Chapter 11
Numerical Trajectory Optimization for Airborne Wind Energy Systems Described by High Fidelity Aircraft Models

Greg Horn, Sébastien Gros, Moritz Diehl

Abstract In order to study design tradeoffs in the development of an AWE system, it is useful to develop a code to optimize a trajectory for arbitrary objective function and constraints. We present a procedure for using direct collocation to optimize such a trajectory where a model is specified as a set of differential-algebraic equations. The six degree of freedom single-kite, pumping-mode AWE model developed in Chap. 10 is summarized, and two typical periodic optimal control problems are formulated and solved: maximum power and number of cycles per retraction. Finally, a procedure for optimally transitioning between two fixed trajectories is presented.

11.1 Introduction and Problem Statement

In the development of AWE systems, there arise design decisions which cannot easily be quantified with analytical methods. Numerical optimal control techniques are often used to study maximum power generation [2, 4, 8, 9, 16], and there are many other useful applications such as comparing circular and figure eight trajectories, deciding how many loops to fly before retraction in a pumping system, or studying the effect on average power of varying things like minimum altitude or power output variation. In these studies, simplified models are often used where for instance, a lift coefficient C_L and some form of "tether roll angle" are controlled directly. It is possible to obtain more accurate results by optimizing trajectories using a full six degree of freedom aircraft model, but the optimization problem becomes larger and more difficult to solve. This chapter describes one numerical approach that is well-suited to solving optimal control problems with these larger models.

In Sect. 11.1 the Optimal Control Problem (OCP) is motivated and a general form is stated for continuous time. In Sect. 11.2 the direct collocation technique for

Greg Horn (✉) · Sébastien Gros · Moritz Diehl
KU Leuven, Electrical Engineering Department, Arenberg Kasteelpark 10, 3001 Leuven, Belgium,
e-mail: ghorn@esat.kuleuven.be

U. Ahrens et al. (eds.), *Airborne Wind Energy,* Green Energy and Technology,
DOI: 10.1007/978-3-642-39965-7_11, © Springer-Verlag Berlin Heidelberg 2014

discretizing and numerically solving the OCP is described. Section 11.3 describes a homotopy procedure for automatically generating initial guesses for the numerical solver. Section 11.4 summarizes the model equations developed in Chap. 10 and then solves two periodic optimal control problems for a small scale AWE system. Section 11.5 describes a procedure for transitioning from one trajectory to another.

11.1.1 Statement of Optimal Control Problem

An AWE system can be modeled generally as a set of implicit differential-algebraic equations (DAE):

$$0 = \mathbf{f}(\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \theta, t) \tag{11.1}$$

with differential states $\mathbf{x}$, algebraic variables $\mathbf{z}$, control inputs $\mathbf{u}$, parameters θ, and time t. To ensure these implicit equations are well-posed it is necessary that $\frac{\partial \mathbf{f}}{\partial(\dot{\mathbf{x}},\mathbf{z})}$ be non-singular, an assumption referred to as index 1.

A common goal for an AWE system is to maximize average power output. The average power $\overline{P}$ over a trajectory can be written either as an integral over the trajectory or as a function of the state at final time T:

$$\overline{P} = \frac{1}{T}\int_0^T P(t)\mathrm{d}t = \frac{E(T)}{T} \tag{11.2}$$

where the energy harvested by the system E would be a differential state of the system satisfying $\dot{E} = P$. A quantity often penalized in an optimization problem is the integral of squared control action, which can also be written either as an integral term or as a final term. Finally, a number of constraints must be respected including bounds on variables such as actuator limits or minimum altitude, nonlinear inequalities $\mathbf{h}$ such as minimum airspeed or allowed range in angle of attack, and boundary conditions $\mathbf{c}$ which may be static or periodic.

An OCP can be written as:

$$\begin{array}{lll}
\underset{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.),\theta,T}{\text{minimize}} & J_\mathrm{M}(\mathbf{x}(T),\theta,T) + \displaystyle\int_0^T J_\mathrm{L}(\mathbf{x}(t),\mathbf{z}(t),\mathbf{u}(t),\theta,t,T)\,\mathrm{d}t & \\
\text{subject to} & 0 = \mathbf{f}(\dot{\mathbf{x}}(t),\mathbf{x}(t),\mathbf{z}(t),\mathbf{u}(t),\theta,t), & t \in [0,T] \\
& 0 \geq \mathbf{h}(\mathbf{x}(t),\mathbf{z}(t),\mathbf{u}(t),\theta,t), & t \in [0,T] \\
& \mathbf{x}_\mathrm{min} \leq \mathbf{x}(t) \leq \mathbf{x}_\mathrm{max}, & t \in [0,T] \\
& \mathbf{z}_\mathrm{min} \leq \mathbf{z}(t) \leq \mathbf{z}_\mathrm{max}, & t \in [0,T] \\
& \mathbf{u}_\mathrm{min} \leq \mathbf{u}(t) \leq \mathbf{u}_\mathrm{max}, & t \in [0,T] \\
& \theta_\mathrm{min} \leq \theta \leq \theta_\mathrm{max} & \\
& T_\mathrm{min} \leq T \leq T_\mathrm{max} & \\
& \mathbf{c}(\mathbf{x}(0),\mathbf{x}(T)) = 0 &
\end{array} \tag{11.3}$$

where J_M and J_L are the so called Mayer and Lagrange terms of the cost function.

11.2 Discretization by Direct Collocation

There are many techniques available to numerically solve OCPs of the form of Eq. (11.3). In this work we use the direct collocation method. The system is first approximated by discretization, and then solved with a general-purpose Nonlinear Program (NLP) solver.

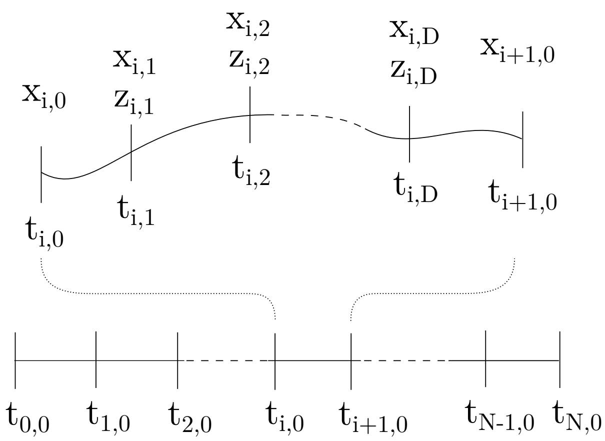

Fig. 11.1 Trajectory Discretization

In direct collocation, a trajectory is broken into N intervals $I_i = [t_{i,0}, t_{i+1,0}], i = 0, \dots N-1$ (Fig. 11.1). It is convenient to scale time on interval I_i according to $t = t_{i,0} + \tau \frac{T}{N}$ with $\tau \in [0,1]$. The differential state on interval I_i is approximated as a Lagrange interpolating polynomial $\mathbf{x}_i^D$ of degree D, with $D+1$ control points $\mathbf{x}_{i,j}$ placed respectively at τ_j:

$$\mathbf{x}_i^D(t) = \sum_{j=0}^{D} \xi_j(\tau)\mathbf{x}_{i,j} \tag{11.4}$$

where

$$\xi_j(\tau) = \prod_{k=0,k\neq j}^{D} \frac{\tau_k - \tau}{\tau_k - \tau_j}. \tag{11.5}$$

The time derivative of this polynomial on an intermediate point is given by

$$\dot{\mathbf{x}}_i^D(t_{i,0} + \tau\frac{T}{N}) = \sum_{j=0}^{D} \frac{N}{T}\frac{\mathrm{d}\xi_j(\tau)}{\mathrm{d}\tau}\mathbf{x}_{i,j} = \sum_{j=0}^{D} \frac{N}{T}\xi_j'(\tau)\mathbf{x}_{i,j}. \tag{11.6}$$

Given an initial value $\mathbf{x}_{i,0}$, the model equations can be satisfied by enforcing Eq. (11.1) at the collocation nodes $\tau_1, \dots, \tau_D$. This results in the collocation equations:

$$0 = \mathbf{f}\left(\sum_{k=0}^{D} \frac{N}{T} \xi_k'(\tau_j)\mathbf{x}_{i,k},\, \mathbf{x}_{i,j},\, \mathbf{z}_{i,j},\, \mathbf{u}_i,\, \theta,\, t_{i,j}\right),\; j = 1,\ldots,D. \tag{11.7}$$

When Eq. (11.7) is satisfied, the final value $\mathbf{x}_{i+1,0}$ can be recovered by evaluating Eq. (11.4) at $\tau = 1$:

$$\mathbf{x}_{i+1,0} = \mathbf{x}_i^D(t_{i+1,0}) = \sum_{j=0}^{D} \xi_j(1)\mathbf{x}_{i,j}. \tag{11.8}$$

The collocation points τ_j must be chosen as the roots of shifted Gauss-Jacobi polynomials so that Eq. (11.8) is an accurate Gauss quadrature integration [3].The special Gauss-Jacobi polynomials Gauss-Legendre or Gauss-Radau are often used for their A-stability and for their high-order accuracy. Numerical values for these roots can be found in [3], though it is convenient to use the SciPy function scipy.special.js_roots [10].

We summarize Eqns. (11.7) and (11.8) with $X_i = (\mathbf{x}_{i,1},\ldots,\mathbf{x}_{i,D})$, $Z_i = (\mathbf{z}_{i,1},\ldots,\mathbf{z}_{i,D})$, $i = 0,\ldots,N-1$, $\mathbf{x}_i = \mathbf{x}_{i,0}$, $i = 0,\ldots,N$, as

$$\begin{aligned} 0 &= \mathbf{G}(X_i, Z_i, \mathbf{u}_i, \theta, T) \\ \mathbf{x}_{i+1} &= \phi(\mathbf{x}_i, X_i). \end{aligned} \tag{11.9}$$

11.2.1 Quadrature States

In Sect. 11.1.1 it was stated that some integral terms such as Eq. (11.2) can be evaluated by adding a differential state to the problem and evaluating it at T. If this integral term is used only in the cost function, it can be beneficial to evaluate it without adding an additional state to the system.

Consider the problem where some derivative $\dot{q}$ is known at the collocation nodes, and $q(\tau = 1)$ should be computed by assuming that $q(\tau = 0) = 0$ and integrating over one collocation interval. Writing out Eq. (11.6) at the collocation nodes:

$$\begin{bmatrix} \dot{q}(\tau_1) \\ \vdots \\ \dot{q}(\tau_D) \end{bmatrix} = \frac{N}{T} \begin{bmatrix} \xi_1'(\tau_1) & \cdots & \xi_D'(\tau_1) \\ \vdots & \ddots & \vdots \\ \xi_1'(\tau_D) & \cdots & \xi_D'(\tau_D) \end{bmatrix} \begin{bmatrix} q(\tau_1) \\ \vdots \\ q(\tau_D) \end{bmatrix}, \tag{11.10}$$

and solving for $q(\tau_j)$ yields:

$$\begin{bmatrix} q(\tau_1) \\ \vdots \\ q(\tau_D) \end{bmatrix} = \frac{T}{N} \begin{bmatrix} \xi_1'(\tau_1) & \cdots & \xi_D'(\tau_1) \\ \vdots & \ddots & \vdots \\ \xi_1'(\tau_D) & \cdots & \xi_D'(\tau_D) \end{bmatrix}^{-1} \begin{bmatrix} \dot{q}(\tau_1) \\ \vdots \\ \dot{q}(\tau_D) \end{bmatrix}. \tag{11.11}$$

Combining this with Eq. (11.8) yields:

$$
\begin{aligned}
q(\tau=1) &= \begin{bmatrix} \xi_1(1) & \cdots & \xi_D(1) \end{bmatrix} \frac{T}{N} \begin{bmatrix} \xi_1'(\tau_1) & \cdots & \xi_D'(\tau_1) \\ \vdots & \ddots & \vdots \\ \xi_1'(\tau_D) & \cdots & \xi_D'(\tau_D) \end{bmatrix}^{-1} \begin{bmatrix} \dot{q}(\tau_1) \\ \vdots \\ \dot{q}(\tau_D) \end{bmatrix} \\
&= \frac{T}{N} \Lambda^T \begin{bmatrix} \dot{q}(\tau_1) \\ \vdots \\ \dot{q}(\tau_D) \end{bmatrix},
\end{aligned}
\tag{11.12}
$$

where Λ is a constant vector since both $\xi_j(\tau_k)$ and $\xi_j'(\tau_k)$ are constant. Integrating over all collocation intervals yields the value at T:

$$
q(T) = \frac{T}{N} \Lambda^T \sum_{i=0}^{N-1} \begin{bmatrix} \dot{q}(t_{i,1}) \\ \vdots \\ \dot{q}(t_{i,D}) \end{bmatrix} \tag{11.13}
$$

so the integral term of the cost function from Eq. (11.3) can be computed as:

$$
\int_0^T J_{\mathrm{L}}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \theta, t, T)\, \mathrm{d}t = \frac{T}{N} \Lambda^T \sum_{i=0}^{N-1} \begin{bmatrix} J_{\mathrm{L}}(\mathbf{x}_{i,1}, \mathbf{z}_{i,1}, \mathbf{u}_i, \theta, t_{i,1}, T) \\ \vdots \\ J_{\mathrm{L}}(\mathbf{x}_{i,D}, \mathbf{z}_{i,D}, \mathbf{u}_i, \theta, t_{i,D}, T) \end{bmatrix}. \tag{11.14}
$$

11.2.2 NLP Statement

The full NLP can then be written as

$$
\begin{aligned}
\underset{\mathbf{x}, X, Z, \mathbf{u}, \theta, T}{\text{minimize}} \quad & J_{\mathrm{M}}(\mathbf{x}_N, \theta, T) + \frac{T}{N} \sum_{j=1}^{D} \Lambda_j \sum_{i=0}^{N-1} J_{\mathrm{L}}(\mathbf{x}_{i,j}, \mathbf{z}_{i,j}, \mathbf{u}_i, \theta, t_{i,j}, T) & \\
\text{subject to} \quad & \mathbf{x}_{i+1} = \phi(\mathbf{x}_i, X_i), & i = 0, \ldots, N-1 \\
& 0 = \mathbf{G}(X_i, Z_i, \mathbf{u}_i, \theta, T), & i = 0, \ldots, N-1 \\
& 0 \geq \mathbf{h}(X_i, Z_i, \mathbf{u}_i, \theta, T), & i = 0, \ldots, N-1 \\
& \mathbf{x}_{\min} \leq \mathbf{x}_i \leq \mathbf{x}_{\max}, & i = 0, \ldots, N \\
& X_{\min} \leq X_i \leq X_{\max}, & i = 0, \ldots, N-1 \\
& Z_{\min} \leq Z_i \leq Z_{\max}, & i = 0, \ldots, N-1 \\
& \mathbf{u}_{\min} \leq \mathbf{u}_i \leq \mathbf{u}_{\max}, & i = 0, \ldots, N-1 \\
& \theta_{\min} \leq \theta \leq \theta_{\max} & \\
& T_{\min} \leq T \leq T_{\max} & \\
& \mathbf{c}(\mathbf{x}_0, \mathbf{x}_N) = 0 &
\end{aligned}
\tag{11.15}
$$

This problem can be solved with a general-purpose NLP solver.

11.3 Initial Guess by Homotopy Using Pseudo-Forces

Since the NLP is in general non-convex due to the nonlinear dynamics constraints, the NLP solver will only find a local solution. Which local solution is found depends greatly on the initial guess, and a bad initial guess can even cause the solver to diverge and be unable to find any feasible trajectory. A good initial guess is therefore essential in solving the problem. An initial trajectory could be generated by numerical simulation with an automatic feedback controller or a human in the loop, but this can take continuing effort to maintain. A homotopy strategy such as [6] can be implemented to automatically generate a reasonable initial guess.

Much of the nonlinearity in the dynamic equations arises from the forces $\mathbf{F}$ and moments $\mathbf{M}$ of the AWE system. A new system can be made which is identical except that the forces and moments are augmented by the addition of fictitious forces $\tilde{\mathbf{F}}$, $\tilde{\mathbf{M}}$:

$$\begin{pmatrix} \hat{\mathbf{F}} \\ \hat{\mathbf{M}} \end{pmatrix} = \gamma \begin{pmatrix} \mathbf{F} \\ \mathbf{M} \end{pmatrix} + (1-\gamma) \begin{pmatrix} \tilde{\mathbf{F}} \\ \tilde{\mathbf{M}} \end{pmatrix} \tag{11.16}$$

A parameter γ is stepped from 0 to 1 and a simple tracking problem with cost function

$$J = \sum_{i=0}^{N-1} \sum_{j=1}^{D} \left(||\mathbf{r}_{i,j} - \bar{\mathbf{r}}_{i,j}||_2^2 + \tilde{\mathbf{F}}_{i,j}^T \Sigma_{\mathbf{F}}^{-1} \tilde{\mathbf{F}}_{i,j} + \tilde{\mathbf{M}}_{i,j}^T \Sigma_{\mathbf{M}}^{-1} \tilde{\mathbf{M}}_{i,j} \right) \tag{11.17}$$

is solved at each γ, using each solution as the initial guess for the next problem. Here $\mathbf{r}$ is the aircraft's position in Cartesian coordinates $[x, y, z]$. The tracked trajectory $\bar{\mathbf{r}}$ is usually a simple circle or figure-eight, and the initial guess is given simply as the tracked trajectory itself, with attitude such that the aircraft nose is tangent to the velocity, the aircraft belly is pointing to the origin, and the wing is perpendicular to both.

Treating $\tilde{\mathbf{F}}_{i,j}$, $\tilde{\mathbf{M}}_{i,j}$ as design variables in the NLP allows the optimizer to freely choose forces and moments to ensure that the trajectory remains feasible at each γ. When γ is small, the trajectory is unrealistic as fictitious forces dominate. As γ approaches 1, the penalization $\Sigma_{\mathbf{F}}$ and $\Sigma_{\mathbf{M}}$ drive the fictitious forces to zero and the trajectory converges such that model equations are satisfied.

11.4 Two Periodic Optimal Control Problems

Using the model equations developed in Chap. 10, we have implemented the NLP as stated by Eq. (11.15) for a 0.6 kg, 0.1 m^2 kite described in [5]. We have used degree 4 interpolating polynomials with Radau polynomial roots as collocation points. In this work we use the solver IPOPT [15] with an interface provided by the optimization environment CasADi [1] which also delivers efficient model function derivatives.

11.4.1 A Six Degree of Freedom Tethered Aircraft Model

The system has differential states $\mathbf{x} = [\mathbf{r}, \dot{\mathbf{r}}, l, \dot{l}, \ddot{l}, R, \omega, \phi_{\text{ail}}, \phi_{\text{elev}}]$, where $\mathbf{r}$ is again the position in Cartesian coordinates $[x, y, z]$, algebraic variable λ is associated with the constraint $x^2 + y^2 + z^2 - l^2 = 0$, ω is the aircraft angular velocity in the body frame, R is the direction cosine matrix, and ϕ are aileron and elevator angles. The control inputs are $\mathbf{u} = [\dddot{l}, \dot{\phi}_{\text{ail}}, \dot{\phi}_{\text{elev}}]$. Using derivatives of control surface angles instead of the angles themselves will allow penalization and thus suppression of high frequency control inputs as explained in Sect. 11.4.2.

The model dynamics are:

$$\begin{bmatrix} mI_3 & 0 & \mathbf{r} \\ 0 & J & 0 \\ \mathbf{r}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{d}{dt}\dot{\mathbf{r}} \\ \frac{d}{dt}\omega \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\text{a}}(\mathbf{x}) - mg\mathbf{1}_3 \\ \mathbf{M}_{\text{a}}(\mathbf{x}) - \omega \times J\omega \\ -\dot{\mathbf{r}}^T\dot{\mathbf{r}} + \dot{l}^2 + l\ddot{l} \end{bmatrix}. \tag{11.18}$$

where $\mathbf{F}_{\text{a}}$, $\mathbf{M}_{\text{a}}$ are the aerodynamic forces and moments on the kite, $\mathbf{1}_3$ is the identity matrix, and J is the moment of inertia dyadic of the aircraft. The rotational kinematic equation is:

$$\dot{R} = R\Omega \tag{11.19}$$

where Ω is the skew matrix of ω. Combining Eqns. (11.18) and (11.19) with the trivial kinematics

$$\frac{d}{dt}\begin{bmatrix} \mathbf{r} \\ l \\ \dot{l} \\ \ddot{l} \\ \phi_{\text{ail}} \\ \phi_{\text{elev}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{l} \\ \ddot{l} \\ \dddot{l} \\ \dot{\phi}_{\text{ail}} \\ \dot{\phi}_{\text{elev}} \end{bmatrix} \tag{11.20}$$

yields the full model equations compatible with the form of Eq. (11.1), with $\frac{\partial \mathbf{f}}{\partial(\dot{\mathbf{x}}, \mathbf{z})}$ non-singular. The tether tension is λl, so the power harvested is $\lambda l \dot{l}$. A full derivation of these equations and modifications to include tether mass and bridling can be found in Chap. 10.

Because the model equations use non-minimal coordinates (i.e., there are more generalized coordinates than degrees of freedom), the constraint associated with λ and its derivative must be enforced as initial conditions:

$$\begin{aligned} 0 &= x(0)^2 + y(0)^2 + z(0)^2 - l(0)^2, \\ 0 &= x(0)\dot{x}(0) + y(0)\dot{y}(0) + z(0)\dot{z}(0) - l(0)\dot{l}(0). \end{aligned} \tag{11.21}$$

Likewise, the initial rotation matrix $R(0)$ must be orthonormal. This can be accomplished by enforcing the six upper or lower triangular components of:

$$0 = R(0)^T R(0) - I. \tag{11.22}$$

We use aerodynamic coefficients fit from wind tunnel data, and a simple log wind profile:

$$w(z) = w_0 \frac{\log\left(\frac{z+z_t}{z_t}\right)}{\log\left(\frac{z_0}{z_t}\right)} \tag{11.23}$$

with $w_0 = 10\frac{m}{s}$, $z_0 = 100$ and $z_t = 0.1$.

11.4.2 Maximum Power Crosswind Orbit

A trajectory which generates maximum power will want to maximize Eq. (11.2). The average power is computed using the approach from Sect. 11.2.1 with $J_{\mathrm{L}} = P/T = \lambda l\dot{l}/T$. Regularization on the control actions $\Sigma_{\mathbf{u}}$ is added, and the cost function is:

$$J = \sum_{i=0}^{N-1} \mathbf{u}_i^T \Sigma_{\mathbf{u}}^{-1} \mathbf{u}_i - \frac{1}{N} \sum_{j=1}^{D} \Lambda_j \sum_{i=0}^{N-1} \lambda_{i,j} l_{i,j} \dot{l}_{i,j}. \tag{11.24}$$

The regularization $\Sigma_{\mathbf{u}}$ is positive-definite and often diagonal, and has the effect of penalizing high bandwidth in the control surfaces, discouraging overly aggressive maneuvers. Regularization in the controls also keeps the optimization problem well-posed and improves convergence[11].

Because of the non-minimal coordinates, simply enforcing $\mathbf{x}_0 = \mathbf{x}_N$ to make the trajectory periodic results in an overconstrained NLP which will cause problems in the NLP solution. Our periodic conditions are $[l_0, y_0, z_0, \dot{l}_0, \dot{y}_0, \dot{z}_0, \ddot{l}_0, \omega_0, \phi_0] = [l_N, y_N, z_N, \dot{l}_N, \dot{y}_N, \dot{z}_N, \ddot{l}_N, \omega_N, \phi_N]$, and the three upper off-diagonal components of $R_0^T R_N = I$. These combined with Eqns. (11.21) and (11.22) are the boundary conditions.

Since the tether is modeled as a rigid constraint, tether tension must be constrained to be positive ($\lambda l \geq 0$). Angle of attack α is constrained to be less than the value at which stall is expected. Aerodynamic control surfaces ϕ were also bound to within reasonable values, and altitude was constrained to be positive. Simply constraining altitude positive permits the kite to fly at exactly ground level which would be disastrous in real life, but in this case wind shear causes the minimum altitude to be greater than zero. Solving for optimal trajectories which robustly respect safe minimum altitudes is a difficult problem outside the scope of this chapter, but a treatment can be found in [13, 14].

This NLP takes around 30 seconds to a minute to solve on a modern desktop computer for a grid of around $N = 100$. The optimized trajectory (Fig. 11.2) is the well-known one [7] where the kite reels out at around one-third the wind speed at high C_L for most of the trajectory, and then reels in as quickly as possible with low C_L (Fig. 11.3). In Fig. 11.4 the optimized trajectory is compared to the local steady state theoretical limit, known as Loyd's limit [12]. Power roughly tracks Loyd's limit but as the kite traverses the cycle, power first undershoots and then exceeds

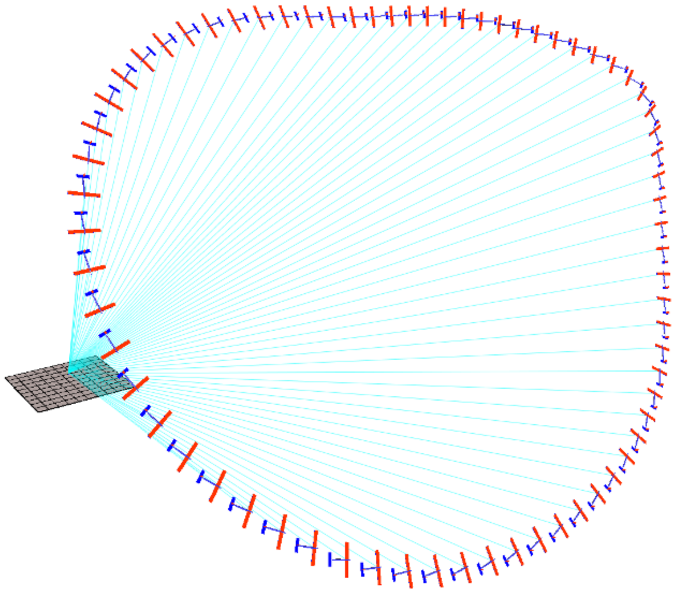

Fig. 11.2 Power optimal crosswind trajectory

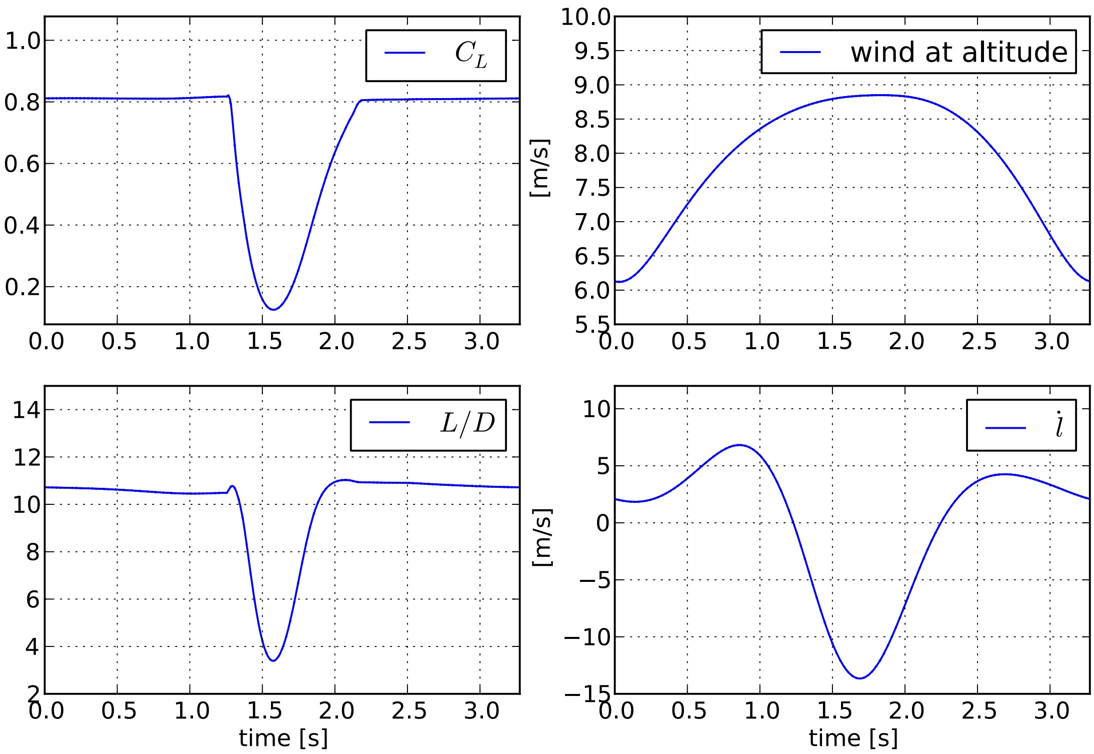

Fig. 11.3 Reel-out profile, local wind, and lift/drag coefficients in a power optimal crosswind trajectory.

Loyd's limit due to the assistance and hindrance of gravity. The average power does not exceed Loyd's limit.

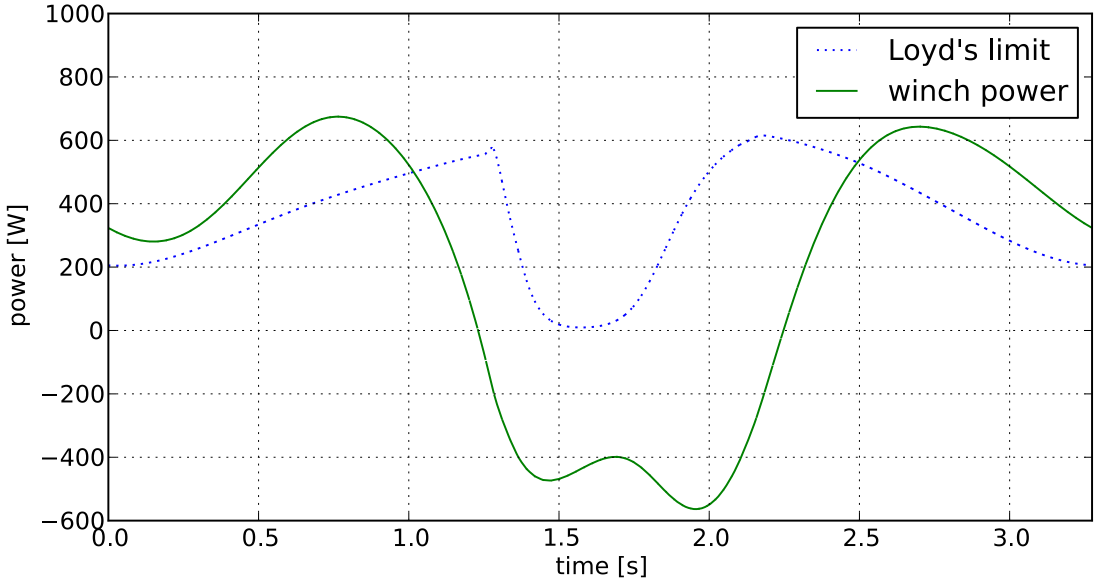

Fig. 11.4 Actual trajectory power compared to steady-state theoretical limit

11.4.3 Number of Loops Per Pumping Cycle

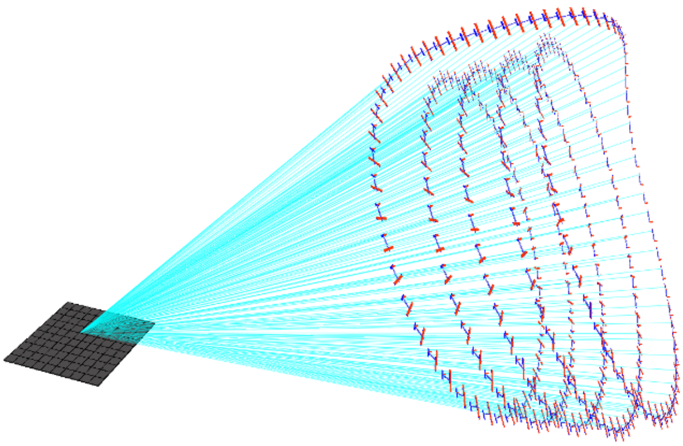

Fig. 11.5 Pumping trajectory with five loops per cycle

In a pumping system, the aircraft reels out for a number of loops before quickly reeling in (Fig. 11.5). An interesting question which is well suited to numerical optimization is how many loops to fly per cycle.

Because of the non-convexity of the problem, solving the same NLP as in Sect. 11.4.2 with different initial conditions can result in different locally power optimal trajectories. Simply concatenating the single-loop initial conditions n times as an initial guess usually results in a locally optimal n-loop trajectory, though it is possible to fall into another local optimum along the way.

A sweep was performed over initial number of loops n and average power was observed (Table 11.1). The solver converged as desired for up to seven loops per

Table 11.1 Effect of number of loops per reel out on average power

# loops	1	2	3	4	5	6	7
power (W)	328.9	344.6	350.2	353.0	354.5	355.1	355.3

cycle, but for eight loops the solver converged to a trajectory where four loops were flown per cycle, repeating twice per trajectory. This only indicates that the problem is indeed non-convex, and concatenating the optimal single loop trajectory eight times is not a good enough initial guess. Nonetheless, Table 11.1 shows that it is more efficient to fly multiple loops per reel-in, but there is insignificant gain after about four loops per cycle.

11.5 Startup Trajectory as Transition Between Two Periodic Orbits

An AWE system using a carousel for rotational startup must transition to crosswind flight. Assuming that an initial and final trajectory are known, the connecting trajectory must be found.

11.5.1 Rotational Holding Trajectory

The final crosswind trajectory was solved for in Sect. 11.4.2, but an initial periodic trajectory is needed which is easy and safe to fly. We use minimum weighed quadratic control actions for the cost function:

$$J = \sum_{i=0}^{N-1} \mathbf{u}_i^T \Sigma_{\mathbf{u}}^{-1} \mathbf{u}_i. \tag{11.25}$$

A typical holding trajectory is shown in Fig. 11.6.

11.5.2 Transition

The transition NLP has new boundary conditions and cost function, and two new parameters. The known initial and final trajectory are each fit with a Fourier expansion so that their differential states $\mathbf{x}$ become closed form functions $\psi(\theta)$ of one phase parameter:

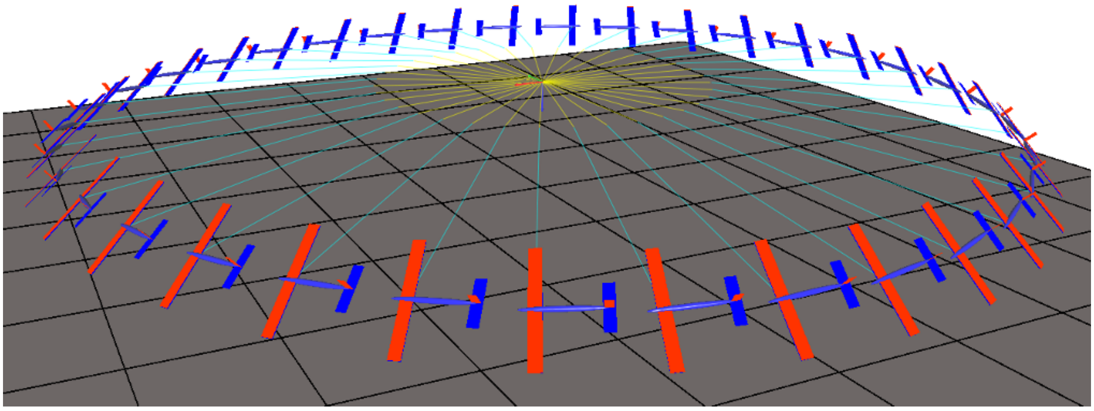

Fig. 11.6 Startup holding trajectory

$$\psi(\theta) = \mathbf{a}_0 + \sum_{k=1} \left(\mathbf{a}_k \cos\left(\frac{2\pi k}{T}\theta\right) + \mathbf{b}_k \sin\left(\frac{2\pi k}{T}\theta\right) \right) \tag{11.26}$$

where the coefficients $\mathbf{a}_k$, $\mathbf{b}_k$ are chosen to minimize

$$\sum_{i,j} ||\psi(\theta_{i,j}) - \mathbf{x}_{i,j}||_2^2 \tag{11.27}$$

for a chosen grid of $\theta_{i,j}$.

As in Sect. 11.4.2, enforcing $\mathbf{x}_0 = \psi_0(\theta_0)$, $\mathbf{x}_N = \psi_F(\theta_F)$ as boundary conditions would result in an overdetermined NLP. Like before, our conditions are

$$\begin{aligned} [l_0, y_0, z_0 \ldots] &= [\psi_{0,l}(\theta_0), \psi_{0,y}(\theta_0), \psi_{0,z}(\theta_0) \ldots] \\ [l_N, y_N, z_N \ldots] &= [\psi_{F,l}(\theta_F), \psi_{F,y}(\theta_F), \psi_{F,z}(\theta_F) \ldots] \end{aligned} \tag{11.28}$$

and the three upper off-diagonal components of

$$\begin{aligned} R_0^T \psi_{0,R}(\theta_0) &= I \\ R_N^T \psi_{F,R}(\theta_F) &= I. \end{aligned} \tag{11.29}$$

These combined with Eqns. (11.21) and (11.22) are the full boundary conditions.

The new parameters θ_0 and θ_F allow the transition to begin and end at arbitrary points on the initial and crosswind trajectories respectively. The choice of cost function here is subjective – transition should be made quickly but safely. One cost function which reflects this is a minimum time with a quadratic penalty on control inputs:

$$J = T + \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{u}_i^T \Sigma_{\mathbf{u}}^{-1} \mathbf{u}_i \tag{11.30}$$

The transition problem has a longer time scale than a periodic problem, so the number of collocation intervals must be higher. A typical transition is shown in Fig. 11.7.

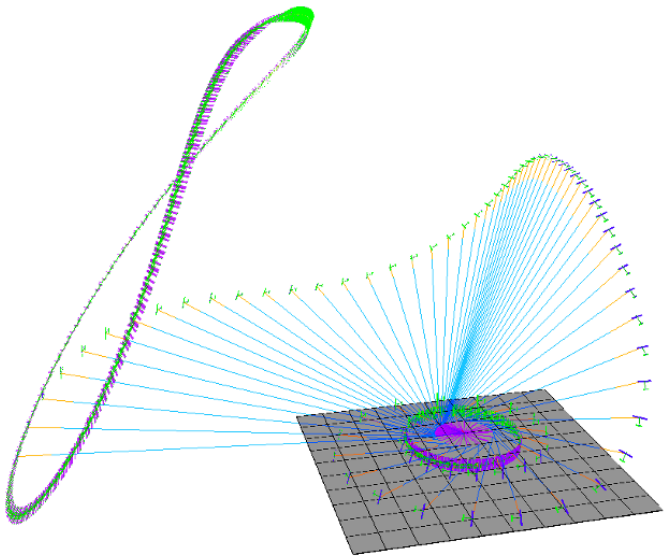

Fig. 11.7 Transition trajectory

11.6 Conclusions

This chapter has summarized the direct collocation technique for numerically solving optimal control problems. We summarized the model developed in Chap. 10 and showed how it can be used with collocation. A power maximization problem was solved, and the variation of optimal average power with number of loops per retraction was investigated. It was shown that for a small scale AWE system, efficiency could be improved by flying multiple loops but there was insignificant gain in more than about four loops per cycle. Finally, we presented a technique for solving for an optimal transition between two fixed trajectories.

Acknowledgements This research was supported by Research Council KUL: PFV/10/002 Optimization in Engineering Center OPTEC, GOA/10/09 MaNet and GOA/10/11 Global real- time optimal control of autonomous robots and mechatronic systems. Flemish Government: IOF/KP/SCORES4CHEM, FWO: PhD/postdoc grants and projects: G.0320.08 (convex MPC), G.0377.09 (Mechatronics MPC); IWT: PhD Grants, projects: SBO LeCoPro; Belgian Federal Science Policy Office: IUAP P7 (DYSCO, Dynamical systems, control and optimization, 2012-2017); EU: FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), Eurostars SMART, ACCM.

References

1. Andersson, J., Åkesson, J., Diehl, M.: CasADi – A symbolic package for automatic differentiation and optimal control. In: Forth, S., Hovland, P., Phipps, E., Utke, J., Walther, A. (eds.) Recent Advances in Algorithmic Differentiation, Vol. 87, Lecture Notes in Computational Science and Engineering, pp. 297–307. Springer, Berlin (2012). doi: 10.1007/978-3-642-30023-3_27

2. Argatov, I., Silvennoinen, R.: Energy conversion efficiency of the pumping kite wind generator. Renewable Energy **35**(5), 1052–1060 (2010). doi: 10.1016/j.renene.2009.09.006
3. Biegler, L.: Nonlinear Programming. MOS-SIAM Series on Optimization. SIAM, Philadelphia (2010). doi: 10.1137/1.9780898719383
4. Fagiano, L., Milanese, M., Piga, D.: Optimization of airborne wind energy generators. International Journal of Robust and Nonlinear Control **22**(18), 2055–2083 (2011). doi: 10.1002/rnc.1808
5. Geebelen, K., Ahmad, H., Vukov, M., Gros, S., Swevers, J., Diehl, M.: An experimental test set-up for launch/recovery of an Airborne Wind Energy (AWE) system. In: Proceedings of the 2012 American Control Conference, pp. 5813–5818, Montréal, Canada, 27–29 June 2012. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6315033
6. Gros, S., Diehl, M.: A Relaxation Strategy for the Optimization of Airborne Wind Energy Systems. In: Proceedings of the European Control Conference (ECC13), Zurich, Switzerland, 17–19 July 2013
7. Houska, B.: Robustness and Stability Optimization of Open-Loop Controlled Power Generating Kites. M.Sc.Thesis, Ruprecht-Karls-Universität, Heidelberg, 2007. http://www.kuleuven.be/optec/files/Houska2007a.pdf
8. Houska, B., Diehl, M.: Optimal control for power generating kites. In: Proceedings of the 9th European Control Conference, pp. 3560–3567, Kos, Greece, 2–5 July 2007. http://www.kuleuven.be/optec/files/Houska2007.pdf
9. Houska, B., Diehl, M.: Optimal control of towing kites. In: Proceedings of the 45th IEEE Conference on Decision and Control, pp. 2693–2697, San Diego, CA, USA, 13–15 Dec 2006. doi: 10.1109/CDC.2006.377210
10. Jones, E., Oliphant, T., Peterson, P.: SciPy: Open source scientific tools for Python. http://www.scipy.org/ (2001). Accessed 5 July 2013
11. Kameswaran, S., Biegler, L.: Simultaneous dynamic optimization strategies: Recent advances and challenges. Computers and Chemical Engineering **30**(11–12), 1560–1575 (2006). doi: 10.1016/j.compchemeng.2006.05.034
12. Loyd, M. L.: Crosswind kite power. Journal of Energy **4**(3), 106–111 (1980). doi: 10.2514/3.48021
13. Sternberg, J., Gros, S., Houska, B., Diehl, M.: Approximate Robust Optimal Control of Periodic Systems with Invariants and High-Index Differential Algebraic Systems. In: Proceedings of the 7th IFAC Symposium on Robust Control Design, pp. 678–683, Aalborg, Denmark, 20–22 June 2012. doi: 10.3182/20120620-3-DK-2025.00089
14. Sternberg, J., Goit, J., Gros, S., Meyers, J., Diehl, M.: Robust and Stable Periodic Flight of Power Generating Kite Systems in a Turbulent Wind Flow Field. In: Proceedings of the 15th IFAC Workshop on Control Applications of Optimization, Rimini, Italy, 13–16 Sept 2012. doi: 10.3182/20120913-4-IT-4027.00009
15. Wächter, A., Biegler, L.: On the Implementation of a Primal-Dual Interior Point Filter Line a Search Algorithm for Large-Scale Nonlinear Programming. Mathematical Programming **106**(1), 25–57 (2006). doi: 10.1007/s10107-004-0559-y
16. Williams, P., Lansdorp, B., Ockels, W.: Optimal Crosswind Towing and Power Generation with Tethered Kites. AIAA Journal of Guidance, Control, and Dynamics **31**(1), 81–93 (2008). doi: 10.2514/1.30089