Iterative Learning-Based Path Optimization with Application to Marine Hydrokinetic Energy Systems

Mitchell Cobb[1], James Reed[2], Joshua Daniels[3], Ayaz Siddiqui[4], Max Wu[5],
Hosam Fathy[6], Kira Barton[7], and Chris Vermillion[8]

***Abstract*—This paper presents an iterative learning control-based approach for optimizing the flight path geometry of a tethered marine hydrokinetic energy (MHK) system. Tethered MHK systems, which replace the tower and turbine of a conventional system with a tether and a lifting body, capture energy by driving a generator with the tension in the tether. By spooling out tether during the high tension portions of cross-current flight and spooling in during low tension portions, net positive energy is generated over one cycle. Because the net energy generation is sensitive to the shape of the flown path, we employ an iterative learning update law to adapt the path shape from one lap to the next. Additionally, we present a realistic system model, along with lower-level path-following and power take-off (PTO) controllers. We then demonstrate the efficacy of our algorithm on this model in both uniform and realistic flow environments.**

I. Introduction

Traditional devices for capturing energy from wind or ocean currents consist of a turbine mounted on a large tower. While these have proven effective and reliable, they are often suboptimal. Because turbines are fixed in space, they are usually not located at the best point in the flow. Often, the winds or currents with the most available energy are located at very high altitudes or close to the surface in very deep parts of the ocean. It is these areas that represent the greatest potential for energy generation. Work in [1] showed that the wind energy available at 600m was often five times that of the wind available to most towered systems and for ocean current systems. A study in [2] showed that there was more energy present in the Gulf Stream off the coast of North Carolina than the entire energy demand of the state. Unfortunately, the locations with the most available Gulf Stream energy often have depths greater than 200 m. At these depths, the capital costs associated with tower construction render it infeasible.

Tethered energy systems, examples of which are shown in figure 1, represent one possible solution to these challenges. In both the airborne wind energy (AWE) systems and the marine hydrokinetic energy (MHK) systems, the tower is replaced by a tether (or tethers) and the turbine is replaced with a high lift-to-drag kite. Energy capture is then accomplished via either *fly-gen*, where turbines are mounted on the kite, or *ground-gen* where the tension in the tether is used to drive a generator. In this work, we focus on the latter method. In this method, the tether is spooled out under high tension, generating positive power, and then spooled in under low tension, consuming power. Because the power generated during spool out is much larger than the power consumed during spool in, the result is net positive energy generation over one cycle.

In addition to reducing capital costs by eliminating the need to build a rigid tower, tethered energy systems present two opportunities. First, the system can be repositioned to the optimal location in the flow profile. Second, the kite can be flown perpendicular to the prevailing flow. This *crosswind* or *cross-current* flight, initially examined in [3], has been shown to significantly increase power generation.

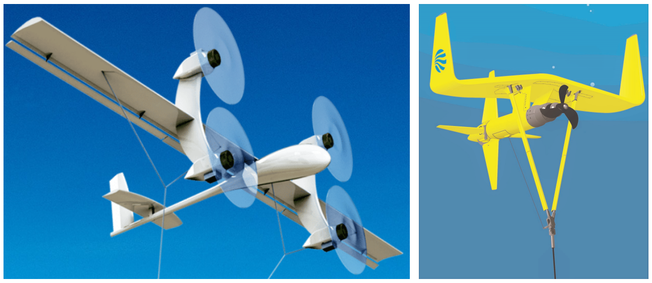


Figure 1: Examples of tethered energy systems, both the AWE system from Windlift [4], left, and the MHK system from Minesto [5] employ on-board turbines to harvest energy. Image credits: Windlift (left) and Minesto (right).

Given the limitations of existing dynamic modeling tools, [6], [7], and [8] have investigated on-line adaptation of some attributes of the crosswind flight. Specifically, [6] and [7] seek to optimize the flight path *location* whereas this work seeks to optimize the flight path *shape*. Additionally, [8] optimizes a state setpoint trajectory, whereas this work seeks to optimize a path defined only in terms of spatial coordinates, with no reference to time. Furthermore, none of the aforementioned works consider path optimization in the context of a full spool-in, spool-out cycle.

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

[2] James Reed is an a PhD candidate at North Carolina State University jcreed2@ncsu.edu.

[3] Joshua Daniels is a research assistant at North Carolina State University jldanie5@ncsu.edu.

[4] Ayaz Siddiqui is a PhD candidate at North Carolina State University asiddiq2@ncsu.edu.

[6] Max Wu is a PhD candidate at The University of Michigan maxwu@umich.edu.

[7] Hosam Fathy is a Professor in the Department of Mechanical Engineering at The University of Maryland hfathy@umd.edu.

[8] Kira Barton is an Associate Professor in the Department of Mechanical Engineering at The University of Michigan bartonkl@umich.edu.

[9] 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. and Windlift, LLC.

Because of the highly repetitive nature of the power generation cycle, there exists a significant opportunity for improvement or learning from one cycle to the next. In applications such as the one considered in this work, where the existing dynamic modeling tools have large parametric or nonparametric uncertainties, the repetitive method of operation represents one method for circumventing the challenges associated with model-based control and optimization. To address these challenges and opportunities, the authors proposed an algorithm in [9] and then developed it in [10] and [11]. This algorithm consists of two key steps:

1) metamodel identification, in which a recursive least squares fit is performed to approximate the relationship between flight path shape and performance, and
2) an iterative learning update, in which information from past iterations is used to select the flight path geometry used in the next cycle.

Although work by these authors in a prior conference publication ([11]) did consider a tethered MHK system, it utilized an artificially simplified dynamic model and only investigated spatiotemporally constant flow profiles. This work extends and build upon that work. It differs in two crucial ways.

- First, this work employs a significantly higher fidelity plant model, without any of the artificial simplifications or kinematic constraints that were essential in the prior work.
- Second, this work achieves energy capture using an *intracycle* spooling strategy wherein the tether is reeled out during the high-tension portions of the path, and reeled in during the low-tension portions of the path. Prior work used an *intercycle* spooling strategy wherein multiple high-tension cross-current laps were performed before re-configuring the system to achieve low tether tension and then spooling in. In an MHK application where the flow resource is characterized by a significant shear profile (variation of flow speed with vertical position), such a strategy enables the system to remain at a relatively consistent depth while also avoiding the need to periodically suspend and reinstate cross-current motion.

This work makes three core contributions.

- We show that when the artificial kinematic constraints are lifted, and an improved, realistic, 6DOF control-oriented dynamic model is used, our algorithm continues to work effectively.
- We show that even when implemented on an improved, realistic, 6DOF control-oriented dynamic model, in a realistic 3-D spatiotemporally varying flow field incorporating both low-frequency flow variations and high-frequency turbulence, our algorithm works effectively.
- We present an iterative learning, intracycle power take-off (PTO) controller designed to spool out during the high tension portions of the path, and spool in during the low tension portions, all while preventing any net change in tether length from iteration to iteration such that areas of high current velocities in a spatiotemporally varying flow environment can be targeted.

II. Plant Model

As shown in figure 2, a crosscurrent, tethered MHK system is comprised of three elements,

- a ground station, which houses a winch-generator system and can be either floating, or fixed to the sea floor,
- a flexible tether or tethers, and
- a lifting body, which flies mostly perpendicular to the prevailing current.

By flying perpendicular to the prevailing flow, the apparent flow, which is the combined effect of the prevailing flow and the velocity of the system, can be much larger than the prevailing flow alone. The result is larger hydrodynamic forces, larger tether tensions, and more net power production.

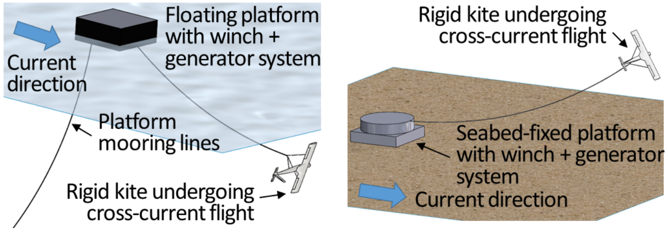

Floating platform with winch + generator system
Current direction
Platform mooring lines
Rigid kite undergoing cross-current flight
Rigid kite undergoing cross-current flight
Seabed-fixed platform with winch + generator system
Current direction

Figure 2: A diagram showing a tethered MHK system deployed from both a floating platform (left) and the sea floor (right).

Figure 3 shows two possible tether spooling strategies that can be used to extract energy. Intercycle spooling, shown on the left of figure 3, performs several cycles of crosscurrent flight before switching the kite to a low-tension configuration and spooling tether in. Intracycle spooling, shown on the right of figure 3, leverages the natural highs and lows that occur in the tether tension during one cycle by spooling out during the naturally high tension portions and in during the naturally low tension portions.

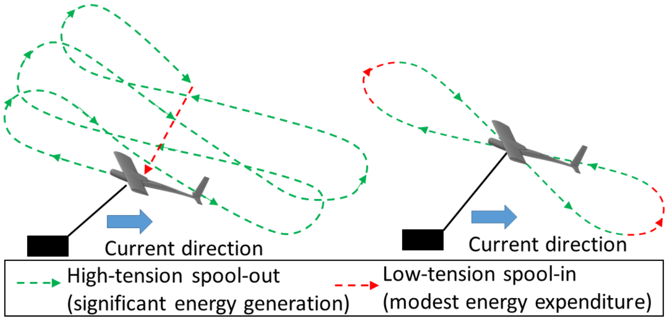

Current direction
Current direction
High-tension spool-out (significant energy generation)
Low-tension spool-in (modest energy expenditure)

Figure 3: A diagram showing two possible tether spooling strategies, intercycle spooling (left) and intracycle spooling (right).

The dynamic model used in this work has been developed in prior work by the authors in [12], but is included in its entirety here for completeness. The MHK system is modeled as a combination of three elements:

1) A rigid lifting body wherein forces and moments are calculated from lift, drag, buoyancy, and gravity.

2) A tether model comprised of a series of non-compressive spring-dampers subject to fluid dynamic drag, buoyancy, and gravity. One end of the tether is attached to the lifting body and exerts a force on it, and the other is fixed to the ocean floor.
3) A winch, modeled as a simple combination of first order filters, rate limiters, integrators and lumped efficiency factors, which is used to estimate instantaneous power production.

Specific values of plant parameters used in simulation are summarized at the end of the section in table I.

A. Preliminaries: Coordinate Systems & Important Quantities

The plant model and lower-level controller are described using three coordinate systems which are shown in figure 4.

- The Cartesian ground frame coordinate system $\overline{G}$, which is taken to be an inertial reference frame, defined by the origin point, G, and the unit vectors $\vec{i}_{\overline{G}}$, $\vec{j}_{\overline{G}}$, and $\vec{k}_{\overline{G}}$.
- The body fixed coordinate system $\overline{B}$, which is defined by a point at the center of mass of the kite, B and the unit vectors $\vec{i}_{\overline{B}}$, $\vec{j}_{\overline{B}}$, and $\vec{k}_{\overline{B}}$. The $\vec{i}_{\overline{B}}$ unit vector points backwards along the fuselage, and the $\vec{j}_{\overline{B}}$ unit vector points towards the starboard. The unit vector $\vec{k}_{\overline{B}}$ is then calculated as the cross product $\vec{i}_{\overline{B}} \times \vec{j}_{\overline{B}}$.
- The tangent frame coordinate system $\overline{T}$, which is centered at the point B but uses the spherical unit vectors associated with the ground frame coordinate system. Specifically, the unit vectors of this coordinate system are calculated at a generic spatial position, $\vec{r} \in \mathbb{R}^3$, according to

$$\vec{i}_{\overline{T}} = \frac{\vec{r}|_{\vec{r}=\vec{x}}}{\|\vec{r}|_{\vec{r}=\vec{x}}\|}, \quad \vec{j}_{\overline{T}} = \frac{\frac{d\vec{r}}{d\Phi}|_{\vec{r}=\vec{x}}}{\|\frac{d\vec{r}}{d\Phi}|_{\vec{r}=\vec{x}}\|}, \quad \vec{k}_{\overline{T}} = \frac{\frac{d\vec{r}}{d\Theta}|_{\vec{r}=\vec{x}}}{\|\frac{d\vec{r}}{d\Theta}|_{\vec{r}=\vec{x}}\|} \tag{1}$$

where $\vec{x} \in \mathbb{R}^3$ is the vector pointing from G to B which describes the location of the system.

These coordinate systems are depicted graphically in figure 4 which also shows a spherical coordinate system defined by an azimuth angle, Φ and elevation angle Θ.

The control design, presented later in section IV relies heavily on two geometric quantities which are derived from the coordinate systems. These are the *velocity angle*, $\gamma \in \mathbb{R}$, and *tangent roll angle*, $\xi \in \mathbb{R}$. Both of these quantities are calculated using the *tangent plane*, which is defined to be the plane spanned by $\vec{j}_{\overline{T}}$ and $\vec{k}_{\overline{T}}$ with $\vec{j}_{\overline{T}}$ and $\vec{k}_{\overline{T}}$ as unit vectors.

Intuitively, the velocity angle describes the direction that a system is moving, on the sphere centered at G, with radius $\|\vec{x}\|$. Mathematically, it is the angle between the projection of a vector onto the tangent plane, and $\vec{k}_{\overline{T}}$ (or local north). This is consistent with the velocity angle used in other works such as [13]. Therefore, the velocity angle, γ, corresponding to a three-dimensional velocity vector, $\vec{v} \in \mathbb{R}^3$ is given by

$$\gamma(\vec{v}) = \text{atan}\left(\frac{\vec{v} \cdot \vec{k}_{\overline{T}}(\vec{x})}{\vec{v} \cdot \vec{j}_{\overline{T}}(\vec{x})}\right). \tag{2}$$

Note $\gamma(\vec{v}(t), \vec{x}(t))$ has been written as $\gamma(\vec{v})$, suppressing the dependence on both time, t and position $\vec{x}(t)$ for notational clarity.

The tangent roll angle, ξ, is the angle that determines the component of hydrodynamic lift that contributes to turning in the tangent plane. Mathematically, ξ is calculated to be the angle between the body unit vector, $\vec{j}_{\overline{B}}$ and the tangent plane. Mathematically, this is calculated as

$$\tan\left(\xi(\vec{j}_{\overline{B}}(t))\right) = \frac{\vec{j}_{\overline{B}} \cdot \vec{i}_{\overline{T}}(\vec{x})}{\sqrt{\left(\vec{j}_{\overline{B}} \cdot \vec{j}_{\overline{T}}(\vec{x})\right)^2 + \left(\vec{j}_{\overline{B}} \cdot \vec{k}_{\overline{T}}(\vec{x})\right)^2}}. \tag{3}$$

Note that once again, the dependence on $\vec{x}$ has been omitted for clarity.

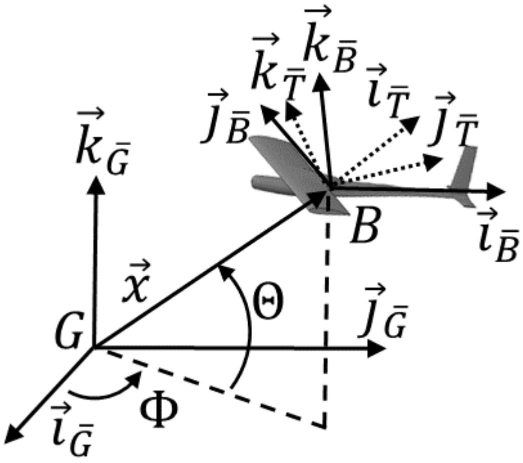


Figure 4: A diagram of the coordinate systems used in the plant and flight controller. The unit vectors of the $\overline{T}$ frame are drawn with dashed lines to differentiate them from the $\overline{B}$ frame unit vectors. The azimuth angle, Φ, and elevation angle, Θ, are also shown.

B. 6 DoF Lifting Body MHK Model

The state variables describing the position and orientation (and rates of change of the position and orientation) of coordinate system $\overline{B}$ relative to coordinate system $\overline{G}$ evolve according to standard nonlinear equations of motion:

$$\dot{\vec{\mu}} = f(\vec{\mu}, \vec{\omega}) \tag{4}$$

$$\mathrm{J}\dot{\vec{\omega}} = \vec{M}_{Net} - \vec{\omega} \times \mathrm{J}\vec{\omega} \tag{5}$$

$$\dot{\vec{x}} = \mathrm{R}(\vec{\mu})\vec{v} \tag{6}$$

$$\mathrm{M}\dot{\vec{v}} = \left(\vec{F}_{Net}(t) - \vec{\omega} \times \vec{v}\right) \tag{7}$$

Here, the orientation of the kite coordinate system $\overline{B}$ relative to the ground coordinate system, $\overline{G}$, is described by the vector of conventional Tait-Bryan angles, $\vec{\mu} \triangleq [\ \phi \quad \theta \quad \psi\ ]^T$, where ϕ is roll, θ is pitch, and ψ is yaw. The matrix $\mathrm{J} \in \mathbb{R}^{3\times3}$ is the inertia matrix, and $\vec{M}_{Net}$ is the sum of all applied moments expressed in the $\overline{B}$ frame. Here, the position vector, $\vec{x}$, is expressed in the $\overline{G}$ frame. The vector $\vec{v}$ is the associated velocity, expressed in the $\overline{B}$ frame. The matrix $\mathrm{R} \in \mathbb{R}^{3\times 3}$ is the rotation matrix, calculated based on $\vec{\mu}$, that describes the relative orientation of $\overline{B}$ with respect to $\overline{G}$. The variable $\mathrm{M} \in \mathbb{R}^{3\times 3}$ is the diagonal mass matrix, $\vec{F}_{Net}$ is the sum of all forces applied to the kite expressed in the $\overline{B}$ frame, $\vec{\omega} \triangleq [\ \omega_x \quad \omega_y \quad \omega_z\ ]^T$ is the angular velocity of $\overline{B}$ relative to $\overline{G}$. Finally, the function $\vec{f}(\vec{\mu}, \vec{\omega})$ is the standard nonlinear function that relates the angular velocity $\vec{\omega}$ and Euler angles $\vec{\mu}$ to the rates of change of the Euler angles. This can be found in standard reference texts such as [14].

The kite is subject to forces and moments resulting from four fluid dynamic surfaces (a port wing, starboard wing, horizontal stabilizer and vertical stabilizer), buoyancy, gravity, and the tether. These forces and moments are calculated as:

$$\begin{aligned}\vec{F}_{Net} =& \vec{F}_{Thr} + (V\rho - m)\, g\vec{k}_{\bar{G}} \\ &+ \frac{1}{2}\rho A_r \sum_{i_a=1}^{4} \|\vec{v}_{a,i_a}\|^2 \left(C_{L,i_a}\vec{u}_{L,i_a} + C_{D,i_a}\vec{u}_{D,i_a}\right)\end{aligned} \tag{8}$$

$$\vec{M}_{Net} = \frac{1}{2}\rho A_r \sum_{i_a=1}^{4} \|\vec{v}_{a,i_a}\|^2 \vec{r}_{a,i_a} \times \left(C_{L,i_a}\vec{u}_{L,i_a} + C_{D,i_a}\vec{u}_{D,i_a}\right) \tag{9}$$

where in (8), the first term is the force exerted at the center of mass by the tether on the lifting body, the second term describes the net buoyant force, and the last term describes the fluid dynamic forces. Here, V is the volume of the kite, ρ is the fluid density, m is the mass of the kite, and g is the acceleration due to gravity. Here, we assume that both the center of buoyancy and the tether attachment point are close to the center of mass, and thus do not contribute an appreciable moment in equation (9).

The index $i_a \in \{1, 2, 3, 4\}$ refers to each of the four independent fluid dynamic surfaces, port wing, starboard wing, horizontal stabilizer and vertical stabilizer. Therefore, the resulting force depends on the apparent flow at the aerodynamic center of each surface, which is calculated as:

$$\vec{v}_{a,i_a}(t) = \vec{v}_f(t, \vec{x} + \vec{r}_{a,i_a}) - \left(\vec{v} + \vec{\omega} \times \vec{r}_{a_i}\right), \tag{10}$$

where $\vec{v}_f(t, \cdot) : \left(\mathbb{R}, \mathbb{R}^3\right) \to \mathbb{R}^3$ is the spatially and temporally varying flow profile discussed in detail in Section III and $\vec{r}_{a,i_a}$ is the vector from the center of mass of the kite to the fluid dynamic center of the i_a^{th} surface. The fluid dynamic coefficients of equations (8) and (9) are obtained by modeling each fluid dynamic surface independently in the Athena Vortex Lattice (AVL) software of [15] and are parameterized as functions of the associated control surface deflections, δ_{ia}, as:

$$\begin{aligned}C_{L,i_a}(\vec{v}_{a_i}, \delta_{i_a}) &= C_{L_0,i_a}(\vec{v}_{a,i_a}) + C_{L_1,i_a}\delta_{i_a} + C_{L_2,i_a}\delta_{i_a}^2 \\ C_{D,i_a}(\vec{v}_{a_i}, \delta_{i_a}) &= C_{D_0,i_a}(\vec{v}_{a,i_a}) + C_{D_1,i_a}\delta_{i_a} + C_{D_2,i_a}\delta_{i_a}^2\end{aligned} \tag{11}$$

where the control sensitivity coefficients, C_{L_1,i_a}, C_{L_2,i_a}, C_{D_1,i_a}, and C_{D_2,i_a} are obtained from a quadratic fit of the discrete results from AVL. Note that the control surface deflections are limited to a range of ±30° to reflect actuator limitations. The spanwise lift coefficient distributions, $C_{l,i}(y)$, obtained from the software are heuristically corrected to account for nonlinear stall behavior that is not accurately accounted for in AVL.

Finally, the unit vectors describing the direction of the lift and drag forces are calculated from the apparent flow direction vector at the i_a^{th} hydrodynamic center:

$$\vec{u}_{D,i_a} = \frac{\vec{v}_{a,i_a}}{\|\vec{v}_{a,i_a}\|}, \tag{12}$$

$$\vec{u}_{L,i_a} = \begin{cases} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \dfrac{\left[\vec{u}^x_{D,i_a}\ 0\ \vec{u}^z_{D,i_a}\right]^T}{\|\left[\vec{u}^x_{D,i_a}\ 0\ \vec{u}^z_{D,i_a}\right]\|} & i_a \neq 4, \\ \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \dfrac{\left[\vec{u}^x_{D,i_a}\ \vec{u}^y_{D,i_a}\ 0\right]^T}{\|\left[\vec{u}^x_{D,i_a}\ \vec{u}^y_{D,i_a}\ 0\right]\|} & i_a = 4, \end{cases} \tag{13}$$

where the components of the drag direction vector are given by the dot product with the appropriate unit vector of the $\bar{B}$ coordinate system, $\vec{u}^{(\star)}_{D,i} = \vec{u}_{D,i_a} \cdot (\star)_{\bar{B}}$. Note that $i_a = 4$ refers to the vertical stabilizer, thus requiring the case structure of (13).

C. Tether Model

In this work, we adopt the tether model used in [16] and [17]. The tether is modeled as a series of $N_n - 1$ massless, cylindrical, non-compressive spring-damper links that connect N_n lumped nodal point masses, referred to as "nodes". Each of these links are subjected to buoyancy, gravity, and fluid drag. The forces acting on the tether are first calculated at the center of each link and then distributed to each node to model the dynamics. The net buoyant force at the center of the $i_l{}^{\text{th}}$ link, $\vec{F}^B_{i_l}$, is given by

$$\vec{F}^B_{i_l} = \frac{1}{2}\left(\rho - \rho_T\right)\pi r_T^2 \frac{l_T}{N_n - 1} g\, \vec{k}_{\overline{G}} \tag{14}$$

where $i_l \in \{1, 2, \ldots, N_n - 2, N_n - 1\}$ refers to the link being considered, ρ_T is the density of the tether, r_T is the radius of the tether cross section, l_T is the unspooled tether length, N_n is the number of nodes, and g is the acceleration of gravity. The fluid drag at the center of the i_l^{th} link, $\vec{F}^D_{i_l}$ is calculated as

$$\vec{F}^D_{i_l} = \frac{1}{2}\rho C_{D,T} A_{P,i} \|\vec{v}_{a,i_l}\| \vec{v}_{a,i_l} \tag{15}$$

where $C_{D,T}$ is the drag coefficient of the tether, A_{P,i_l} is the projected area of the i_l^{th} link, and $\vec{v}_{a,i_l}$ is the apparent flow velocity at the center of the i_l^{th} link. In calculating the projected areas, A_{P,i_l}, apparent flow velocities, $\vec{v}_{a,i_l}$, and spring-damper forces, it first useful to calculate the unit vector pointing from the i_l^{th} node to the $(i_l + 1)^{\text{th}}$ node, $\vec{u}_{T,i_l}$

$$\vec{u}_{T,i_l} = \frac{\vec{x}_{T,i_l+1} - \vec{x}_{T,i_l}}{\|\vec{x}_{T,i_l+1} - \vec{x}_{T,i_l}\|}. \tag{16}$$

The apparent flow velocities and projected areas can then be calculated as

$$\vec{v}_{a,i_l}(t) = \vec{v}_f\left(t, \frac{1}{2}\left(\vec{x}_{T,i_l+1} + \vec{x}_{T,i_l}\right)\right) - \frac{1}{2}\left(\vec{v}_{T,i_l+1} + \vec{v}_{T,i_l}\right) \tag{17}$$

$$A_{P,i_l} = 2 r_T \frac{l_T}{N_n - 1} \left\| \vec{u}_{T,i_l} \times \frac{\vec{v}_{a,i_l}}{\|\vec{v}_{a,i_l}\|} \right\| \tag{18}$$

where $\vec{x}_{T,i_l}$, $\vec{v}_{T,i_l}$, $\vec{x}_{T,i_l+1}$, and $\vec{v}_{T,i_l+1}$ are the positions and velocities of the i_l^{th} and $(i_l+1)^{\text{th}}$ nodes respectively.

Finally, the nonlinear spring-damper force in the i_l^{th} link is given by

$$\vec{F}_{i_l}^T = \begin{cases} \vec{0} & \|\vec{x}_{T,i_l+1} - \vec{x}_{T,i_l}\| < \frac{l_T}{N_n - 1}, \\ \vec{F}_{i_l}^T & \text{otherwise}, \end{cases} \tag{19}$$

where

$$\begin{aligned} \vec{F}_{i_l}^T = & \Bigg(- E_y \frac{\pi r_T^2}{l_T} (N_n - 1) \left(\|\vec{x}_{T,i_l+1} - \vec{x}_{T,i_l}\| - \frac{l_T}{N_n - 1} \right) \\ & - 2\zeta \sqrt{E_y \frac{\pi r_T^2}{l_T} m} \left(\vec{v}_{T,i_l+1} - \vec{v}_{T,i} \right) \cdot \vec{u}_{T,i} \Bigg) \vec{u}_{T,i_l}, \end{aligned} \tag{20}$$

where E_y is the Youngs modulus of the tether, and m is a user-defined mass which, combined with the damping ratio ζ define the size of the damping force. Note that the node attached to the ground is assumed to be fixed, and thus does not have any dynamics associated with it, and the dynamics of the node attached to the kite are dictated by kite. Thus, the positions and velocities of the $N_n - 2$ intermediate nodes, indexed by $i_n \in \{2, 3, \ldots, N_n-2, N_n-1\}$ where i_n = 2 refers to the node adjacent to the ground node and i_n =$N_n - 1$ refers to the node adjacent to the kite, are given by:

$$\dot{\vec{x}}_{T,i_n} = \vec{v}_{T,i_n} \tag{21}$$

$$\dot{\vec{v}}_{T,i_n} = \frac{N_n}{\pi r_T^2 l_T \rho_T} \left(\vec{F}_{i_n+1}^{Net} - \vec{F}_{i_n}^{Net} \right) \tag{22}$$

where $\vec{F}_{i_n}^{Net} = \vec{F}_{i_n}^{B} + \vec{F}_{i_n}^{D} + \vec{F}_{i_n}^{T}$.

D. Winch Model

The total length of unspooled tether, l_T is calculated according to a filtered saturation-plus-integrator lumped parameter model. First, the commanded tether speed $u_T(t)$ is saturated to be within the range from u_{min} to u_{max}. The result is then passed through a first order filter with time constant τ_w to obtain the achieved tether release speed, $\tilde{u}_T$.

The instantaneous estimate of power production, $P(t)$ is calculated from the achieved tether speed, $\tilde{u}_T(t)$ and the net force on the ground node, as

$$P(t) = \begin{cases} \eta_m \|\vec{F}_1^{Net}\| \tilde{u}_T(t), & \tilde{u}_T(t) \leq 0, \\ \frac{1}{\eta_{gen}} \|\vec{F}_1^{Net}\| \tilde{u}_T(t), & \text{otherwise} \end{cases} \tag{23}$$

where η_m is a lumped motor efficiency and η_{gen} is a lumped generator efficiency.

III. Realistic Ocean Current Model

The flow field, which is characterized as a function of depth ($\vec{z_G}$), down-current location, ($\vec{x_G}$), cross-current location ($\vec{y_G}$), and time (t), is computed as the superposition of a low-frequency flow profile and high-frequency turbulence model as follows:

$$\tilde{V}_{comb}(\vec{x_G}, \vec{y_G}, \vec{z_G}, t) = \tilde{V}_{turb}(\vec{x_G}, \vec{y_G}, \vec{z_G}, t) + \tilde{V}(\vec{z_G}, t), \tag{24}$$

Table I: Important plant model parameters

Variable	Description	Value	Units
Kite			
M	Mass	945	kg
–	Density	1000	kg/m^3
J_{xx}	Inertia tensor xx element	6303	kgm^2
J_{yy}	Inertia tensor yy element	2080	kgm^2
J_{zz}	Inertia tensor zz element	8320	kgm^2
–	Fuselage length	8	m
A_r	Reference area	10	m^2
–	Port wing chord	1	m
–	Port wing span	5	m
–	Starboard wing chord	1	m
–	Starboard wing span	5	m
–	Horizontal stabilizer chord	0.5	m
–	Horizontal stabilizer span	4.0	m
–	Vertical stabilizer chord	0.6	m
–	Vertical stabilizer span	2.0	m
Tether			
N_n	Number of nodes	5	–
r_T	Radius	7.2	mm
E_y	Youngs modulus	50	GPa
ζ	Damping ratio	0.75	–
m	Damping mass	945	kg
$C_{D,T}$	Drag coefficient	0.5	–
ρ_T	Density	1300	kg/m^3
Winch			
τ_w	Time constant	1	s
η_{gen}	Generator efficiency	1.0	–
η_m	Motor efficiency	0.95	–

where $\tilde{V}(\vec{z_G}, t)$, $\tilde{V}_{turb}(\vec{x_G}, \vec{y_G}, \vec{z_G}, t)$, and $\tilde{V}_{comb}(\vec{x_G}, \vec{y_G}, \vec{z_G}, t)$ represent the low-frequency flow field, high-frequency turbulent field, and combined flow model, respectively. Because the total cross-current motion of the kite spans a tiny fraction of the total current resource (e.g., the Gulf Stream), spatial variations in $\vec{y_G}$ and $\vec{x_G}$ are neglected in the low-frequency model.

The low frequency model used in this work is the Mid-Atlantic Bight South-Atlantic Bight Regional Ocean Model (MSR). This model, which was developed by North Carolina State University's Ocean Observing and Modeling Group, provides current profiles at 42 different locations in the Gulf Stream at 25 m vertical resolution. The MSR model is detailed in [18].

The turbulent high-frequency components of the ocean currents are calculated based on a discretization of the flow velocity's power spectral density (PSD) equation. Specifically, the model leverages fundamental techniques described in [19] to generate a spatiotemporally varying turbulence profile that can be applied to the hydrodynamic center of each component in the dynamic model. Based on inputs of turbulence intensity, time-averaged flow velocity profile, a specified frequency range, standard deviations, and spatial correlation coefficients for the flow velocities, the model outputs a spatial grid of time-varying velocity vectors.

IV. Control Details

The proposed control strategy is comprised of three main components,

- the *path-following flight controller* which seeks to robustly track the prescribed path,
- the *power take-off controller* which is responsible for extracting power from the system by modulating tether tension and release rate, u_T, and
- the *ILC based path optimization controller* which operates between laps, or iterations, of the system and seeks to optimize the path geometry.

These three controllers are shown in figure 5 along with the plant and flow profile model.

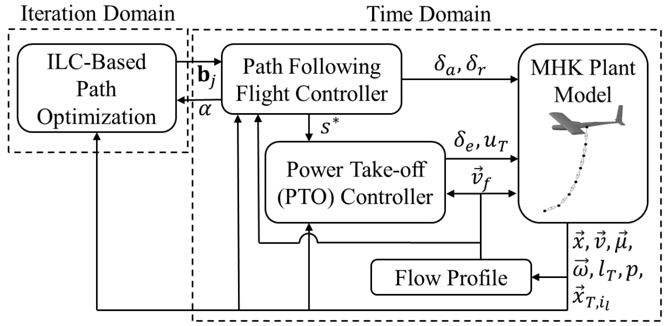


Figure 5: Block diagram showing the three main components of the control structure along with the MHK kite plant and flow profile models.

A. Path Shape definition

Before addressing how to follow or optimize a path, we consider how to *define* a path. In order to define a path that we wish to track, we require two elements, a path function that describes a general figure-8 shape in three dimensions, as a function of a parametric variable, s, and a set of numbers, $\mathbf{b} \in \mathbb{R}^{n_b}$, which describe one specific instance of the general form of the equation. In general, the path function, $\vec{\Gamma}$, can be any function where the path shape is closed. That is, it starts and ends at the same point. In this work, we select the lemniscate of Booth, which is consistent with [20], and thus $n_b = 4$. In spherical coordinates, the path is then described at any radius by the equations for azimuth and elevation angles, Φ_Γ and Θ_Γ,

$$\Phi_\Gamma(s, \mathbf{b}) = \frac{(\frac{b_1}{b_2})^2 \sin(s) \cos(s)}{1 + (\frac{b_1}{b_2})^2 (\cos(s))^2} + b_3 \tag{25}$$

$$\Theta_\Gamma(s, \mathbf{b}) = \frac{b_1 \sin(s)}{1 + (\frac{b_1}{b_2})^2 (\cos(s))^2} + b_4. \tag{26}$$

Here, a single instance of the path is specified by the vector $\mathbf{b}$, elements of which are known as *basis parameters*, $\mathbf{b} \triangleq [\ b_1\ b_2\ b_3\ b_4\ ]$. The parameters b_1 and b_2 define the overall width and height of the desired path shape, while b_3 and b_4 describe the mean path azimuth (Φ) and (Θ) respectively. For simplicity, in this work, we only seek to optimize b_1 and b_2, however, extending the algorithm to include all parameters is straightforward. The shape of the current path in Cartesian coordinates at the radial distance r, $\vec{\Gamma}(s, \mathbf{b}, r)$, is then given by the standard transformation from spherical to Cartesian coordinates.

B. Lower-Level Path Following Flight Controller

The flight control strategy is responsible for ensuring that the kite robustly tracks a prescribed figure-8 cross-current flight path. The path tracking controller shown initially in figure 5 is provided in more detail here in figure 6. This controller contains four levels, each of which accept feedback from the plant in calculating their outputs. This modular, hierarchical control structure is based on work in [20] and is partitioned into the following blocks:

1) A *path-following* block that accepts the path geometry as defined by the basis parameters $\mathbf{b}$, and outputs a desired velocity angle, γ_{des}, as defined in equation (2).
2) A *tangent roll angle selection* block, which accepts a desired velocity angle and outputs a desired tangent roll angle, which is the angle that dictates the component of hydrodynamic lift that contributes to turning in order to follow the path.
3) A *desired moment selection* block which accepts the tangent roll setpoint and side slip angle setpoint, and outputs a desired moment vector represented in the body frame $\overline{B}$.
4) A *control allocation module*, which accepts the desired moment vector and computes the required control surface deflections to be actuated by the ailerons, elevators, and rudder of the kite.

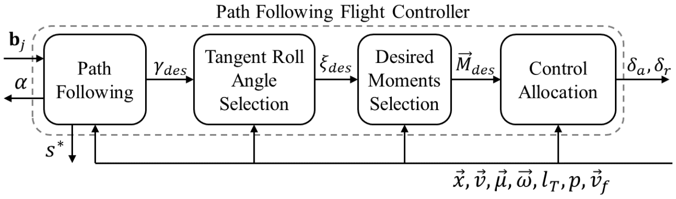

Path Following Flight Controller
$\mathbf{b}_j$
α
Path Following
γ_{des}
Tangent Roll Angle Selection
ξ_{des}
Desired Moments Selection
$\vec{M}_{des}$
Control Allocation
δ_a, δ_r
s^*
$\vec{x}, \vec{v}, \vec{\mu}, \vec{\omega}, l_T, p, \vec{v}_f$

Figure 6: Detailed block diagram showing the heirarchical structure of the path-following flight controller shown in figure 5.

1) Path-Following: Given a prescribed path, defined at the current radius, $r = \|\vec{x}\|$, and the set of basis parameters prescribed from the upper level ILC based path optimization, $\mathbf{b}_j$, this controller first calculates the path variable s^* corresponding to the closest point on the path. Specifically, it numerically solves the minimization problem

$$s^* = \arg\min_s \ \alpha(s), \quad \text{where}$$
$$\tan(\alpha(s)) = \frac{\|\vec{x} \times \vec{\Gamma}(s, \mathbf{b}_j, r)\|}{|\vec{x} \cdot \vec{\Gamma}(s, \mathbf{b}_j, r)|}. \tag{27}$$

Here, $\alpha(s)$ is the angle between the position vector, $\vec{x}$ and the path at the current radius, $\vec{\Gamma}(s, \mathbf{b}_j, r)$. In this work, this minimization problem is solved at every time step using the Golden Section minimization technique. The Golden Section search method was chosen because it requires only zeroth-order information, but other numerical minimization techniques (for example: Newton's method) can be used as well.

Given the closest path parameter, s^*, we then calculate a three dimensional vector representing the desired velocity of the system. This desired velocity vector is taken to be a weighted average between the *perpendicular vector*, $\vec{p}^*_\perp$, and the *parallel vector*, $\vec{p}^*_\parallel$. Intuitively, the parallel vector is the vector that perfectly aligns with the path, and the perpendicular vector is the vector from the current position, to the closest point on the path, which is by definition, perpendicular to the path. Mathematically, the perpendicular vector is given by

$$\vec{p}^*_\perp = \frac{\hat{p}_\perp}{\|\hat{p}_\perp\|} \text{ where } \hat{p}_\perp = \begin{bmatrix} (\vec{\Gamma}(s^*, \mathbf{b}_j, r) - \vec{x}) \cdot \vec{j}_{\overline{T}}(\vec{x}) \\ (\vec{\Gamma}(s^*, \mathbf{b}_j, r) - \vec{x}) \cdot \vec{k}_{\overline{T}}(\vec{x}) \\ 0 \end{bmatrix} \tag{28}$$

where $\vec{j}_{\overline{T}}(\vec{x})$ and $\vec{k}_{\overline{T}}(\vec{x})$ are defined in (1) and shown in figure 4. The parallel vector, $\vec{p}^*_\parallel$, is a unit vector that lies parallel to the path at the path variable corresponding to the closest point on the path, s^* and is calculated by

$$\vec{p}^*_\parallel = \frac{\hat{p}_\parallel}{\|\hat{p}_\parallel\|} \quad \text{where} \quad \hat{p}_\parallel = \left.\frac{d\vec{\Gamma}}{ds}\right|_{s=s^*}. \tag{29}$$

The desired velocity unit vector, $\vec{v}_{des}$ is then calculated as the linearly weighted sum of the perpendicular and parallel vectors according to

$$\begin{aligned} \overline{\alpha}(s^*) &= \min\left\{\alpha(s^*), \alpha_0\right\} \\ \vec{v}_{des} &= \left(1 - \frac{\overline{\alpha}(s^*)}{\alpha_0}\right)\vec{p}^*_\parallel + \frac{\overline{\alpha}(s^*)}{\alpha_0}\vec{p}^*_\perp. \end{aligned} \tag{30}$$

Here, α_0 serves as an upper limit on the possible angle used in the weighting. Intuitively, this means that if the angle between the system and the path is more than α_0, then the weighting will be entirely on the second term, making the system head directly towards the closest point on the path.

The desired velocity angle, which is the output of the leftmost block in figure 6, is then given by $\gamma(\vec{v}_{des})$ where $\gamma(\cdot)$ is the velocity angle calculated according to equation (2).

2) Tangent Roll Angle Selection: The next stage of the flight controller maps this desired velocity angle to a desired tangent roll angle, ξ_{des}. Recall that the tangent roll angle, ξ, describes the orientation of the system relative to the tangent plane and determines the amount of fluid dynamic force that contributes to turning on a sphere of radius $\|\vec{x}\|$ centered at G.

The desired tangent roll angle is calculated using saturated proportional control based on the error in the velocity angle, specifically:

$$\xi_{des} = \min\{\max\{k_\gamma\left(\gamma(\vec{v}) - \gamma(\vec{v}_{des})\right), \xi_{min}\}, \xi_{max}\}, \tag{31}$$

where k_γ is the proportional gain. We then calculate an error signal, $e_\xi(t) = \xi(\vec{j}_{\overline{B}}(t)) - \xi_{des}$, where the current tangent roll angle, $\xi(\vec{j}_{\overline{B}}(t))$, is calculated using equation (3).

3) Desired Moment Vector Selection: In selecting the desired moments, we utilize the rolling moment to control tangent roll angle, ξ, and yawing moment to drive aerodynamic side slip angle, β, to zero. The PTO controller articulates the elevator to passively trim the system to a high angle of attack, resulting in large tether tension, during spool out and a low angle of attack, resulting in low tether tension, during spool in. Therefore, it is desirable that the deflection of the ailerons and rudder contribute negligible or zero pitching moment. This helps ensure that the PTO controller will be solely responsible for the pitch behavior. Ultimately, the desired moment vector is given by two PID controllers,

$$\vec{M}_{des} = \begin{bmatrix} k_{p_L} e_\xi(t) + k_{i_L}\int_0^t e_\xi(\tau)d\tau + k_{d_L}\dot{e}_\xi(t) \\ 0 \\ k_{p_N}\beta(t) + k_{i_N}\int_0^t \beta(\tau)d\tau + k_{d_N}\dot{\beta}(t) \end{bmatrix} \tag{32}$$

where $\beta(t)$ is the fluid dynamic side slip angle.

4) Control Allocation Module: In order to map the desired moment vector to control surface deflections, we invert a linearized approximation of the nonlinear mapping from deflections to hydrodynamic moments. This approximation is calculated by neglecting the effect of angular velocity on the apparent flow at each fluid dynamic surface, then linearizing to obtain an expression in the following form:

$$\vec{M}_{net} = \vec{M}_o + A\begin{bmatrix}\delta_a \\ \delta_e \\ \delta_r\end{bmatrix}, \tag{33}$$

where δ_a, δ_e, and δ_r represent the deflection angles of the ailerons, elevator, and rudder, respectively. The variable $\vec{M}_o$ is given by:

$$\vec{M}_o = \frac{1}{2}\rho A_r \|\vec{v}_a\|^2 \sum_{i_a=1}^{4} \vec{r}_{a,i_a} \times (C_{L_o,i_a}\vec{u}_{L,i_a} + C_{D_o,i_a}\vec{u}_{D,i_a}), \tag{34}$$

and the matrix A is formed by re-arranging the cross products and deflection angles in (9) and (11) into a matrix where the results of the cross products form the columns of the matrix:

$$\begin{aligned} A &= \frac{1}{2}\rho A_r \|\vec{v}_a\|^2 \left[\vec{a}_1 - \vec{a}_2, \vec{a}_3, \vec{a}_4\right] \quad \text{where} \\ \vec{a}_i &= \vec{r}_{a,i_a} \times (C_{L_1,i_a}\vec{u}_{L,i_a} + C_{D_1,i_a}\vec{u}_{D,i_a}). \end{aligned} \tag{35}$$

The resulting control surface deflections, δ_a and δ_r, are computed by solving equation (33) for $\vec{M}_{net} = \vec{M}_{des}$ where $\vec{M}_{des}$ is given by equation (32). Note that the result for δ_e obtained from this method is not used. Instead the value of δ_e obtained from the PTO controller is passed to the plant.

C. Power Take-off (Winch) Controller

The iterative learning power take-off controller used in this work seeks to satisfy the net-zero spooling constraint by approximating the behavior over the next lap using feedback from the previous lap. The commanded rate of tether release, $u_T(t)$, is set by a spooling controller that seeks to spool tether out at a high angle of attack during the portions of the lap in which large tensions are possible, then spool tether in at a low angle of attack during the remainder of the lap. The intra-cycle spooling algorithm in this work is designed to maintain a consistent tether length each lap, represented by the constraint:

$$\int_{t_{0,j}}^{t_{f,j}} \tilde{u}_j(\tau)d\tau = 0. \tag{36}$$

where the index j refers to the lap or iteration number. This ensures that the kite remains in a consistent depth within the ocean velocity profile. In attempting to find the command sequence that satisfies this constraint, we make several key simplifying assumptions:

- The winch is capable of achieving the commanded speed.
- The winch is capable of achieving that speed quickly, relative to the rate of change of the command.
- The commanded spooling speed is piecewise constant over each of N_R "spooling regions", and alternates between spooling in and spooling out at the maximum speed, u_{spool}.

The first two approximations should hold for a well designed winch/generator system, meaning that $\tilde{u}_T(t) \approx u_T(t)$. The three approximations together mean that the constraint equation of (36) can be written as:

$$0 = \mathbf{1}^{1\times N_R}\underline{U}^{j-1}\Delta_T^j, \tag{37}$$

where the matrix $\underline{U}^{j-1} \in \mathbb{R}^{N_R\times N_R}$ is a diagonal matrix where the element in the p^{th} and q^{th} column is given by:

$$\underline{U}^{j-1}_{p,q} = \begin{cases} u^{j-1}_{spool} & p = q = \text{odd} \\ -\,u^{j-1}_{spool} p = q = \text{even} \\ 0 & p \neq q. \end{cases} \tag{38}$$

As derived in [3], u^{j-1}_{spool} is taken to be one third of the mean flow speed at the vehicle center of mass (point B) over the last lap of the system. The vector $\Delta_T^j \in \mathbb{R}^{N_R}$ is a vector containing the time durations required to traverse each section of the path during next (j^{th}) lap. Because the timings of the next lap are not known beforehand, it is desirable to define our tether spooling controller in terms of the path variable, s, *not* time. Therefore, we transform the time-domain constraint of (37) to a path-domain constraint by using a numerical approximation of the time derivative of the path variable from the previous lap in each spooling region. Here, we denote the spooling region with the index $i_R = 1, 2, \ldots, N_R$. Specifically, we approximate the i_R^{th} element of Δ_T^j, written as Δ^j_{T,i_R} in terms of the path variable using logged data from the previous lap, $j-1$. This is where the feedback from the last lap enters in to the calculation of the control input (spooling regions) for the next lap. Specifically,

$$\Delta^j_{T,i_R} \approx \frac{s^{j-1}_{i_R+1} - s^{j-1}_{i_R}}{\delta s^{j-1}_{i_R}}. \tag{39}$$

Note that $s^{j-1}_{i_R}$ refers to the value of the path variable at the end of the i_R^{th} region during the previous lap, $j-1$. Additionally, $\delta s^{j-1}_{i_r}$ is the mean of the time derivative of $s(t)$ over the i_R^{th} section of the path. Therefore

$$\Delta_T^j = \begin{bmatrix} \frac{1}{\delta s_1^{j-1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\delta s_2^{j-1}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \frac{1}{\delta s^{j-1}_{N_R+1}} \end{bmatrix} \underline{D} \begin{bmatrix} s_1^{j-1} \\ s_2^{j-1} \\ \vdots \\ s^{j-1}_{N_R+1} \end{bmatrix}, \tag{40}$$

$$\triangleq \underline{\delta s}^{j-1}\underline{D}S^{j-1}. \tag{41}$$

Furthermore, because the path is defined using a path variable $s \in \{0, 1\}$, $s^j_{N_R+1} = 1$ for all j. The discrete difference operation matrix $\underline{D} \in \mathbb{R}^{N_R\times N_R}$ is a matrix with ones along the main diagonal and negative ones on the diagonal underneath the main diagonal. Thus, after every lap, the problem of satisfying our approximation of the net-zero spooling constraint becomes one of solving an approximated version of the constraint equation,

$$0 = \mathbf{1}^{1\times N_R}\underline{U}^{j-1}\underline{\delta s}^{j-1}\underline{D}S^j, \tag{42}$$

for the vector $S^j \in \mathbb{R}^{N_R+1}$, the elements of which define the spooling regions for the next lap. Note that in general, this is a single scalar equation and cannot be solved uniquely for the elements of S^j. However, if we prescribe a structure to the spooling regions, we can reduce the number of parameters that define the spooling regions to one, resulting in a unique solution. In the case of the figure-8 path geometry, we know that the tension profile over the course of a figure-8 exhibits two local minima, which occur roughly at $s = 0.25$ and $s = 0.75$. Therefore, we parameterize our vector S^j in terms of a single scalar variable, σ_w^j, which defines the width of the spooling region. Therefore, S^j takes the form

$$S^j = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.75 \\ 0.75 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 0 \end{bmatrix} \sigma_w^j. \tag{43}$$

By substituting this expression into (42), we can solve directly for the width of the spool-in regions, σ_w^j. This then defines a simple, switched spooling control structure:

$$\mathbf{u}_T^j(s^*(t)) = \begin{cases} u_{in} & 0.25 - \sigma_w^j \leq s^*(t) \leq 0.25 + \sigma_w^j \text{ or} \\ & 0.75 - \sigma_w^j \leq s^*(t) \leq 0.75 + \sigma_w^j, \\ u_{out} & \text{otherwise.} \end{cases} \tag{44}$$

While (44) will yield zero net spooling under *nominal* conditions, it is not robust to disturbances that cause the actual flight speed (and therefore the time required to traverse a particular section of the figure-8) to differ from that which was used in computing $\mathbf{u}_T^j(s^*(t))$. To add robustness to the spooling strategy, we utilize a simple feedback controller to track a target tether length, $l^j_{T,SP}(s(t))$, which is obtained by integrating (44) over the path as follows:

$$l^j_{T,SP}(s(t)) = l_{T,0} + \int_0^{s(t)} \frac{\mathbf{u}_T^j(\sigma)}{\delta s^{j-1}} d\sigma \tag{45}$$

where $l_{T,0}$ is the target tether length selected by the user.

It should also be noted that if the time spent switching between the minimum and maximum spool speeds is small compared to the time spent spooling in or spooling out, then this iterative learning winch controller still is able to meet the zero net tether spooled per lap constraint. On the other hand, if the system cannot physically meet the spooling commands, the outer level ILC (which treats the system as a black box) will compensate for this and penalize the set of basis parameters resulting in a path geometry that results in poor performance.

D. ILC-Based Path Optimization Controller

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. By parameterizing the path in terms of a small set of parameters, this work ensures that the optimization problem is computationally feasible for real-time implementation.

The path optimization problem is then solved by applying the following iterative adaptation law between each iteration:

$$\mathbf{b}_{j+1} = \mathbf{b}_j + K_L \nabla \tilde{J}(\mathbf{b}_j). \tag{46}$$

In this update law, $\mathbf{b}_j \in \mathbb{R}^n$ represents the basis parameters at iteration j, the matrix $K_L \in \mathbb{R}^{n \times n}$ represents the learning gain and $\nabla \tilde{J}(\mathbf{b}_j)$ is the gradient of the *estimated* response surface, $\tilde{J}$, at the current basis parameters, $\mathbf{b}_j$, of iteration j.

The structure of the update law in (46), parallels the structure of gradient-based ILC [21], with $\mathbf{b}_j$ taking the place of the control input sequence and $\nabla \tilde{J}(\mathbf{b}_j)$ taking the place of the gradient of a (squared) tracking error cost function with respect to the control input sequence. The first two terms of the adaptation law reflect movement in the direction of perceived increasing performance index.

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 it becomes computationally feasible to solve in real-time. Additionally, we implement a trust region in the basis parameters to eliminate large changes in path shape that could potentially destabilize the system.

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

1) The estimated response surface, $\tilde{J}(\mathbf{b}_j)$, must be updated based on performance over the previous iteration.
2) The estimated optimal basis parameters, $\tilde{\mathbf{b}}_j^*$, or the estimated gradient of the response surface, $\nabla \tilde{J}(\mathbf{b}_j)$, 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 equation (46).

These operations are shown as blocks in figure 7, which is in turn a detailed depiction of the block labelled “ILC-Based Path Optimization” from figure 5. We now consider how to quickly and efficiently perform the response surface estimate update and ILC-based basis parameter update steps. To estimate $\tilde{J}(\mathbf{b}_j)$, 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, $\mathbf{c} \in \mathbb{R}^q$, as follows:

$$\tilde{J}(\mathbf{b}_j) = \mathbf{h}(\mathbf{b}_j)^T \mathbf{c}_j. \tag{47}$$

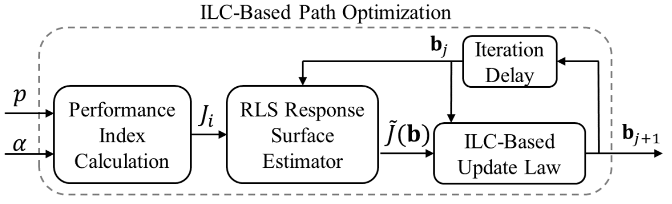


Figure 7: Detailed block diagram of the ILC-based path optimization shown in figure 5.

Here, $\tilde{J}(\mathbf{b}_j)$ 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 the basis parameters). The coefficients to the estimated response surface, $\mathbf{c}$ are then identified at each iteration, j, using recursive least squares (RLS), with an exponential forgetting factor, λ, as follows:

$$\begin{aligned} V_j &= \frac{1}{\lambda}\left(V_{j-1} - \frac{V_{j-1}\mathbf{h}(\mathbf{b}_j)\mathbf{h}(\mathbf{b}_j)^T V_{j-1}}{1 + \mathbf{h}(\mathbf{b}_j)^T V_{j-1}\mathbf{h}(\mathbf{b}_j)}\right), \quad \lambda \le 1 \\ \mathbf{c}_j &= \mathbf{c}_{j-1} + V_j \mathbf{h}(\mathbf{b}_j)\big(J(\mathbf{b}_j) - \mathbf{h}(\mathbf{b}_j)^T \mathbf{c}_{j-1}\big) \end{aligned} \tag{48}$$

where V_j is the inverse of the weighted sample covariance matrix.

Given a parameter estimate, $\mathbf{c}_j$, the corresponding gradient of $\mathbf{h}(\mathbf{b}_j)^T \mathbf{c}_j$, denoted by $\nabla \tilde{J}(\mathbf{b}_j)$, 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, $\mathbf{c}_j$ thus ensuring that the estimated response surface is weighted towards data acquired using similar course geometries *and* under similar flow conditions.

E. MHK-Specific Implementation

Although the algorithm described in this work is generic, the performance index for this application is chosen to achieve two simultaneous goals. The first term of this performance index characterizes the mean power generation over a single iteration, and the second term serves as a penalty on undesirable paths, thus incorporating secondary design objectives. Specifically, the performance index at each iteration is calculated as

$$J_j = \frac{1}{t_{e,j} - t_{s,j}} \int_{t_{s,j}}^{t_{e,j}} P(\tau) - k_w d(\tau) d\tau, \tag{49}$$

where $t_{s,j}$ and $t_{e,j}$ are the start and end times of the j-th iteration, $P(t)$ is the estimated power produced by (23), k_w is a scalar weight, and $d(t)$ is the penalty that describes our secondary design objectives. In this work, we use $d(t) = \alpha(t)$ where $\alpha(t)$ is the interior angle to the path calculated at every instant via the second equation in (27).

Table II summarizes critical parameters used in all three controllers. In this work, we used a diagonal learning gain matrix with $K_L = k_L I^{2\times 2}$ where $k_L \in \mathbb{R}$. The penalty weights were tuned to reflect the goals of the designer. There is a well established trade-off (see []) between robustness and performance when tuning penalty weights for an economic objective. The learning gains were tuned heuristically by comparing changes in the achieved performance to changes in the basis parameters.

Note that table II shows different values for the penalty weight k_w used in (49) and the learning gain k_L used in equation (46) depending on which flow speed is being examined. This is because the power production of the system varies dramatically with flow speed. Under idealized circumstances, the power produced by an MHK system is proportional to the flow speed *cubed*. Thus, the magnitude of the first term in (49) and the gradient in (46) are expected to increase almost eight-fold when the flow speed is doubled. Thus, in order to select weights and gains for simulations, the algorithm was first tuned for a constant 1 m/s flow speed and then the resulting weights and gains were multiplied or divided appropriately by $0.8v_f^3$ where v_f is the flow speed being used in simulation. Note that 0.8 was chosen heuristically and is meant to capture deviations from the ideal power scaling laws. In the case of the variable flow speed simulation $v_f = 1.5\,m/s$ was used.

V. Results

To investigate the efficacy of our control and optimization structure we simulated two test cases. In both test cases, the algorithm was implemented on a realistic, 6DOF kite model with a lumped mass tether model, performing intracycle flight, made possible by our new iterative learning spooling controller. The test cases were:

- a spatially and temporally *constant* flow environment where the flow velocity vector is independent of spatial location and time, and
- a spatially and temporally *varying* flow environment where the flow velocity vectors were calculated using an ocean flow model that combines both high and low frequency flow components as detailed in section III.

In both the constant and varying flow simulation results, we simulated a system deployed from a fixed floating platform located 200 m above the sea floor. The path under consideration had a mean azimuth angle, b_3, of $0°$ and a mean elevation angle, b_4, of $-20°$. Note that the negative elevation angle corresponds to deployment from a floating platform, as depicted on the left of figure 2. A summary of plant parameters used in simulation is given in table I.

In this work, an arbitrary baseline path was chosen for comparison. Other methods that can be used to pick a baseline path include offline optimization, which would require omniscient knowledge of the plant and future flow behavior. Even still, this would be a computationally intensive endeavor, and an interesting research topic in its own regard, but out of the scope of this work. Another possible method for creating a baseline strategy can be seen in [22], where an airborne kite's steering actuation is optimized (which dictates its path shape), in a constant wind environment, using a simplified dynamic model for control development. Adopting this method to a spatiotemporally varying turbulent flow field, constitutes a significant amount of work, and is out of the scope of this paper.

Table II: Important controller parameters

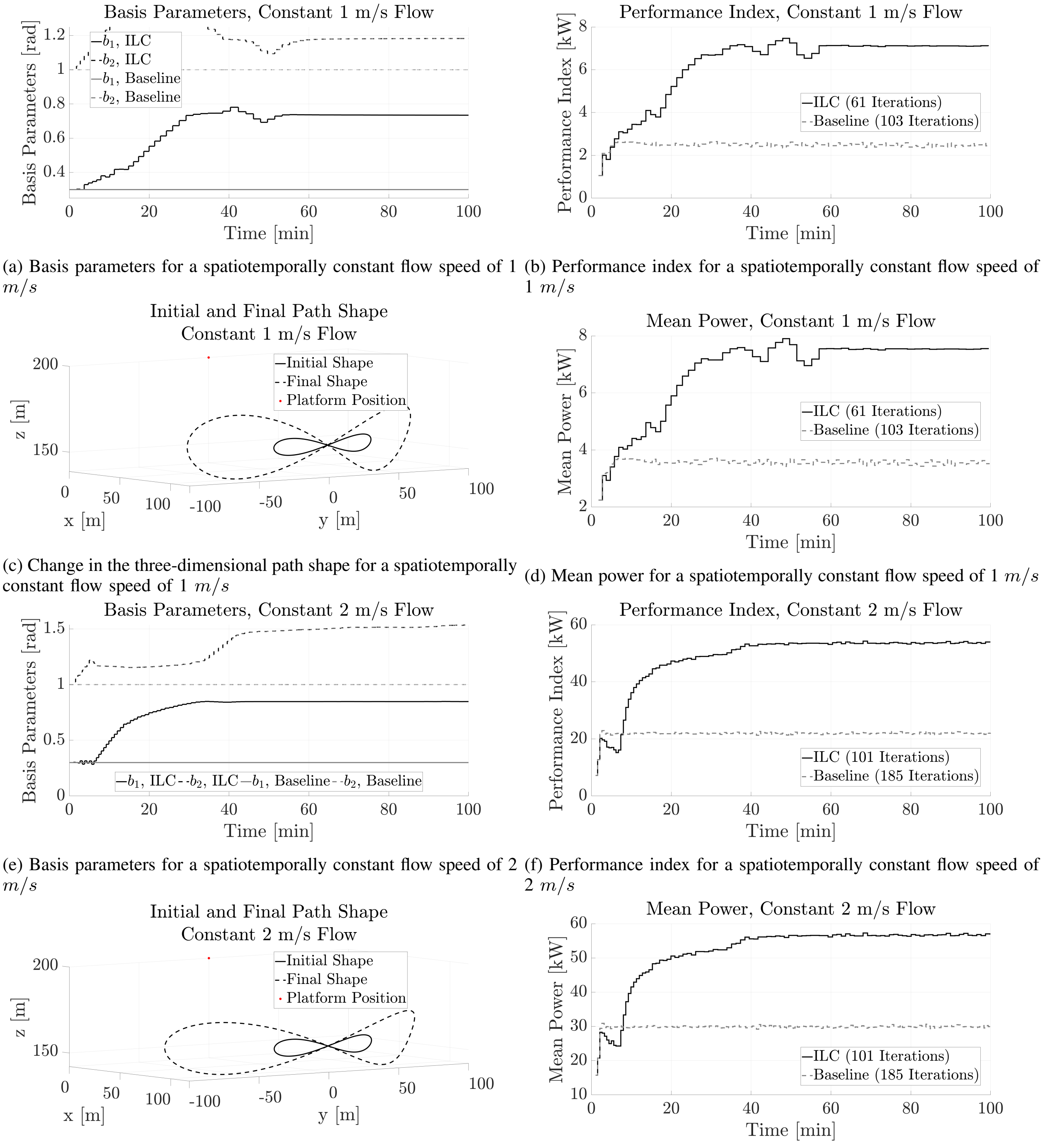
Variable	Description	Value	Units
Path-Optimization Controller			
k_L^1	Learning gain for 1 m/s flow	21×10^{-6}	rad^2/W
k_w^1	Penalty weight for 1 m/s flow	43	kW/rad
k_L^2	Learning gain for 2 m/s flow	3.3×10^{-6}	rad^2/W
k_w^2	Penalty weight for 2 m/s flow	274	kW/rad
k_L^v	Learning gain variable flow	7.8×10^{-6}	rad^2/W
k_w^v	Penalty weight variable flow	116	kW/rad
λ	Forgetting factor	0.95	$-$
Flight Controller			
α_0	Angle weighting limit	6	deg
k_γ	Tangent roll proportional gain	0.2	$-$
ξ_{min}	Min prescribed tangent roll	-20	deg
ξ_{max}	Min prescribed tangent roll	20	deg
k_{pL}	Roll moment proportional gain	6.3×10^5	Nm/rad
k_{iL}	Roll moment integral gain	0	$Nm/rad\ s$
k_{dL}	Roll moment derivative gain	6.3×10^5	Nms/rad
τ_L	Roll moment filter time constant	0.001	s
k_{pN}	Yaw moment proportional gain	5730	Nm/rad
k_{iN}	Yaw moment integral gain	0	$Nm/rad\ s$
k_{dN}	Yaw moment derivative gain	0	Nms/rad
τ_N	Yaw moment filter time constant	1	s
PTO Controller			
$-$	Spool in elevator deflection	23	deg
$-$	Spool out elevator deflection	0	deg

A. Spatiotemporally Constant Flow Profile

The results in this section were obtained for two spatiotemporally constant flow profiles, one in which $\vec{v}_{f,i}(t,\vec{r}_i) = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T \ \forall\ \vec{r}_i, t$, and one in which $\vec{v}_{f,i}(t,\vec{r}_i) = \begin{bmatrix}2 & 0 & 0\end{bmatrix}^T \forall \vec{r}_i, t$. Both flow speeds were compared to baseline simulations wherein the course geometry did not change from the initial conditions. Note that changes in performance for the baseline simulation are due to transient behavior in the iteration domain of the spooling controller. Similarly, the momentary peak in mean power in figure 8d is also due to iteration domain transients in the PTO controller.

Results are shown in figure 8. Figures 8a and 8e show how the basis parameters change over the course of the 100 minute simulation and figures 8c and 8g show the shape of the path at the beginning and end of the optimization for both simulations. It is interesting to note that under both flow speeds, b_1 converged to nearly the same value, while b_2 converged to different values. This suggests that the optimal value of b_1 may be relatively insensitive to flow speed whereas the optimal value of b_2 may vary with the flow speed. Also, because the optimized path shape is significantly larger than the initial path, the optimized path resulted in fewer iterations than the unoptimized path, as noted in the legend of figures 8b, 8d, 8f, and 8h.

Furthermore, the final converged values of power production shown in figures 8d and 8h appear to nearly follow the idealized relationship between flow speed and power production. Finally, note that the optimized path geometry produced approximately 100% to 200% more power than the unoptimized path geometry.



(a) Basis parameters for a spatiotemporally constant flow speed of 1 m/s

(b) Performance index for a spatiotemporally constant flow speed of 1 m/s

(c) Change in the three-dimensional path shape for a spatiotemporally constant flow speed of 1 m/s

(d) Mean power for a spatiotemporally constant flow speed of 1 m/s

(e) Basis parameters for a spatiotemporally constant flow speed of 2 m/s

(f) Performance index for a spatiotemporally constant flow speed of 2 m/s

(g) Change in the three-dimensional path shape for a spatiotemporally constant flow speed of 2 m/s

(h) Performance index for a spatiotemporally constant flow speed of 2 m/s

Figure 8: Summary results for a spatially and temporally constant flow profile at both 1 m/s and 2 m/s.

B. Spatiotemporally Varying Flow Profile

Simulations in the realistic flow scenarios were compared against a baseline wherein the course geometry was fixed at the initial course geometry over the entire duration of the simulation.

Figure 9a shows the evolution of the basis parameters over the course of 100 minutes of simulation and figure 9c shows the shape of the target path for both the first and last iteration. We can see from the plots that the parameters change significantly from the initial conditions resulting in a significantly larger course geometry. Similar to the constant flow results in the previous section, the optimized path geometry was larger than the initial path geometry, resulting in fewer iterations over the whole simulation.

Figures 9b and 9d show how the performance index and mean power vary over the course of the 100 minute simulation time. We can see from figure 9b that the ILC-based path optimization significantly increases the performance index. In fact, figure 9d shows that it produces just under 25 kW on average whereas the unoptimized course produces approximately 12.5 kW on average, an increase of almost 100%. Additionally, this plot demonstrates that the algorithm is not sensitive to high-frequency, spatiotemporal variations in flow speed.

VI. Conclusion

This work presented an ILC-based path optimization implemented on both a realistic, yet control oriented dynamic model of a marine hydrokinetic tethered energy system. This algorithm was also shown to be effective when implemented on a MHK system operating in both a constant flow field, and a realistic 3-D spatiotemporally varying flow field incorporating both low-frequency current information, and high-frequency turbulence. Additionally, the results demonstrated that the ILC-based path optimization was capable of augmenting the average power production by nearly 100% relative to the initial course geometry in a realistic, spatiotemporally varying turbulent flow field.

VII. Acknowledgements

This research was supported by National Science Foundation award number 1727779, titled "Collaborative Research: An Economic Iterative Learning Control Framework with Application to Airborne Wind Energy Harvesting", and US Department of Energy award number DE-EE0008635, titled "Device Design and Robust Periodic Motion Control of an Ocean Kite System for Hydrokinetic Energy Harvesting."

References

[1] K. Archer and K. Caldeira, "Atlas of high altitude wind power," *Carnegie Institute for Science*, 2008.

[2] J. Bane, R. He, M. Muglia, C. Lowcher, Y. Gong, and S. Haines, "Marine hydrokinetic energy from western boundary currents," *Annual Review of Marine Science*, 2017.

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

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

[5] Minesto website. [Online]. Available: https://minesto.com/

[6] 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.

[7] 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.

[8] 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.

[9] 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," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 2698–2704.

[10] M. K. Cobb, K. Barton, H. Fathy, and C. Vermillion, "Iterative learning-based path optimization for repetitive path planning, with application to 3-d crosswind flight of airborne wind energy systems," *IEEE Transactions on Control Systems Technology*, pp. 1–13, 2019.

[11] M. Cobb, K. Barton, H. Fathy, and C. Vermillion, "An iterative learning approach for online flight path optimization for tethered energy systems undergoing cyclic spooling motion," in *2019 American Control Conference (ACC)*, July 2019, pp. 2164–2170.

[12] J. Reed, J. Daniels, A. Siddiqui, M. Cobb, and C. Vermillion, "Optimal exploration and charging for an autonomous under water vehicle with energy-harvesting kite," in *2020 American Control Conference (ACC)*, May 2020, pp. 5494–5499, under Review.

[13] L. Fagiano, A. U. Zgraggen, M. Morari, and M. Khammash, "Automatic crosswind flight of tethered wings for airborne wind energy: Modeling, control design, and experimental results," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1433–1447, July 2014.

[14] L. Meirovitch, *Methods of Analytical Dynamics*, 1998.

[15] M. Drela and H. Youngren. (2017) AVL. [Online]. Available: http://web.mit.edu/drela/Public/web/avl/

[16] A. Siddiqui, P. Ramaprabhu, J. Deese, and C. Vermillion, "Flight dynamics and control of a farm of tethered energy systems in a turbulent field," in *Procceedings of the 2019 Dynamic Systems and Controls Conference*, 10 2019.

[17] U. Fechner, R. van der vlugt, E. Schreuder, and R. Schmehl, "Dynamic model of a pumping kite power system," *Renewable Energy*, 09 2015.

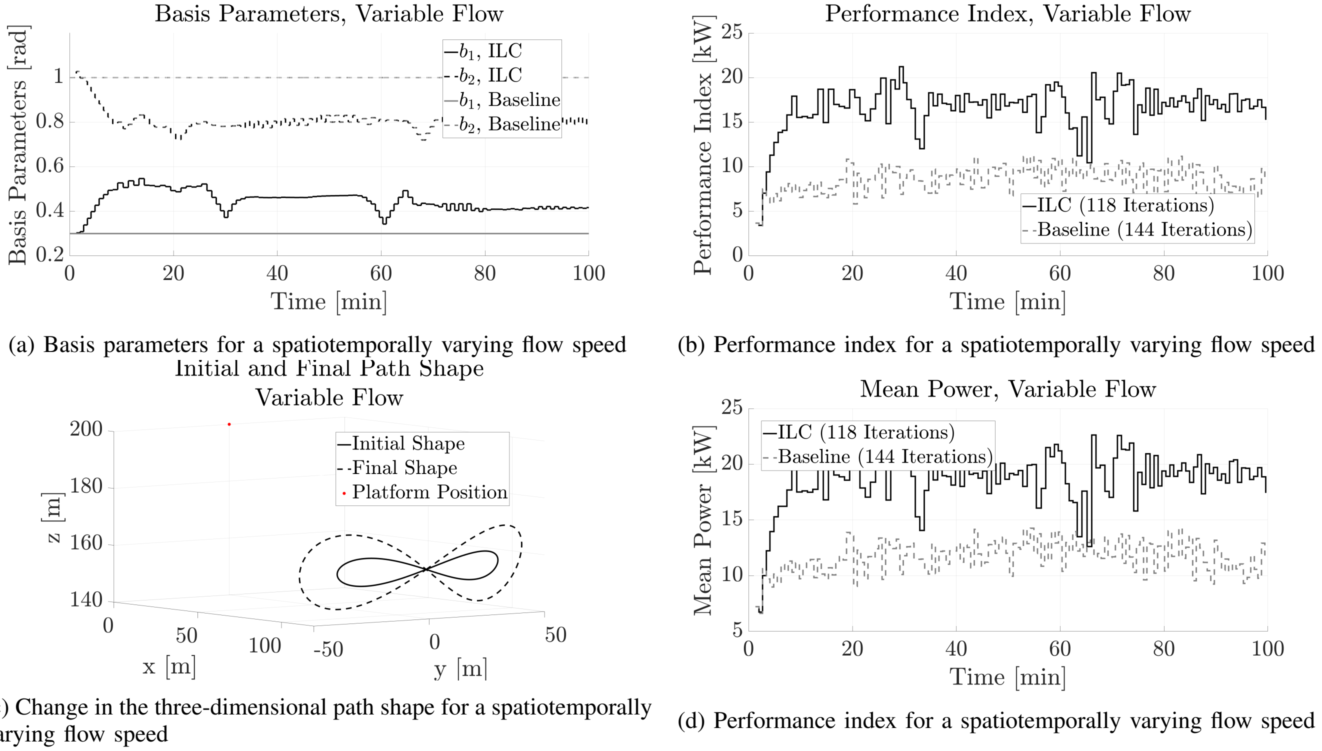
[18] Z. Yang and A. Copping, *Marine Renewable Energy: Resource Characterization and Physical Effects*. Springer, 2017.

[19] P. Pyakurel, J. H. VanZwieten, M. Dhanak, and N. I. Xiros, "Numerical modeling of turbulence and its effect on ocean current turbines," *International Journal of Marine Energy*, vol. 17, pp. 84–97, 2017.

[20] S. Rapp, R. Schmehl, E. Oland, and T. Haas, "Cascaded pumping cycle control for rigid wing airborne wind energy systems," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 11, pp. 2456–2473, 2019.

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

[22] S. Costello, G. François, and D. Bonvin, "Directional real-time optimization applied to a kite-control simulation benchmark," in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 1594–1601.



(a) Basis parameters for a spatiotemporally varying flow speed

(b) Performance index for a spatiotemporally varying flow speed

(c) Change in the three-dimensional path shape for a spatiotemporally varying flow speed

(d) Performance index for a spatiotemporally varying flow speed

Figure 9: Summary results for a spatially and temporally varying flow profile consisting of a low frequency component and high-frequency, turbulent component.