Comparison of Wind Turbine Operating Transitions Through the Use of Iterative Learning Control

Jason Laks, Lucy Pao, Andrew Alleyne

Abstract—In below-rated wind conditions, a wind turbine operates to maximize the amount of available power harvested from the wind and is said to be operating in region 2. In above-rated wind conditions, where regulation is the main objective to prevent over power and speed faults and to mitigate loads, the turbine is said to be in region 3. There is no standard method for operation at the boundary of the two regions and transitions between them can be problematic. In this study, we use iterative learning control to determine the control actuation necessary to accurately track idealized candidate trajectories during the transition between regions 2 and 3. The amount of control actuation required to track a transition trajectory and the ability to do so with minimal collateral loading determines which trajectory is most amenable for a given turbine. Trajectories are also graded by the average power produced during transition since they take the turbine off of the optimal power point.

I. INTRODUCTION

All wind turbine control systems must manage a transition between optimizing power and preventing over-speed and power conditions that occur as wind speeds vary. In below-rated conditions where the turbine cannot produce rated power (region 2), the ratio of the blade-tip speed to wind speed (also known as tip-speed ratio or TSR) is optimized to harvest the maximum power from the wind. In above-rated conditions (region 3), speed and power are regulated at safe levels and if possible, blade pitch and torque are adjusted to reduce structural loading. The boundary between operating regions 2 and 3 is determined by the turbine’s rated power capability and maximum or rated rotor speed.

Other than very slow adjustments of turbine yaw, the actuators available on a utility scale turbine almost always consist only of blade pitch and the ability to adjust the load torque and power output of the generator. In region 2, pitch actuation is typically not used; and in region 3, generator torque is essentially determined by the maximum rated power output and therefore often held constant. Evaluating typical transitions between the regions where these two controls become active is the subject of this study.

Iterative learning control (ILC) is used to compute the pitch and generator torque control signals required to follow candidate trajectories under prescribed wind conditions. Transitions are evaluated for rapidly varying wind conditions that begin and end at the same speed. This is a fair basis for evaluating power performance since the deficit accrued while the rotor is decelerating is balanced against the extra power generated when the rotor is decelerating.

The computation of control signals is not simply a matter of inverting a plant model because the turbine system is quite nonlinear in the transition region. However, the time-varying formulation of ILC is well suited to this problem. This technique provides a means of iteratively computing the control signals required to keep the turbine at the desired operating points as wind speed changes. However, ILC requires highly repeatable conditions as a prerequisite. Hence, in application to wind turbines, this study advocates ILC as a powerful analysis and design tool and not a control method, since it is not possible to arrange for such repeatable conditions in the typical wind turbine environment.

This rest of this paper is organized into four parts and a conclusion. In section II we describe the computation of the transition profiles. Then section III formulates the ILC method used to compute the necessary control actions. The results are presented in section IV. In the remainder of the paper we use the ILC results to evaluate the candidate transition profiles. Comparisons of power, required control authority, and transient load performance, serve as a basis for grading the candidate transition profiles.

II. TRANSITION REGION TRAJECTORIES

A. Operating Regions

A well controlled turbine is a passive aero-dynamic load on the air mass flowing within the area swept out by the rotor blades. The fraction of available wind power captured is characterized by the turbine’s power coefficient $C_p$. This coefficient is a function $C_p(\lambda, \beta)$ of the turbine geometry and operating point as determined by blade pitch $\beta$ and the ratio $\lambda$ of the blade-tip speed to wind speed (i.e., TSR). In this study, we use a numerically computed power coefficient for a model of the three-bladed, controls advanced-research turbine (CART3) at the National Renewable Energy Laboratory’s (NREL’s) National Wind Technology Center (NWTC).

In below-rated wind conditions, the objective is to adjust rotor speed so that it tracks with wind speed to maintain the TSR that maximizes $C_p$. Contours of constant $C_p$ are computed using the code WT_Perf [1] developed at NWTC and are depicted in Fig. 1(a). The amount of power harvested is given by the product of $C_p(\lambda, \beta)$ and the available power in the wind, and can be expressed as [2]

$$P_{aero} = \frac{1}{2} \rho A R^3 \frac{C_p(\lambda, \beta)}{\lambda^3} \Omega^3$$

(1)

where $A$, $R$, $\Omega$, and $\rho$ are the turbine swept area, rotor radius, rotor speed, and air density, respectively. With these definitions, if wind speed is denoted by “w”, TSR is given by

$$\lambda = \frac{\Omega R}{w}.$$ 

(2)

Ideally, in region 2 operation, the turbine will operate at the point $(\lambda_{opt}, \beta_{opt})$ where the power coefficient is a maximum. In region 3, rotor speed is regulated to a constant value by increasing pitch; as a consequence in region 3, TSR decreases with increasing wind speed and the operating point shifts down and to the right in Fig. 1.
The region 3 intercept is chosen to insure that all profiles have linearly to intercept with a pitch angle that holds speed at rated. In any case, a common practice is to introduce a transition region (often referred to as region 2.5) so that generator torque hits rated before speed reaches rated (and the turbine is producing rated power). We evaluate the three transitions, U, C, and L in Fig. 2(a). Their specification is completed by determining the wind speeds that balance with prescribed rotor speed, generator load, and blade pitch. We now describe a method of choosing pitch transitions and then the associated wind speed and TSR.

With pitch near \( \beta_{opt} \), changes in wind speed generate significant variation in aerodynamic torque, so it may be desirable to adjust the pitch set point as the turbine approaches rated speed. In this study, as soon as the prescribed torque profile \( \tau_{gen}(\Omega) \) deviates from the square law, the prescribed pitch set points \( \beta(\Omega) \) are increased linearly to intercept with a pitch angle that holds speed at rated in region 3. The result is displayed in the upper plot of Fig. 2(b). The region 3 intercept is chosen to insure that all profiles have appreciable pitch-to-aerodynamic torque control-authority when the rotor speed hits rated and pitch regulation of speed is required.

Having specified torque and pitch as functions of rotor speed \( \Omega \), the torque contour data (Fig. 1(b)), is used to determine corresponding TSR profiles \( \lambda(\Omega) \) as a function of rotor speed. That is, for each pitch-torque-speed operating point, fix the specified pitch \( \beta(\Omega) \), scale the torque contour data in Fig. 1(b) by \( (\Omega/35)^2 \) and then find the aerodynamic torque balancing against \( \tau_{gen}(\Omega)N_{gb} \), then the intercept at the left (in Fig. 1(b)) gives the corresponding TSR \( \lambda(\Omega) \). The result of this procedure is presented in the lower plot of Fig. 2(a). This also provides a corresponding wind speed

\[
w = \frac{\Omega R}{\lambda(\Omega)} \triangleq W(\Omega).
\]

With these relationships determined, any wind speed can be mapped to a corresponding rotor speed and blade pitch. Where we originally prescribed generator torque and pitch as functions of rotor speed, we now can schedule them as functions of wind speed:

\[
\lambda(w) = W^{-1}(w),
\]

\[
\tau_{gen}(w) = K_s\Omega(w),
\]

\[
\beta(w) = \beta(\Omega(w)).
\]
III. ITERATIVE LEARNING CONTROL

ILC is a technique that computes system inputs that drive the response to track a desired reference— in our case a rotor-speed profile as computed in section II. The problem of computing system inputs cannot simply be formulated as a plant inversion or a linear quadratic regulator problem, since the focus is specifically on a region of non-linear operation.

ILC techniques can compute control inputs that achieve good tracking of desired references in the presence of unknown disturbances. What is required is that the tracking maneuver be attempted repeatedly and that the disturbance stay the same for each iteration. For the purposes of this study, the unknown disturbance is the effect of system non-linearities and wind, along with any other aspect of the output not predicted by a linear, zero-initial-state model. In addition, because of the change in polarity of the pitch-to-torque profiles, and because there are hard limits on the maximum allowable generator torque, we utilize a constrained ILC algorithm.

A. Models for ILC Computation and Simulation

1) Linear Approximation of the Non-Linear Turbine: Along any response trajectory, perturbations in the response of the non-linear dynamics are computed based on small perturbations in turbine control. More explicitly, if the turbine response is determined as a function of state $x$ and input $u$ according to non-linear dynamics

$$\frac{d}{dt} x = F(x, u, w)$$

$$y = G(x, u, w),$$

where $y$ is a vector of system outputs, then the change in state $\Delta x$ and output trajectories $\Delta y$ are approximately given as solutions of

$$\frac{d}{dt} \Delta x(t) = F(x(t) + \Delta x(t), u(t) + \Delta u(t), w) - F(x(t), u(t), w)$$

$$\Delta y(t) = G(x(t) + \Delta x(t), u(t) + \Delta u(t), w) - G(x(t), u(t), w)$$

where

$$A(t) = \frac{\partial F}{\partial x}(x(t), u(t)), \quad B(t) = \frac{\partial F}{\partial u}(x(t), u(t)),$$

$$C(t) = \frac{\partial G}{\partial x}(x(t), u(t)), \quad D(t) = \frac{\partial G}{\partial u}(x(t), u(t)).$$

That is, $(A(t), B(t), C(t), D(t))$ are the Jacobian matrices of the vector functions $F$ and $G$ with respect to their first and second arguments and are evaluated along known state and control trajectories $x(t)$ and $u(t)$. We are interested in finding the necessary change $\Delta u(t)$ such that the resulting change in trajectory $\Delta \phi(t)$ puts the turbine on the desired transition profile. Finally, we note that the wind excitation varies as a function of time, but that function does not change during this process. We repeat the experiment/simulation for each new estimate of $\Delta u$, but during the course of successive experiments, there is no $\Delta u(t)$ to contend with.

Now, assume that the control used at each trial is implemented in discrete time using a zero-order hold (i.e., it is constant between sample hits). At the $k$th sample time $t_k$, define $x(k) \triangleq \Delta x(t_k)$ and $u(k) \triangleq \Delta u(t_k)$. Then the time-varying linear system in (10) can be approximated by its discrete-time equivalent

$$x_{j+1}(k+1) = A_j(k)x_{j+1}(k) + B_j(k)u_{j+1}(k)$$

$$y_{j+1}(k) = C_j(k)x_{j+1}(k) + D_j(k)u_{j+1}(k)$$

where we have introduced the index $j$ to denote iteration. In particular, the discrete-time matrices $(A_j(k), B_j(k), C_j(k), D_j(k))$ are the zero-order-hold equivalents to the linearization $(A(t_k), B(t_k), C(t_k), D(t_k))$ at each sample hit, along the response of the system during the previous iteration. The control $u_{j+1}(k)$ is the perturbation sequence to be determined for use in the next iteration. Details surrounding the computation of the
discrete-time model from the continuous-time model can be found in Franklin, Powell and Workman [5].

2) Non-Linear System Response: At the conclusion of each iteration, the non-linear response is obtained through simulation of the turbine model using the turbine modeling code FAST [6] developed at NWTC. This code also numerically computes linearized models of the turbine at specified rotor and wind speeds and with specified blade pitch. In implementation, we use an array of pre-computed linearized, state-space models obtained from FAST over a 25-by-25 grid of TSR's and blade pitch angles. The matrices used in (11) are the ones from the array that are closest to the operating points (TSR and blade pitch) observed at each sample hit during the previous simulation.

The degrees of freedom (i.e., mechanical dynamics) modeled during linearization represent rotational inertia, a second-order drive-train torsional compliance, and a second-order blade compliance that represents fore-aft/out-of-plane flapping of the turbine blades. During simulation of the non-linear model, additional degrees of freedom are included that encompass another blade flapping mode, an edge-wise (in the plane of the rotor) blade flapping mode, two tower fore-aft modes and two tower side-to-side modes. This is considered a fairly high-fidelity turbine model, but introduces a fair amount of discrepancy between the ILC computation model ((11)) and the full non-linear simulation.

**B. Norm Optimal ILC as Finite-Horizon Tracking Control**

Prior to each simulation, we compute a perturbation to the nominal control input to reduce the tracking error. Computation of the control perturbation is done with respect to the linearized system (dropping the iteration index $j$ from the notation)

\[
\begin{align*}
x(k+1) &= A(k)x(k) + B(k)u(k), \quad (12a) \\
y(k) &= C(k)x(k) + D(k)u(k), \quad (12b) \\
e(k) &= y_d(k) - y(k), \quad (12c)
\end{align*}
\]

where $y_d(k)$ is the desired output (e.g., the rotor-speed profile computed in section II), and $y_{nl}(k)$ is the output of the non-linear system. We assume that our linear model is correct in which case it follows that

\[
y_{nl}(k) = y(k) + d(k),
\]

where $d(k)$ is any part of the response not accounted for by the linear model. Note that $y_{nl}$ is never computed using (13), but instead, it is obtained from simulation. The sequence $e(k)$ is the tracking error that we intend to minimize via the control perturbation $u(k)$ for this iteration, and is yet to be determined. Here it is to be understood that the matrices $(A(k), B(k), C(k), D(k))$ are always the ones that are closest to the operating points observed during the previous iteration. In lieu of the iteration index $j$, we will distinguish any signal obtained from the previous iteration using a tilde notation.

Note that in (12), the “disturbance” $d(k)$ for this iteration is strictly unknown during computation of $u(k)$. However, assume the new control $u(k)$ will not change too much from the $u(k)$ used in the previous iteration. Then we can approximate $d(k)$ with the disturbance $\tilde{y}_{nl}(k) - \tilde{y}(k)$ observed during the previous iteration/simulation. So, for the current iteration we have

\[
y_{nl}(k) = y(k) + (\tilde{y}_{nl}(k) - \tilde{y}(k)).
\]

Substituting this into (12c) for the present iteration gives

\[
e(k) = y_d(k) - (\tilde{y}_{nl}(k) - \tilde{y}(k)) - y(k) = \tilde{y}_r(k) - y(k).
\]

Here, $\tilde{y}_r = (y_d - \tilde{y})$ is a computed reference for the linear model based on the previous iteration. In the absence of modeling errors and the approximation $d = \tilde{d}$, finding the control $u(k)$ that produces this output from the linear model would then cancel with the disturbance $d(k)$ and produce the desired, total response $y_d(k)$.

However, since the final $d(k)$ is unknown, and the formulation hinges on $u = \tilde{u}$, norm optimal ILC finds the control that optimizes a cost function that penalizes large changes in control from one iteration to the next, and the total size of the control. That is, we compute the control minimizing the quadratic norm

\[
f_0(u, x) = \frac{1}{2} \sum_{k=1}^{N} \left[ (y(k) - \tilde{y}_r(k))^T Q_k (y(k) - \tilde{y}_r(k)) + u(k)^T S_k u(k) + (u(k) - \tilde{u}(k)) R_k (u(k) - \tilde{u}(k)) \right] + \frac{1}{2} x_f^T \Pi_{N+1} x_f + r_a (N+1)^T x_f
\]

where the matrices $Q_k$, $S_k$ and $R_k$ are symmetric weights that penalize tracking error, control effort and the change in control, respectively. We also require at least one of $R_k$ or $S_k$ to be positive definite. The part of the cost involving $x_f$, the final state at sample $k = N+1$, is derived from the requirement that the system ends up in a state that minimizes the tracking error without any control:

\[
f_0(u, x) = \frac{1}{2} (C(N+1) x_f - \tilde{y}_r(N+1))^T Q_f \times (C(N+1) x_f - \tilde{y}_r(N+1)) + \frac{1}{2} x_f^T C(N+1)^T Q_f C(N+1) x_f + \frac{1}{2} x_f^T C(N+1)^T Q_f C(N+1) x_f + r_a (N+1)^T x_f
\]

where constant terms have been dropped in the second line.

The cost function in (16) can be organized by writing it in terms of a generalized tracking error $z(k)$

\[
f_0(u, x) = \frac{1}{2} \sum_{k=1}^{N} \left[ y(k) - \tilde{y}_r(k) \right]^T Q_k \left[ \begin{array}{c} u(k) - \tilde{u}(k) \\ 0 \\ 0 \\ 0 \end{array} \right] + \frac{1}{2} x_f^T \Pi_{N+1} x_f + r_a (N+1)^T x_f,
\]

Further, the generalized tracking error can be computed in terms of a generalized reference $\tau(k)$ as

\[
z(k) = \left[ \begin{array}{c} C(k) \\ 0 \\ 0 \\ 0 \end{array} \right] x(k) + \left[ \begin{array}{c} D(k) \\ I_s \end{array} \right] u(k) + \left[ \begin{array}{c} (y_d - \tilde{y}_r)(k) \\ 0 \\ 0 \end{array} \right],
\]

\[
\tilde{y}_r(k) = C_s(k) x(k) + D_{us}(k) u(k) + D_r(k) \tau(k),
\]

\[
x(k+1) = A(k) x(k) + B(k) u(k) + \left[ \begin{array}{c} 0 \\ 0 \end{array} \right] \tilde{y}_r(k),
\]

\[
A(k) x(k) + B(k) u(k) + B_r(k) \tau(k).
\]

4315
**C. Lifted Representation Of Finite Horizon Tracking Control**

In this section, we provide an overview of the methods used in this study for quadratic optimization with equality and inequality constraints. The dimensions of the systems of equations that result can be even larger than those encountered in the more straightforward, lifted ILC implementation [7]. This computational burden is circumvented to a large extent by relying heavily on Riccati recursions to solve the required systems of equations.


\[ v = [u(1)^T \ x(2)^T \ \ldots \ u(N)^T \ x_{1}^T]^T. \]  

(20)

Then, by substituting (19a) for \( z(k) \) into the cost (18) and dropping terms independent of \( u(k) \) and \( x(k) \), the optimization problem can be posed as

\[
\begin{align*}
\min_v & \quad f_0(v) = \frac{1}{2} v^T Hv + g^T v, \\
\text{subj} & \quad C_{eq}v = b, 
\end{align*}
\]  

(21a)

(21b)

where

\[
H = \begin{bmatrix}
Q_u(1) & 0 & 0 & \ldots & 0 \\
0 & Q_u(2) & S_{xu}(2) & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & \Pi_{N+1} \\
\end{bmatrix},
\]  

(22a)

\[
g = \begin{bmatrix}
g_u(1) \\
g_u(2) \\
\vdots \\
g_u(N) \\
g_u(N+1) \\
\end{bmatrix},
\]  

(22b)

\[
g_u(1) = S_{xu}(1)^T x(1) + S_{ur}(1)r(1),
\]  

(23a)

\[
g_u(k) = S_{ur}(k)r(k),
\]  

(23b)

\[
g_u(k) = S_{xu}(k)r(k),
\]  

(23c)

\[
g_u(N+1) = T_u(N+1),
\]  

(23d)

and

\[
Q_u(k) = C_u(k)^T Q_u(k)C_u(k),
\]  

(24a)

\[
Q_u(k) = D_u(k)^T Q_u(k)D_u(k),
\]  

(24b)

\[
S_{xu}(k) = C_u(k)^T Q_u(k)D_u(k),
\]  

(24c)

\[
S_{ur}(k) = C_u(k)^T Q_u(k)D_r(k),
\]  

(24d)

\[
S_{xu}(k) = C_u(k)^T Q_u(k)D_r(k),
\]  

(24e)

\[
The equality constraint derives from the state equation (19b), so that
\]

\[
C_{eq} = \begin{bmatrix}
-B(1) & I & 0 & 0 & \ldots & 0 \\
0 & -A(2) & -B(2) & I & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 0 & I
\end{bmatrix},
\]  

(25)

\[
b = \begin{bmatrix}
A(1)x(1) + B_r(1)r(1) \\
B_r(2)r(2) \\
\vdots \\
B_r(N)r(N)
\end{bmatrix}.
\]  

(26)

Since the cost and constraints are convex functions of the optimization variable, an optimizing solution can be found by solving the associated Karush-Kuhn-Tucker (KKT) system [8]

\[
\begin{bmatrix}
H & C_{eq} \\
C_{eq}^T & 0
\end{bmatrix} \begin{bmatrix}
u \\
\gamma
\end{bmatrix} = \begin{bmatrix}
-g \\
b
\end{bmatrix},
\]  

(27)

where \( \gamma \) is a Lagrange multiplier. The dimensions of this problem can become quite large, but it can be decomposed to obtain a Riccati recursion [8].

Alternatively, one can take a dynamic programming approach starting from (18): peel off the last term of the summation, substitute in for \( z(N) \) and \( x_f = x(N+1) \) in terms of \( x(N) \) and \( u(N) \) using (19a) and (19b), respectively, and solve for the last control \( u(N) \) in terms of \( x(N) \) and the reference \( r(N) \). The result will be a control of the form

\[
u(N) = -K(N)x(N) + u_{ref}(N).
\]  

(28)

This result can be substituted back into the cost (18) and with some simplification it becomes apparent that the process can be back-stepped to the first sample hit at \( k = 1 \). This latter approach, in our opinion, is more straightforward and we provide the results here without derivation. We also note that (29f) below, is a discrete-time Riccati recursion. The optimal control is obtained by first recursing

\[
4316
\]
Fig. 4. The target profiles are evaluated using the wind variation plotted in (a). The ILC weights are constructed as shown in (b), to emphasize difficult tracking sections and bias the torque command towards rated. Also shown in both plots is the rotor-speed response to the nominal torque and pitch controls.

This form of the recursions is convenient for application to the case where inequalities are used and a gradient search is required. Without inequality constraints, (30) is redundant. Also, without inequality constraints, these recursions show it is possible to solve for an optimal control without computing the Lagrange multiplier \( \gamma \). However, the multiplier is required for the gradient search, and by comparing the recursion obtained from decomposition of the KKT system (27) with the recursion given above, it can be shown that (partitioning the \( \gamma \) vector conformably with the number of states)

\[
\gamma(k) = -\Pi_{k+1} x(k+1) - r_u(k+1). 
\]

2) Lifted Representation with Inequality Constraints: In our application, we wish to apply constraints on the pitch and torque controls. In general, it is straightforward to include in the problem statement constraints on any output that can be computed from the control and system state (e.g., \( y_p(k) = C_p(k)x(k) + D_p(k)u(k) \leq Y_{max}(k) \)). Writing such constraints in terms of the optimization variable (20), the optimization problem becomes

\[
\begin{align*}
\min_v \quad & f_0(v) = \frac{1}{2} v^T Hv + g^T v, \\
\text{subj} : \quad & C_{eq} v = b, \\
& P v \leq h,
\end{align*}
\]

where

\[
P = \begin{bmatrix}
D_{eqd}(1) & 0 & 0 & \ldots & 0 \\
-\lambda_{eqd}(1) & 0 & 0 & \ldots & 0 \\
0 & C_p(2) & D_{eq}(2) & \ldots & 0 \\
0 & -C_p(2) & -D_{eq}(2) & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & C_f
\end{bmatrix},
\]

\[
h = \begin{bmatrix}
Y_{dmax}(1) - D_{eqd}(1)r(1) - C_{eq}(1)x(1) \\
-\lambda_{dmin}(1) + D_{eqd}(1)r(1) + C_{eq}(1)x(1) \\
Y_{max}(2) - D_{eq}(2) & -D_{eq}(2) \\
-Y_{min}(2) + D_{eq}(2) & -D_{eq}(2) \\
\vdots & \vdots & \ddots & \vdots & \\
f
\end{bmatrix}.
\]

The final inequality \( C_f x(N+1) \leq f \) normally requires the final state to be in some acceptable polygon.

The cost and the constraints are again convex, so any solution of the associated KKT system gives optimal control and state sequences [8]. However, in this case, the KKT system will contain products between the optimization vector and Lagrange variables, so that the solutions must be obtained using a numerical method.

We take the primal-dual approach in [8] and perform a gradient search to find solutions of the modified KKT system

\[
\begin{align*}
\lambda & = H v + g + C_{eq}^T \gamma + P^T \lambda = 0, \\
\gamma & = C_{eq} v - b = 0, \\
\lambda & = \Lambda d + t = 0,
\end{align*}
\]

where \( d \triangleq P v - h \), the vectors \( \gamma \) and \( \lambda \) are Lagrange multipliers, \( \Lambda \triangleq \text{diag}(\Lambda) \), and \( t \) is a parameter updated each step of the search.

The initial guess at \( v \) must satisfy \( d < 0 \); then, as described in [8], the algorithm includes safeguards to insure

\[
\begin{align*}
P v - h & < 0, \\
\lambda & > 0,
\end{align*}
\]
Finding a solution to (35) entails using Newton’s method to compute search directions. Given an initial guess at values for \((v, \gamma, \lambda)\), it is possible to show that a search direction \((\Delta_v, \Delta_\gamma, \Delta_\lambda)\) can be obtained by solving

\[
\begin{bmatrix}
 H - P^T D^{-1} AP & C_{eq}^T \\
 C_{eq} & 0 
\end{bmatrix}
\begin{bmatrix}
 \Delta_v \\
 \Delta_\gamma 
\end{bmatrix} =
\begin{bmatrix}
 P^T D^{-1} r_v - r_v \\
 -r_\gamma 
\end{bmatrix}
\tag{37}
\]

where \(D = \text{diag}(d)\), and then performing the computation

\[
\Delta_\lambda = -D^{-1} r_v - D^{-1} AP \Delta_v.
\tag{38}
\]

The algorithm then finds new values for the optimization variables \((v, \gamma, \lambda)\) searching along the direction \((\Delta_v, \Delta_\gamma, \Delta_\lambda)\). When the new values solve (35) to the desired precision, the search is terminated; otherwise a new search direction is computed and the process is repeated.

Because of the form of \(P\), the block structure of the system in (37) is identical with the block structure in (27). Therefore, this system can also be solved using the Riccati recursion as follows. Decompose \(\bar{H} = H - P^T D^{-1} AP\) as in (22a), and then compute the recursions (29) and (31) with the substitutions

\[
Q_v(k) = \tilde{Q}_v(k),
\tag{39a}
Q_\alpha(k) = \tilde{Q}_\alpha(k),
\tag{39b}
S_{2u}(k) = \tilde{S}_{2u}(k),
\tag{39c}
g = r_v - P^T D^{-1} r_\lambda,
\tag{39d}
b = -r_\gamma.
\tag{39e}
\]

This gradient search is required for each new computation of constrained controls (i.e., each ILC iteration). The computation of the search direction (the Riccati recursion) may need to be repeated many times to obtain a new control for the next simulation. This is in contrast with the case without inequality constraints, where the Riccati recursion is done only once per iteration. The details surrounding the use of the search direction in the course of the primal-dual algorithm can be found in [8].

D. ILC Configuration for Application to the Wind Turbine

In order to effectively apply the approach laid out so far, the wind turbine plant is generalized slightly as depicted by the grey block in Fig. 3(a). We apply hard constraints to the size of the control perturbation such that the total control remains within desired bounds. This makes penalizing the total size of the control redundant and therefore, \(S_k\) in eq’s (16) and (18) is omitted and \(u(k)\) is not included in \(z(k)\).

However, we find that the high frequency content of the controls needs to be penalized in order to achieve monotonic decreases in the iteration-to-iteration tracking error. So, the plant is augmented with high-pass filters \(F_{hf}\) (as shown in Fig. 3(a)) having particular emphasis at the drive-train resonant frequency. In effect, the generalized output becomes \(y = [\Omega, \tau_{gen}, \tau_{hf}, p_{hf}]^T\), where \(\tau_{hf}\) and \(p_{hf}\) are high-passed versions of the torque and pitch commands, respectively. The torque command \(\tau_{gen}\) is also fed through to an output so that a penalty can be applied that biases this control towards rated. Hence, the desired reference becomes \(y_d = [\Omega(w), \tau_{rated}, 0, 0]^T\).

The weight matrix \(R_k\) penalizing change in the iteration-to-iteration control effort is diagonal, but constant. The tracking weight matrix \(Q_k\) is also diagonal as shown in Fig. 3(b).

All three target profiles are the same until they deviate from the particular emphasis at the drive-train resonant frequency. In effect, the generalized output becomes \(y = [\Omega, \tau_{gen}, \tau_{hf}, p_{hf}]^T\), where \(\tau_{hf}\) and \(p_{hf}\) are high-passed versions of the torque and pitch commands, respectively. The torque command \(\tau_{gen}\) is also fed through to an output so that a penalty can be applied that biases this control towards rated. Hence, the desired reference becomes \(y_d = [\Omega(w), \tau_{rated}, 0, 0]^T\).

The weight matrix \(R_k\) penalizing change in the iteration-to-iteration control effort is diagonal, but constant. The tracking weight matrix \(Q_k\) is also diagonal as shown in Fig. 3(b). The control estimates become nearly constant within 40 iterations, at which point the computed commands achieve peak tracking errors of less than 1% for all profiles. Evident in the spikes of total cost at iterations 5, 9, 13, 17, and 21 in Fig. 3(b), is the fact that the cost function is modified by increasing \(W_\Omega\) by a factor of four every four iterations, for a total of five increases. In addition, at every 25th iteration, the penalty on the generator-torque high frequency content is reduced by half.

The resulting pitch and torque controls are presented in Fig. 5(a). All three target profiles are the same until they deviate from the square law (see Fig. 2(a)) and so the resulting commands are identical until about 18 sec. All profiles require the turbine to “motor” (apply power to the rotor) in order to operate at the optimal TSR during the fast increase in wind speed used in this study.
Fig. 6. Performance metrics for the three candidate transition profiles. However, it is found that the torque can be biased near rated during deceleration and thereby recoup much of the power loss that is incurred during acceleration.

As can be observed in Fig. 5(a), profile L requires the least high frequency actuation and this is consistent with the smoother variation in this profile’s prescribed speed. However, the difference in actuation appears more significant than the differences in the speed profiles presented in Fig. 4(a). Further, profile L results in lower loads on both the blades and the rotor shaft (see Fig. 5(b)).

V. EVALUATION OF TRANSITION PROFILES

At this point, it would appear that Profile L is more natural for the turbine in that it requires marginally less actuation to achieve the target rotor speeds. However, in operation, power and load performance are viewed as more important. Two of the most important loads to consider are drive train and blade fatigue.

In terms of the twisting moment experienced by the rotor shaft (in the lower plot, Fig. 5(b)) profile L appears to create slightly lower loads. More significant is the reduction in blade-root bending moment as can be observed (in the top plot, Fig. 5(b)). This is a result of the fact that profile L ends up with larger/better pitch angles during acceleration. Further, in the absence of catastrophic loads, it is the number and size of the peak-to-peak excursions that produces fatigue wear and in this case as well, profile L is superior.

Damage equivalent load (DEL) is an accepted metric for measuring fatigue wear [9]. In Fig. 6 we compare the resulting DEL’s and average power produced by each profile during the region 2-3-2 cycle. The lower loads exhibited by profile L are achieved with a lower power output. However, profile L’s power output is lower by only about 2%, while its blade-flap DEL is lower by as much as 40%—a very significant decrease. If the 40% decrease in load is borne out in more turbulent conditions, it would imply that the diameter of the rotor can be increased (increasing the amount of power the turbine can harvest) and this could easily make up for the loss in power. There would be some optimal increase in diameter that balances increasing loads (due to a larger bending-radius/moment) with increasing power.

Therefore, we would select profile L as superior. It remains to be seen how a feedback control system will fare with the candidate profiles since, normally, it does not benefit either from ILC or a scheduling of nominal controls based on wind speed. However, there is great interest in the use of advanced measurement technologies such as light detection and ranging (LIDAR) that promise to make set-point scheduling based on wind speed feasible. In particular, our work in LIDAR-based model predictive control [10] will be able to make use of the results of this study directly, since that architecture will explicitly schedule set points based on wind speed.

VI. CONCLUSIONS

Several candidate transition profiles are compared by determining the control actuation and loading required to track set points in changing wind conditions. The perturbations to the nominal, scheduled control set points indicate which profiles may be unrealistic in changing wind conditions. Overall, the approach taken here promises to be an effective method to evaluate transition profiles without the additional variables that would be introduced with the inclusion of a feedback controller and its optimization. However, the approach can also easily be combined with any existing feedback controller.

ILC is an effective technique in computing the control actions required to track candidate profiles. The modeling process had errors in that the non-linear system was modeled as linear, in that the order/degrees of freedom in the linear model were different than the actual system, and in that further errors were introduced due to the fact that the available linearizations were only approximate to the actual operating points. Never the less, ILC still successfully computed the required controls.

Finally, we note that one should consider that it is possible to affect the results (and conclusions) through the choice of the ILC weights. In this study, we were able to bias torque towards rated during deceleration in order to recoup power lost during motoring. In actual operation, this strategy is only straight forward if the system is able to schedule based on wind speed. Similarly, we could also have biased pitch towards larger values during acceleration and thereby reduced the loads exhibited by profiles U and C at the expense of increased torque actuation. The effect and validity of such biases needs to be carefully considered in the evaluation of the results.

REFERENCES


