2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

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 accelerating 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 β and the ratio λ 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]

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

where A, R, Ω, and ρ 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

$$\lambda = \frac{\Omega R}{\mathrm{w}}. \tag{2}$$

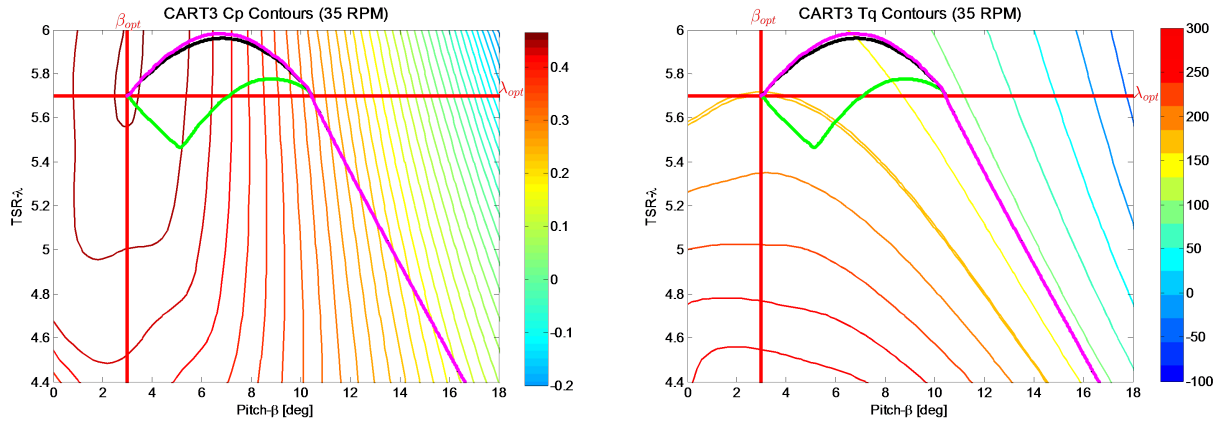
Ideally, in region 2 operation, the turbine will operate at the point $(\lambda_{opt}, \beta_{opt})$ where the power coefficient is a maximum. In region 3, rotor speed is regulated to a constant value by increasing pitch; as a consequence in region 3, TSR decreases with increasing wind speed and the operating point shifts down and to the right in Fig. 1. The

The authors are grateful for support from an industry sponsor.

Jason Laks is a doctoral candidate in the Dept. of Electrical, Computer, and Energy Engineering, University of Colorado, Boulder, e-mail: jhlaks@colorado.edu.

Lucy Pao is the Richard & Joy Dorf Professor in the Dept. of Electrical, Computer, and Energy Engineering, University of Colorado, Boulder.

Andrew Alleyne is a professor in the Dept. of Mechanical Science and Engineering, University of Illinois, Urbana-Champaign.



(a) Power Coefficient Contours

(b) Torque Contours

Fig. 1. Typical power and torque contours. Also shown are possible routes traced as the operating point transitions from region 2 $(\lambda_{opt}, \beta_{opt})$ to region 3.

path traced out as the turbine transitions between regions 2 and 3 is not unique and three possibilities are depicted.

Aero-dynamic torque $(= P/\Omega)$ is given by

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

and contours of constant torque are depicted in Fig. 1(b). Inspection of the torque contours shows that when pitch is near β_{opt}, there is less sensitivity to pitch. This– combined with the fact that changes in pitch away from β_{opt} only decrease aero-dynamic efficiency– limits the use of pitch as a control input in region 2. So, in region 2, torque control is used and while in region 3, speed is regulated using pitch while holding torque at rated to maintain maximum power.

In region 2 it is common practice to compute the generator torque that balances against the aero-dynamic torque at the maximum power point [3]. This is known as the square law (wherein N_{gb} is the ratio of the gear box transferring torque from the low-speed rotor shaft to the high-speed generator shaft):

$$\begin{aligned}\tau_{gen}(\Omega) = \frac{\tau_{aero}(\lambda_{opt}, \beta_{opt})}{N_{gb}} &= \frac{1}{2N_{gb}}\rho A R^3 \frac{C_p(\lambda_{opt}, \beta_{opt})}{\lambda_{opt}^3}\Omega^2 \\ &\triangleq K_g \Omega^2. \end{aligned} \tag{4}$$

This essentially completes the specification of operating points (nominal control set-points) in region 2. Blade pitch is set at β_{opt}, generator torque follows the square law, and the speed of the turbine is not actively controlled. For the CART3, in region 3, generator torque is held constant and speed is regulated using blade pitch. Thus, in region 3 the operating points are determined by the blade pitch required to keep aero-dynamic and generator torque balanced.

B. Transitions Between Operating Regions

As wind speeds increase and the turbine increases rotor speed to track with available power, there is an eventual transition from region 2 to region 3. This transition can be problematic [4], since the torque control may saturate before pitch control becomes active to regulate speed. This is depicted in Fig. 2(a) where rated torque and power occur at the upper-right of the top plot.

In region 2, use of the square law $(\tau_{gen} = K_g\Omega^2)$ brings the generator torque to rated before the rotor speed reaches the rated maximum, as depicted by Profile C in Fig. 2(a). This is the case for the CART3, but other turbine designs may result in the square law hitting rated speed before rated torque. 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 β_{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 Ω, 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

$$\mathrm{w} = \frac{\Omega R}{\lambda(\Omega)} \triangleq W(\Omega). \tag{5}$$

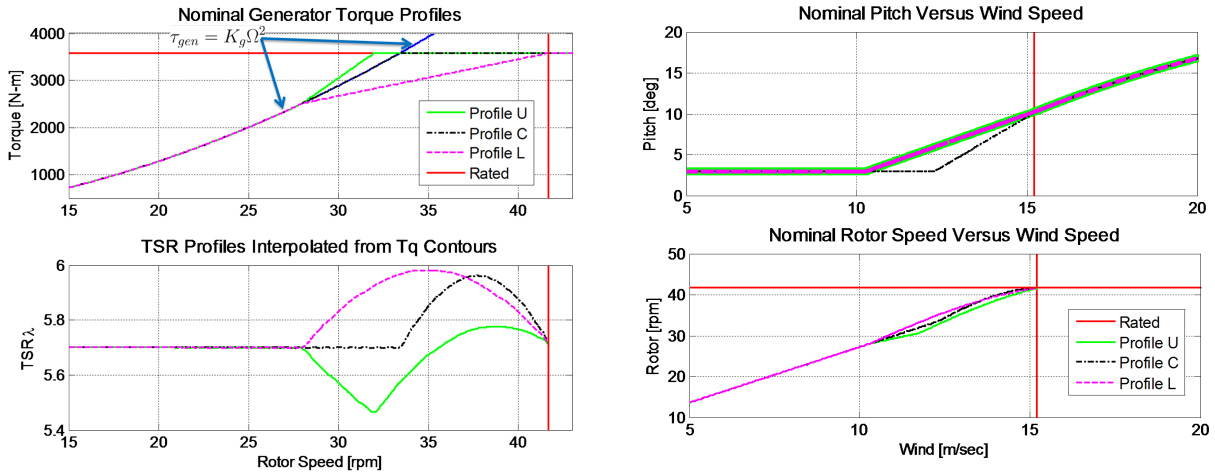
With these relationships determined, any wind speed can be mapped to a corresponding rotor speed and blade pitch. Where we originally *prescribed* generator torque and pitch as functions of rotor speed, we now can *schedule* them as functions of wind speed:

$$\Omega(\mathrm{w}) = W^{-1}(\mathrm{w}), \tag{6}$$

$$\tau_{gen}(\mathrm{w}) = K_g\Omega(\mathrm{w}), \tag{7}$$

$$\beta(\mathrm{w}) = \beta(\Omega(\mathrm{w})). \tag{8}$$

We now consider the three rotor-speed profiles (as functions of wind speed as in Fig. 2(b) or as functions of time as in Fig. 4(a)) as objectives in varying wind conditions and determine the change in pitch and generator torque required to keep rotor speed on these goals. This is done for the wind speed variation presented in



(a) Torque And TSR Profiles

(b) Rotor speed and Pitch Profiles

Fig. 2. Possible operating point transitions between regions 2 and 3 plotted versus rotor speed (a) and versus wind speed (b). Once generator torque and pitch are prescribed, there is then a corresponding TSR (bottom of plot in (a)) that generates a balancing aerodynamic torque.

Fig. 4(a); the corresponding variation in rotor speed as a function of time is obtained using (6). This wind variation requires that the turbine accelerate and decelerate symmetrically as shown. We will find that there will be significant differences ($\gg 5\%$) in the loading to which the blades are subjected, despite the fact that there are only subtle differences in the target speed profiles.

These operating point profiles have been formulated using static considerations. When wind changes as a function of time, simply changing torque and pitch according to the static profiles will not account for the accelerations that are required to keep the rotor speed on the specified trajectory. This issue is resolved using ILC.

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

$$\frac{d}{dt}x = F(x,u,\mathrm{w}) \quad (9a)$$
$$y = G(x,u,\mathrm{w}), \quad (9b)$$

where y is a vector of system outputs, then the change in state Δ_x and output trajectories Δ_y are approximately given as solutions of

$$\begin{aligned}\frac{d}{dt}\Delta_x(t) &\triangleq F(x+\Delta_x,u+\Delta_u,\mathrm{w}) - F(x,u,\mathrm{w})\\ &\approx \mathbf{A}(t)\Delta_x(t) + \mathbf{B}(t)\Delta_u(t)\\ \Delta_y(t) &\triangleq G(x+\Delta_x,u+\Delta_u,\mathrm{w}) - G(x,u,\mathrm{w})\\ &\approx \mathbf{C}(t)\Delta_x(t) + \mathbf{D}(t)\Delta_u(t)\end{aligned}$$

where

$$\mathbf{A}(t) = \left.\frac{\partial F}{\partial x}\right|_{(x(t),u(t))}, \qquad \mathbf{B}(t) = \left.\frac{\partial F}{\partial u}\right|_{(x(t),u(t))}, \quad (10a)$$
$$\mathbf{C}(t) = \left.\frac{\partial G}{\partial x}\right|_{(x(t),u(t))}, \qquad \mathbf{D}(t) = \left.\frac{\partial G}{\partial u}\right|_{(x(t),u(t))}. \quad (10b)$$

That is, $(\mathbf{A}(t),\mathbf{B}(t),\mathbf{C}(t),\mathbf{D}(t))$ 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 $\Delta_x(t)$ 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 Δ_u, but during the course of successive experiments, there is no $\Delta_{\mathrm{w}}(t)$ 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

$$x_{j+1}(k+1) = A_j(k)x_{j+1}(k) + B_j(k)u_{j+1}(k), \quad (11a)$$
$$y_{j+1}(k) = C_j(k)x_{j+1}(k) + D_j(k)u_{j+1}(k), \quad (11b)$$

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 $(\mathbf{A}(t_k),\mathbf{B}(t_k),\mathbf{C}(t_k),\mathbf{D}(t_k))$ 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)

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k), && \text{(12a)}\\ y(k) &= C(k)x(k) + D(k)u(k), && \text{(12b)}\\ e(k) &= y_d(k) - y_{nl}(k). && \text{(12c)} \end{aligned}$$

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

$$y_{nl}(k) = y(k) + d(k), \tag{13}$$

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 $\widetilde{u}(k)$ used in the previous iteration. Then we can approximate $d(k)$ with the disturbance $\widetilde{y}_{nl}(k) - \widetilde{y}(k)$ observed during the previous iteration/simulation. So, for the current iteration we have

$$y_{nl}(k) \approx y(k) + (\widetilde{y}_{nl}(k) - \widetilde{y}(k)). \tag{14}$$

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

$$e(k) = \underbrace{y_d(k) - (\widetilde{y}_{nl}(k) - \widetilde{y}(k))}_{\widetilde{y}_r(k)} - y(k) = \widetilde{y}_r(k) - y(k) \tag{15}$$

Here, $\widetilde{y}_r$ $(= y_d - \widetilde{d})$ is a computed reference for the linear model based on the previous iteration. In the absence of modeling errors and the approximation $d \approx \widetilde{d}$, 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 $u \approx \widetilde{u}$, 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

$$\begin{aligned} f_0(u,x) =& \frac{1}{2}\sum_{k=1}^{N}\big[(y(k)-\widetilde{y}_r(k))^T Q_k (y(k)-\widetilde{y}_r(k))\\ &+u(k)^T S_k u(k) + (u(k)-\widetilde{u}(k))R_k(u(k)-\widetilde{u}(k))\big]\\ &+\frac{1}{2}x_f^T \Pi_{N+1} x_f + r_a(N+1)^T x_f \end{aligned} \tag{16}$$

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:

$$\begin{aligned} \frac{1}{2}e_f^T e_f =& \frac{1}{2}(C(N+1)x_f - \widetilde{y}_r(N+1))^T Q_f\\ &\times (C(N+1)x_f - \widetilde{y}_r(N+1))\\ =& \frac{1}{2}x_f^T \underbrace{C(N+1)^T Q_f C(N+1)}_{\Pi_{N+1}} x_f\\ &+ \underbrace{(-\widetilde{y}_r(N+1)^T Q_f C(N+1))}_{r_a(N+1)^T} x_f, \end{aligned} \tag{17}$$

where constant terms have been dropped in the second line.

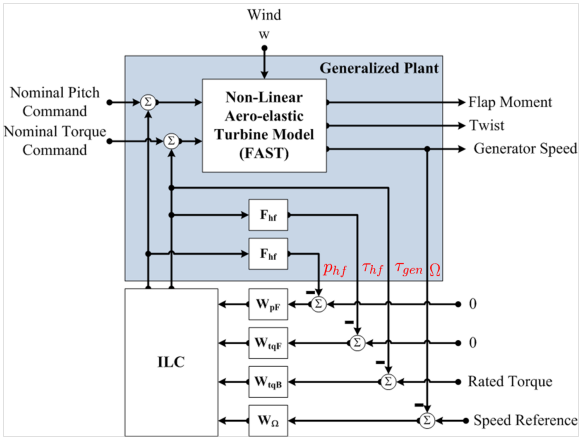
The cost function in (16) can be organized by writing it in terms of a generalized tracking error $z(k)$

$$\begin{aligned} f_0(u,x) =& \frac{1}{2}\sum_{k=1}^{N} \underbrace{\begin{bmatrix} y(k)-\widetilde{y}_r(k)\\ u(k)-\widetilde{u}(k)\\ u(k)\end{bmatrix}^T}_{z(k)^T} \underbrace{\begin{bmatrix} Q_k & 0 & 0\\ 0 & R_k & 0\\ 0 & 0 & S_k\end{bmatrix}}_{Q_z(k)} \underbrace{\begin{bmatrix} y(k)-\widetilde{y}_r(k)\\ u(k)-\widetilde{u}(k)\\ u(k)\end{bmatrix}}_{z(k)}\\ &+\frac{1}{2}x_f^T \Pi_{N+1} x_f + r_a(N+1)^T x_f,\\ =& \frac{1}{2}\sum_{k=1}^{N} z(k)^T Q_z(k) z(k) + \frac{1}{2}x_f^T \Pi_{N+1} x_f + r_a(N+1)^T x_f. \end{aligned} \tag{18}$$

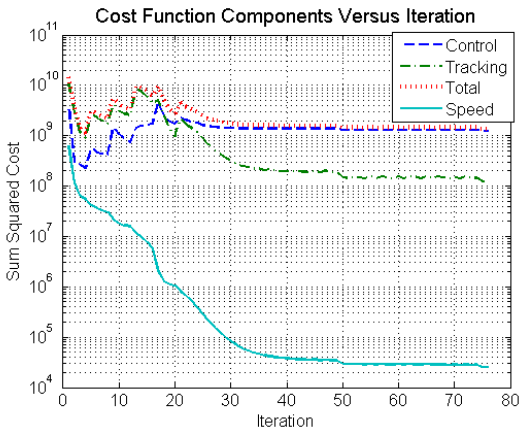
Further, the generalized tracking error can be computed in terms of a generalized reference $r(k)$ as

$$\begin{aligned} z(k) =& \underbrace{\begin{bmatrix} C(k)\\ 0\\ 0\end{bmatrix}}_{C_z(k)} x(k) + \underbrace{\begin{bmatrix} D(k)\\ I_u\\ I_u\end{bmatrix}}_{D_u(k)} u(k) + \underbrace{\begin{bmatrix} -I_y & 0\\ 0 & -I_u\\ 0 & 0\end{bmatrix}}_{D_r(k)} \underbrace{\begin{bmatrix}\widetilde{y}_r(k)\\ \widetilde{u}(k)\end{bmatrix}}_{r(k)},\\ =& C_z(k)x(k) + D_u(k)u(k) + D_r(k)r(k), \end{aligned} \tag{19a}$$

$$\begin{aligned} x(k+1) =& A(k)x(k) + B(k)u(k) + \underbrace{\begin{bmatrix}0 & 0\end{bmatrix}}_{B_r(k)} \underbrace{\begin{bmatrix}\widetilde{y}_r(k)\\ \widetilde{u}(k)\end{bmatrix}}_{r(k)},\\ =& A(k)x(k) + B(k)u(k) + B_r(k)r(k). \end{aligned} \tag{19b}$$



(a) Turbine Configuration for Generalized Tracking



(b) ILC Costs Versus Iteration

Fig. 3. Augmenting the turbine system to emphasize high frequency content of control effort (F_{hf} as shown in (a)) provides monotonic decreases in speed-tracking error (as in (b)). Speed tracking error, as shown in (b), is un-weighted, but the other components depicted include the cost function weighting.

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 straight forward, 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.

1) Lifted Representation with Equality Constraints: Define the lifted [7] optimization variable

$$v = [u(1)^T \; x(2)^T \ldots \; u(N)^T \; x_f^T]^T. \tag{20}$$

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

$$\min_v \quad f_0(v) = \frac{1}{2} v^T H v + g^T v, \tag{21a}$$

$$\text{subj:} \quad C_{eq} v = b, \tag{21b}$$

where

$$H = \begin{bmatrix} Q_u(1) & 0 & 0 & \cdots & 0 \\ 0 & Q_x(2) & S_{xu}(2) & \cdots & 0 \\ 0 & S_{xu}(2)^T & Q_u(2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \Pi_{N+1} \end{bmatrix}, \tag{22a}$$

$$g = \begin{bmatrix} g_u(1)^T & g_x(2)^T & \cdots & g_u(N)^T & g_x(N+1)^T \end{bmatrix}^T, \tag{22b}$$

$$g_u(1) = S_{xu}(1)^T x(1) + S_{ur}(1) r(1), \tag{23a}$$

$$g_u(k) = S_{ur}(k) r(k), \tag{23b}$$

$$g_x(k) = S_{xr}(k) r(k), \tag{23c}$$

$$g_x(N+1) = r_a(N+1), \tag{23d}$$

and

$$Q_x(k) = C_z(k)^T Q_z(k) C_z(k), \tag{24a}$$

$$Q_u(k) = D_u(k)^T Q_z(k) D_u(k), \tag{24b}$$

$$S_{xu}(k) = C_z(k)^T Q_z(k) D_u(k), \tag{24c}$$

$$S_{xr}(k) = C_z(k)^T Q_z(k) D_r(k), \tag{24d}$$

$$S_{ur}(k) = D_u(k)^T Q_z(k) D_r(k). \tag{24e}$$

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

$$C_{eq} = \begin{bmatrix} -B(1) & I & 0 & 0 & \cdots & 0 \\ 0 & -A(2) & -B(2) & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & I \end{bmatrix}, \tag{25}$$

$$b = \begin{bmatrix} A(1)x(1) + B_r(1) r(1) \\ B_r(2) r(2) \\ \vdots \\ B_r(N) r(N) \end{bmatrix}. \tag{26}$$

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]

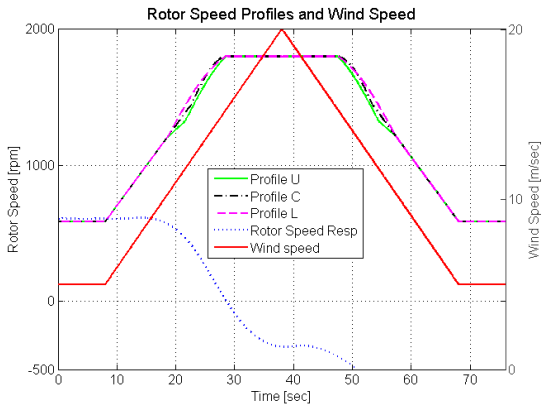
$$\begin{bmatrix} H & C_{eq}^T \\ C_{eq} & 0 \end{bmatrix} \begin{bmatrix} v \\ \gamma \end{bmatrix} = \begin{bmatrix} -g \\ b \end{bmatrix}, \tag{27}$$

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

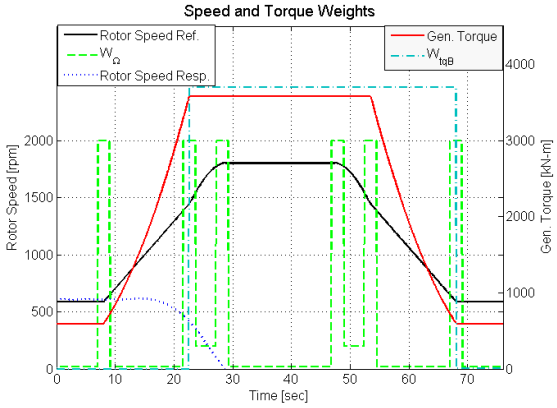
Alternatively, one can take a dyanmic programming approach starting from (18): peel off the last term of the summation, substitute in for $z(N)$ and $x_f = x(N+1)$ in terms of $x(N)$ and $u(N)$ using (19a) and (19b), respectively, and solve for the last control $u(N)$ in terms of $x(N)$ and the reference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

$$u(N) = -K(N) x(N) + u_{ref}(N). \tag{28}$$

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 straight forward 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



(a) Rotor-speed Prrofiles



(b) ILC Weights Relative to Speed and Torque Profiles

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.

backwards

$$R_u(k) = Q_u(k) + B(k)^T \Pi_{k+1} B(k), \tag{29a}$$
$$M(k) = S_{xu}(k) + A(k)^T \Pi_{k+1} B(k), \tag{29b}$$
$$K(k) = R_u(k)^{-1} M(k)^T, \tag{29c}$$
$$r_a(k) = (A(k) - B(k)K(k))^T (r_a(k+1) + \Pi_{k+1} b(k)) + g_x(k) - K(k)^T g_u(k), \tag{29d}$$
$$u_{ref}(k) = -R_u(k)^{-1} \left[g_u(k) + B(k)^T (r_a(k+1) + \Pi_{k+1} b(k)) \right], \tag{29e}$$
$$\Pi_k = Q_x(k) + A(k)^T \Pi_{k+1} A(k) - M(k) R_u(k)^{-1} M(k)^T, \tag{29f}$$

where

$$r_a(N+1) = g_x(N+1), \tag{30}$$

and then recursing forwards

$$u(1) = u_{ref}(1), \tag{31a}$$
$$u(k) = -K(k)x(k) + u_{ref}(k), \tag{31b}$$
$$x(k+1) = A(k)x(k) + B(k)u(k) + b(k). \tag{31c}$$

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 γ. 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 γ vector conformably with the number of states)

$$\gamma(k) = -\Pi_{k+1} x(k+1) - r_a(k+1). \tag{32}$$

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

$$\min_v \quad f_0(v) = \frac{1}{2} v^T H v + g^T v, \tag{33a}$$
$$\text{subj:} \quad C_{eq} v = b, \tag{33b}$$
$$P v \le h, \tag{33c}$$

where

$$P = \begin{bmatrix} D_{upd}(1) & 0 & 0 & \cdots & 0 \\ -D_{und}(1) & 0 & 0 & \cdots & 0 \\ 0 & C_p(2) & D_{up}(2) & \cdots & 0 \\ 0 & -C_n(2) & -D_{un}(2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & C_f \end{bmatrix}, \tag{34a}$$

$$h = \begin{bmatrix} Y_{dmax}(1) - D_{rpd}(1)r(1) - C_{pd}(1)x(1) \\ -Y_{dmin}(1) + D_{rnd}(1)r(1) + C_{nd}(1)x(1) \\ Y_{max}(2) - D_{rp} r(2) \\ -Y_{min}(2) + D_{rn} r(2) \\ \vdots \\ f \end{bmatrix}. \tag{34b}$$

The final inequality $C_f x(N+1) \le f$ 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.

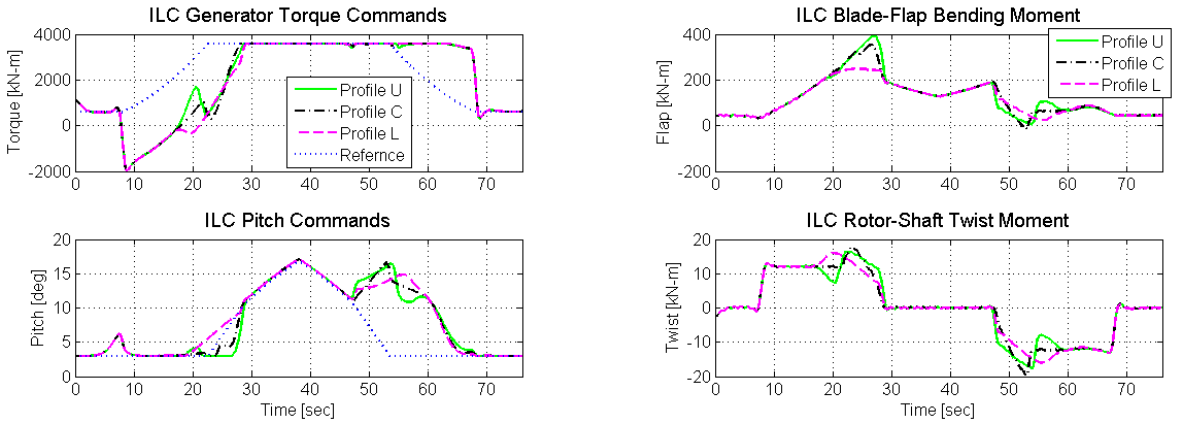
We take the primal-dual approach in [8] and perform a gradient search to find solutions of the modified KKT system

$$r_v \triangleq Hv + g + C_{eq}^T \gamma + P^T \lambda = 0, \tag{35a}$$
$$r_\gamma \triangleq C_{eq} v - b = 0, \tag{35b}$$
$$r_\lambda \triangleq \Lambda d + t = 0, \tag{35c}$$

where $d \triangleq Pv - h$, the vectors γ and λ 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

$$Pv - h < 0, \tag{36a}$$
$$\lambda > 0, \tag{36b}$$



(a) Resulting Torque and Pitch Commands

(b) Resulting Bend and Twist Loads

Fig. 5. Results after the final iteration: (a) the resulting torque and pitch commands for each profile; (b) the resulting loads on the blades and rotor shaft.

(which are key to the unmodified KKT system).

Finding a solution to (35) entails using Newton's method to compute search directions. Given an initial guess at values for (v, γ, λ), it is possible to show that a search direction $(\Delta_v, \Delta_\gamma, \Delta_\lambda)$ can be obtained by solving

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

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

$$\Delta_\lambda = -D^{-1} r_\lambda - D^{-1} \Lambda P \Delta_z. \tag{38}$$

The algorithm then finds new values for the optimization variables (v, γ, λ) 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 $\widehat{H} \triangleq H - P^T D^{-1} \Lambda P$ as in (22a), and then compute the recursions (29) and (31) with the substitutions

$$Q_x(k) = \widehat{Q}_x(k), \tag{39a}$$
$$Q_u(k) = \widehat{Q}_u(k), \tag{39b}$$
$$S_{xu}(k) = \widehat{S}_{xu}(k), \tag{39c}$$
$$g = r_v - P^T D^{-1} r_\lambda, \tag{39d}$$
$$b = -r_\gamma. \tag{39e}$$

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 pertubation 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 τ_{hf} and p_{hf} are high-passed versions of the torque and pitch commands, respectively. The torque command τ_{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 $y_d = [\Omega(\text{w}), \tau_{rated}, 0, 0]^T$.

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 $Q_k = \text{diag}([W_\Omega(k), W_{tqB}(k), W_{tqF}, W_{pF}])$; but the weights $W_\Omega(k)$ and $W_{tqB}(k)$ on speed error and torque vary with sample hit k as shown in Fig. 4(b). In particular, the torque command is only biased towards rated when the turbine nominal torque is at rated, and during deceleration; the rotor-speed tracking error is weighted most heavily where there are abrupt changes in the slope of the desired speed profile.

IV. ILC Results

ILC computes the controls necessary to track each of the three candidate rotor-speed profiles as the wind speed drives the turbine into and out of region 3. A typical progression of the component costs are presented 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_Ω by a factor of four every four iterations, for a total of five increases. In addition, at every 25^{th} 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.

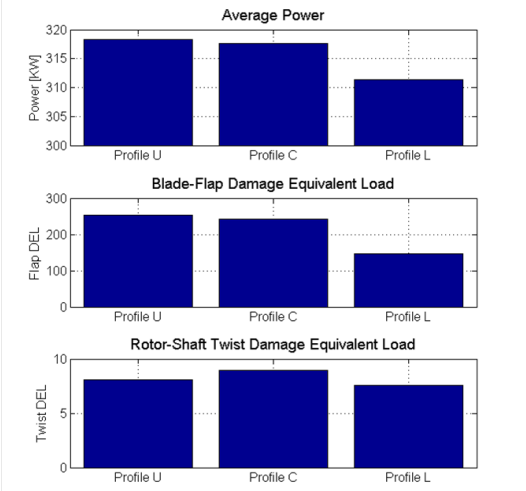


Fig. 6. Performance metrics for the three candidate transition profiles.

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 fair 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

[1] M. Buhl, *WT_Perf User's Guide: Version 3.1*. Golden, CO: National Renewable Energy Laboratory, Dec. 2004.

[2] L. Pao and K. Johnson, "Control of wind turbines," *Control Systems Magazine, IEEE*, vol. 31, no. 2, pp. 44 –62, 2011.

[3] K. Johnson, L. Pao, M. Balas, and L. Fingersh, "Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture," *Control Systems Magazine, IEEE*, vol. 26, no. 3, pp. 70 – 81, June 2006.

[4] L. Pao and K. Johnson, "A tutorial on the dynamics and control of wind turbines and wind farms," in *American Control Conference*, June 2009, pp. 2076 –2089.

[5] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*. Addison-Wesley, 1997.

[6] J. Jonkman and M. L. Buhl, *FAST User's Guide*. Golden, CO: National Renewable Energy Laboratory, 2005.

[7] K. Barton and A. Alleyne, "A norm optimal approach to time-varying ILC with application to a multi-axis robotic testbed," *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 1, pp. 166 –180, 2011.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. New York, NY: Cambridge University Press, March 2004.

[9] H. Okamura, S. Sakai, and I. Susuki, "Cumulative fatigue damage under random loads," *Fatigue and Fracture of Engineering Materials and Structures*, vol. 1, no. 4, pp. 409–419, May 1979.

[10] J. H. Laks, L. Y. Pao, E. Simley, A. Wright, N. Kelley, and B. Jonkman, "Model predictive control using preview measurements from lidar," in *Proc. 49th AIAA/ASME Wind Energy Symposium*, Orlando, FL, Jan. 2011.