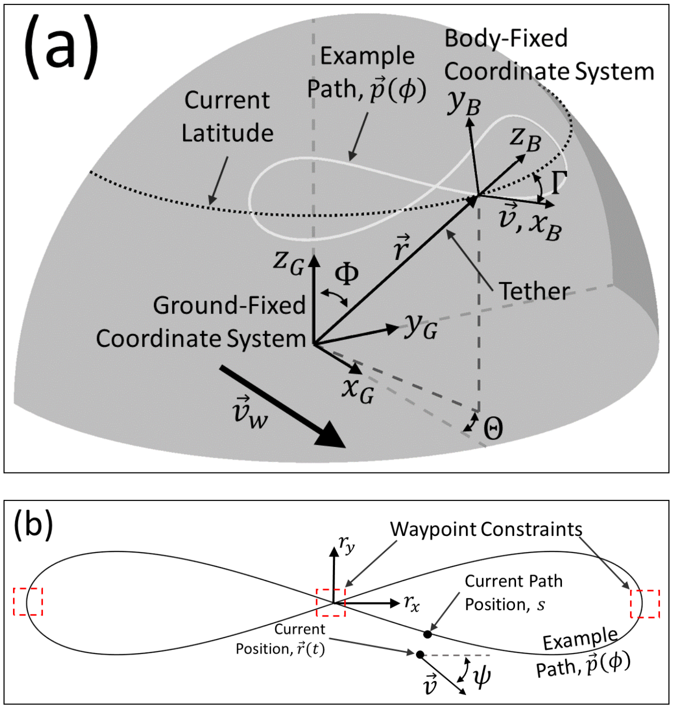
(a)
Example Path, $\vec{p}(\phi)$
Body-Fixed Coordinate System
Current Latitude
y_B
z_B
Γ
$\vec{v}, x_B$
$\vec{r}$
Tether
z_G
Φ
Ground-Fixed Coordinate System
y_G
x_G
$\vec{v}_W$
Θ
(b)
Waypoint Constraints
r_y
r_x
Current Path Position, s
Current Position, $\vec{r}(t)$
Example Path, $\vec{p}(\phi)$
$\vec{v}$
ψ

Fig. 4: Schematic depiction of crosswind flight, in full 3D (top) and planar projection (bottom). The quantities r_x and r_y on the bottom diagram represent arc lengths in the Θ and $-\Phi$ directions of the top diagram, respectively.

A. Simulation Model

The simulation model used in this work, illustrated in Fig. 4, is developed as a simplified analog that captures the essential features of the optimization problem for the full AWE system while simplifying complex nonlinear dynamics of more complicated models.

In particular, we consider motion along the surface of a sphere (assuming the tether remains taut and maintains a constant length). We consider motion in two directions: the azimuthal direction, denoted in Fig. 4 (a) through the angle Θ and in Fig. 4 (b) through the arc length r_x, and the elevation direction, denoted in Fig. 4 (a) through the angle $-\Phi$ and in Fig. 4 (b) through the arc length r_y. We model the AWE system's flight speed according to a *velocity polar*, which encodes the steady-state flight speed as a function of the system's heading relative to the wind (denoted by ψ). A velocity polar for the AWE system, depicted for a wind speed of 3 $\frac{m}{s}$, is shown in Fig. 5. The radial distance of the polar curve represents the steady-state flight velocity for any particular heading angle. Because the wing under consideration is symmetric, the polar curve is symmetric about the vertical axis. The control input for the system is the heading setpoint, denoted by u; it is assumed that a lower-level controller manipulates control surfaces to drive this heading to its setpoint.

Ultimately, the above approximations result in a simplified 2D model, consisting of four states: two for position, one for the speed of the system, and one for the heading of the system:

$$\begin{aligned} \dot{r}_x &= v\cos(\psi), & \dot{v} &= \frac{1}{\tau_v}(V_{SS}(\psi) - v), \\ \dot{r}_y &= v\sin(\psi), & \dot{\psi} &= \frac{1}{\tau_\psi}(u - \psi). \end{aligned} \tag{27}$$

In (27), v represents the speed of the AWE system, $V_{SS}(\psi)$ represents the steady-state speed for a particular heading, as obtained from the velocity polar, τ_v represents a lumped time constant associated with the translational dynamics, and τ_ψ represents a lumped time constant associated with the closed-loop rotational dynamics.

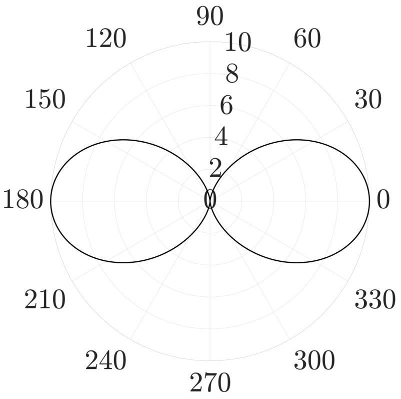

Fig. 5: Velocity polar for a 3 m/s wind speed: Position along the radial axis corresponds to the maximum achievable flight speed in m/s, and position along the angular axis corresponds to the heading (ψ) in degrees.

B. Lower-Level Path Following Controller

The stabilizing controller is implemented as a pure-pursuit path following controller. At every instant in time, it performs three steps in order to calculate the feedback component of the control signal, $u_{fb}(t)$. First, the controller solves the one-dimensional minimization problem given in (4) to obtain the current position along the path, s. It then adds a constant, user-specified amount, s_t, to this value to obtain the path parameter of the "target" point along the path, ϕ^*, where $\phi^*(t) = s(t) + s_t$. Finally, the feedback element of the control signal is calculated as the angle between the target point, $\vec{p}(\phi^*)$ and the current position, $\big(r_x(t), r_y(t)\big)$. The one-dimensional minimization problem is solved in this work using the golden section method.

The path chosen for the AWE application comprises a figure 8 and is defined mathematically according to:

$$\vec{p}(\phi) = \left[\tfrac{W}{2}\cos\left(\left(2\phi + \tfrac{3}{2}\right)\pi\right) \quad \tfrac{H}{2}\sin\left(2\left(2\phi + \tfrac{3}{2}\right)\pi\right)\right]^T, \tag{28}$$

where W and H are the overall width and height of the path.

C. AWE-Specific Economic ILC Update

For the AWE application, at every iteration, the updated control signal is obtained by solving an economic optimization problem that encodes both the desire to generate more energy and the requirement that the system reach the waypoints. Thus, the economic optimization problem used to calculate the updated feedforward control input is given by the solution to the following LP, which is analogous to the general formulation from Section II:

$$\delta\underline{u}^* = \arg\max_{\delta\underline{u}} \quad \Psi_v \delta\underline{x} \tag{29}$$

$$\text{subject to:} \quad -\underline{\Delta}_{tol} \leq \Psi_w \delta\underline{x} \leq \underline{\Delta}_{tol} \tag{30}$$

$$\underline{b}_{lo} \leq \delta\underline{u} \leq \underline{b}_{hi}, \tag{31}$$

$$\psi_{min} \leq \delta\psi_N + \psi_N^k \leq \psi_{max}, \tag{32}$$

where $\Psi_v = \begin{bmatrix} \Psi_{v1} & \dots & \Psi_{vN} \end{bmatrix}$ and $\Psi_{vi} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$ for every i. In other words, Ψ_v picks off the iteration-to-iteration deviations in velocities (δv) from the state variables and sums them, producing a spatially-averaged velocity deviation (to be maximized) as a linear performance metric.

The first inequality constraint encodes a range of admissible values for the states at the user-specified waypoints. The vector $\underline{\Delta}_{tol} = \begin{bmatrix} \Delta_{x,1} & \Delta_{y,1} & \dots & \Delta_{x,n_w} & \Delta_{y,n_w} \end{bmatrix}^T$ encodes the allowable deviations of states from waypoints where n_w is the number of waypoints specified by the user and $\Delta_{x,i}, \Delta_{y,i} \in \mathbb{R}$ define the "box" around each waypoint that the system is constrained to reach. The matrix Ψ_w is a block matrix designed to pick off the δx and δy state values from the lifted state vector $\delta\underline{x}$ at the specified waypoints. Specifically, denoting $\mathbb{S}_w$ as the set of spatial indices corresponding to waypoints, the elements of Ψ_w, denoted by $w_{i,j}$ are given by:

$$w_{i,j} = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, & s_j \in \mathbb{S}_w \\ 0^{l \times n}, & \text{otherwise,} \end{cases} \tag{33}$$

where $i = 1,\ 2,\ \dots,\ n_w$ and $j = 1,\ 2,\ \dots,\ N$.

The vectors $\underline{b}_{lo}$, $\underline{b}_{hi}$ define the allowable limits on deviation in the control input from one iteration to the next. For this work, we restrict the change in the control input to be constant. That is, $-\underline{b}_{lo} = \underline{b}_{hi} = \begin{bmatrix} \delta u_{lim} & \delta u_{lim} & \dots \end{bmatrix}^T$ where $\delta u_{lim} \in \mathbb{R}$ is a user-defined constant.

The final constraint ensures that the heading at the completion of each lap lies within a prescribed tolerance.

IV. Simulation Results and Discussion

In order to assess the performance of the AWE system, we look at two quantities, the value of the performance index from iteration to iteration and a direct measure of energy capture. Table I provides the numerical values of critical parameters used in simulation.

Variable	Description	Value	Units
W	Path width	50	m
H	Path height	15	m
V_{ss}	Max steady state speed	10	m/s
τ_v	Speed time constant	1	s
τ_Ψ	Heading time constant	0.1	s
$\mathbb{S}_w$	Waypoint path parameters	$\{0.25, 0.5, 0.75, 1\}$	-
Δ_x	Waypoint x tolerance	0.25	m
Δ_y	Waypoint y tolerance	0.25	m
Δ_ψ	Final allowable heading error	25	deg
δu_{lim}	Max allowable ctrl. change	1	deg
Δ_s	Path discretization level	0.005	-
v_{wind}	Wind speed	4	m/s
s_t	Target lead distance	0.02	-

TABLE I: Parameter values used in simulation results.

The performance index as a function of iteration (figure-8 lap) is shown in Fig. 6. To quantify energy production in a non-dimensional manner that lends insight into the benefit of crosswind motion, we use a metric that we term the *energy augmentation ratio* (EAR). This number is a ratio of the amount of energy generated by the system under crosswind flight, to the amount of energy that the same system would have generated if it had been motionless. Over a single lap, denoted by the index z, it is calculated as:

$$EAR = \frac{\int_{t_{i,z}}^{t_{f,z}} |\vec{v}_{wind}(t) - \vec{v}(t)|^3 dt}{\int_{t_{i,z}}^{t_{f,z}} |\vec{v}_{wind}(t)|^3 dt}, \tag{34}$$

where the $t_{i,z}$ and $t_{f,z}$ are the initial and final times of the z^{th} iteration respectively. The term in the denominator is proportional to the amount of energy generated by a turbine pointed directly into the wind in a stationary system, whereas the term in the numerator is proportional to (with the same proportionality constant) the amount of energy produced by the system under crosswind flight. The quantity $\vec{v}_{wind}(t) - \vec{v}(t)$ is commonly referred to as apparent wind, which is the wind actually presented to the turbine during crosswind flight. The progression of EAR as a function of iteration number is shown in Fig. 7. A comparison of Figs. 6 and 7 confirms that the average velocity-based performance index used as a surrogate economic metric in this work indeed correlates extremely well with the EAR. This is unsurprising considering that energy augmentation is driven by apparent wind speed, which is heavily driven by crosswind flight speed, v. Faster crosswind flight almost always correlates with greater energy generation. Most importantly, it can be seen from these figures that the performance is indeed improving (and converging) as the iterations progress.

Fig. 8 shows the progression of flight paths at selected iterations. This figure shows that small changes in the flight path between waypoints can lead to significant (nearly 20 percent) additional energy augmentation.

Finally, Fig. 9 provides the instantaneous value of $\frac{|\vec{v}_{wind}(t) - \vec{v}(t)|^3}{|\vec{v}_{wind}(t)|^3}$, which can be termed the *power* augmentation ratio, over the course of a lap. In addition to the later laps exhibiting a higher average power augmentation ratio, it can be observed that these laps are completed in a shorter amount of time (approximately 90 percent of the time taken in the initial lap). This underscores the importance of a flexible-time ILC update strategy.

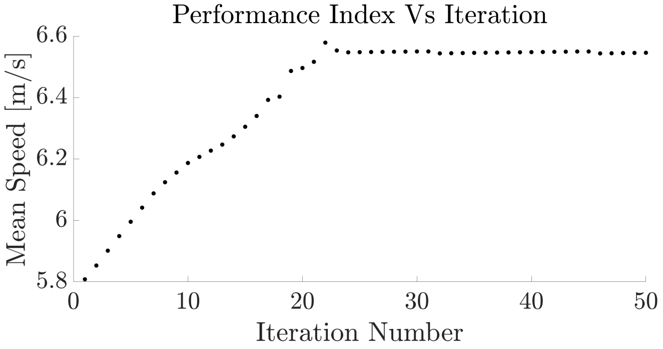

Fig. 6: Performance index vs. iteration number.

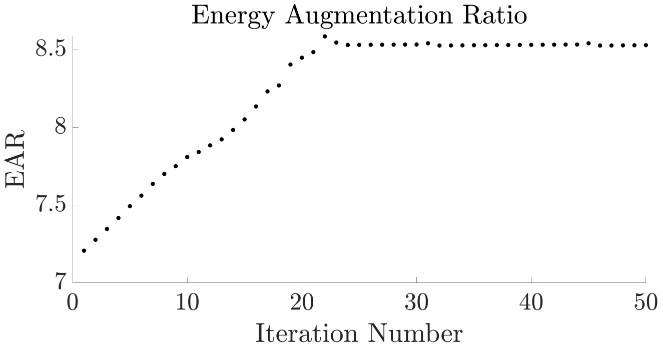

Fig. 7: EAR vs. iteration number, as calculated by (34).

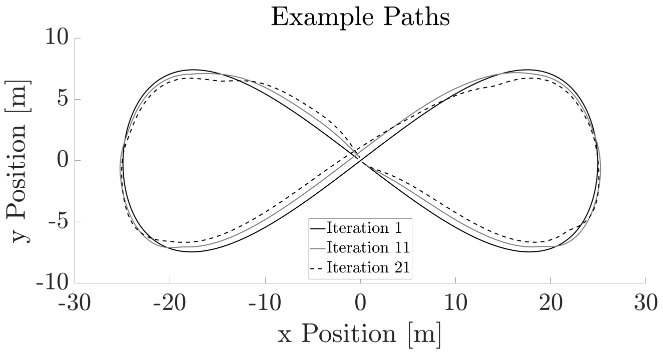

Fig. 8: Example figure 8 path at 1^{st}, 11^{th}, and 21^{st} iteration.

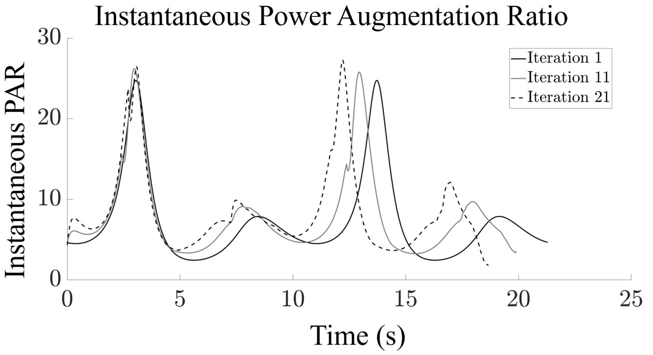

Fig. 9: Instantaneous PAR at 1^{st}, 11^{th}, and 21^{st} iteration.

V. Conclusions

In this paper, we presented a flexible-time ILC formulation wherein the ILC update is performed in an entirely spatial domain, for the purposes of maximizing (or minimizing) an economic metric. Flexibility in total iteration time is achieved through the relaxation of waypoint following constraints. The framework was then applied to a simplified 2D model of an AWE system, where we demonstrated in simulation that the approach leads to significantly increased energy generation.

References

[1] C. Freeman and Y. Tan, "Iterative learning control with mixed constraints for point-to-point tracking," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 604–616, 2012.

[2] C. Freeman, "Constrained point-to-point iterative learning control with experimental validation," *Control Engineering Practice*, vol. 20, no. 5, pp. 489–498, 2012.

[3] C. Freeman, Z. Cai, E. Rogers, and P. Lewin, "Iterative learning control for multiple point-to-point tracking application," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 590–600, 2010.

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

[5] Y. Chen, B. Chu, and C. Freeman, "Spatial path tracking using iterative learning control," *IEEE Conference on Decision and Control*, 2016, Las Vegas, NV.

[6] P. M. Sammons, D. Hoelzle, and K. Barton, "Time scale transformed ILC for a class of nonlinear systems with uncertain trial duration," *IEEE Transactions on Control Systems Technology (submitted - under review)*.

[7] M. Loyd, "Crosswind kite power," *Journal of Energy*, vol. 4, no. 3, pp. 106–111, 1980.

[8] P. Williams, B. Lansdorp, and W. Ockels, "Nonlinear control and estimation of a tethered kite in changing wind conditions," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 31, no. 3, pp. 973–978, 2008.

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

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

[11] A. Zgraggen, L. Fagiano, and M. Morari", "Real–time optimization and adaptation of the flight of tethered wings for airborne wind energy," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 434–448, 2015.

[12] M. Kehs, C. Vermillion, and H. Fathy, "Online energy maximization of an airborne wind energy generator in periodic flight," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–11, 2017.

[13] 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," *ASME Journal of Dynamic Systems, Measurement, and Control*, 2018.

[14] M. Cobb, K. Barton, H. Fathy, and C. Vermillion, "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*, 2018, Melbourne, Australia.

[15] M. Cobb, K. Barton, H. Fathy, and C. Vermillion", "An iterative learning approach for online path optimization for tethered energy systems undergoing cyclic spooling motion," *Proceedings of the American Control Conference (accepted)*, 2019, Philadelphia, PA.

[16] M. Cobb, K. Barton, H. Fathy, and C. Vermillion, "Iterative learning-based path optimization for repetitive path planning, with application to 3D crosswind flight of airborne wind energy systems," *Transactions on Control System Technology*, submitted, under review.

[17] Altaeros Energies. [Online]. Available: http://www.altaeros.com/

[18] Windlift. [Online]. Available: http://www.windlift.com/

[19] A. G. Ulsoy and Y. Koren, "Control of machining processes," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 115, 1993.