2014 American Control Conference (ACC)
June 4-6, 2014. Portland, Oregon, USA

Subspace Predictive Repetitive Control for Wind Turbine Load Alleviation using Trailing Edge Flaps

S.T. Navalkar, J.W. van Wingerden, E. van Solingen, T. Oomen and G.A.M. van Kuik

Abstract— **A novel Subspace Predictive Repetitive Control (SPRC) methodology is presented and used to control trailing edge flaps for wind turbines. First, the dynamics of the wind turbine are identified online. This is especially important for trailing edge flaps on a large wind turbine, where a local change in wind conditions can result in significantly altered aerodynamics. Next, a repetitive control (RC) law is formulated for the multivariable problem from the identified dynamics, that guarantees stability when the identification converges to the true system parameters. This is done in a lower-dimensional basis-function subspace, which reduces computations and gives high control over the shape of the actuator signal. The SPRC methodology is validated in a high-fidelity wind turbine simulation environment.**

I. Introduction

Recent research has shown significant potential to mitigate wind turbine loads, by modifying the flow around wind turbine blades, for instance by using Individual Pitch Control (IPC) [1]. However, IPC requires a drastic increase in the pitch duty cycle and is incapable of high frequency load mitigation. Attention has hence recently been focussed on aerodynamic flow control via active flaps located at the trailing edge of wind turbine blades [2], [3]. While IPC techniques have been applied to trailing edge flap control and show load reduction potential, the periodic nature of wind load signals is considered in formulating the control law. Exploiting the structure in wind turbine loading in a multivariable manner can lead to enhanced performance of flap control.

Iterative Learning Control (ILC) and Repetitive Control (RC) are designed to achieve asymptotic rejection of periodic disturbances and have been widely used since their introduction [4]. Here, the optimal feedforward sequence to suppress periodic loading is learnt using the information obtained during the previous periods or 'iterations'. The design and (robust) stability and performance criteria for learning control is given in [5].

Learning control yields an optimised control input, which can be parameterised using basis functions [6], [18]. This is advantageous since it allows precise control over the smoothness of the actuation signal, improving actuator reliability; a critical issue for rotors instrumented with flaps. Sinusoidal basis functions were used successfully for turbine flap control for a scaled non-pitchable rotor in [3].

Despite learning control being substantially robust to model uncertainty [5], traditional implementations require an (approximate) plant model as a starting point. Wind turbine dynamics change with location and manufacturing errors, hence a model thereof with a good uncertainty description may be difficult to arrive at. Further, flaps modify local flow characteristics and their control authority is closely coupled with local wind condition changes. This motivates a model-free algorithm for flap control. Offline system identification and other data-driven techniques have been used before for ILC [7], [8]. The SPRC (Subspace Predictive Repetitive Control) methodology for online identification and RC implementation was introduced in [17] and applied for wind turbine IPC, and the current paper may be considered an extension thereof.

The key contribution of this paper is flap control using RC in a fully multivariable data-driven manner, in the presence of a traditional collective pitch and torque controller. The SPRC framework from [17] is modified for flap control, and is shown to be desirable as it adapts to changes in local dynamics. Despite the severely reduced control authority of the flaps as compared to IPC, both identification and control has been shown to be possible using SPRC.

The outline of the paper is as follows: in Section I, the need for SPRC is motivated. In Section II, the simulation environment and reference turbine are briefly described. Section III sets up the theoretical framework for SPRC. This is validated in Section IV and the conclusions are discussed in Section V.

II. Wind turbine simulations in GH Bladed™

The SPRC control strategy was tested in GH Bladed™ 4.0, a high-fidelity numerical prototyping environment for wind turbines. This software has been validated using experimental data and used for control strategies for wind turbines with trailing edge flap actuators [10]. Bladed uses a multi-body representation of the wind turbine, with the blades and the tower modelled as flexible bodies. The wind turbine interacts with a full 3D turbulent wind field. The blade element momentum theory [11] is used with wake and dynamic stall corrections to define this interaction.

The turbine used for simulations was the INNWIND 10 MW reference turbine [12], described in Table I. Flaps of length 10 m are considered to be installed on the turbine starting at a station 71 m outboard along the blade length,

S.T. Navalkar, J.W. van Wingerden and E. van Solingen are with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, the Netherlands s.t.navalkar@tudelft.nl

T. Oomen is with the Control Systems Technology Group, Eindhoven University of Technology, 5600 MB Eindhoven, the Netherlands

G.A.M. van Kuik is Professor at the Wind Energy Group, Delft University of Technology, 2629 HS Delft, the Netherlands

This work forms part of the EU INNWIND program, a consortium of 25 partners from the industry and academia, responsible for the development of innovations in large scale wind turbine technology.

TABLE I: Innwind D121 Reference Wind Turbine [12]

Description	Symbol	Value
Rated power	P_{rated}	10 MW
Rotor diameter	d_{ro}	178.3 m
Cut-in wind speed	v_{cutin}	4 m/s
Rated wind speed	v_{rated}	11.4 m/s
Cut-out wind speed	v_{cutout}	25 m/s
Rated rotational rotor speed	Ω_{ro}	9.6 rpm
Gearbox ratio	ν	50.0
Pitch-rate limit	$\dot{\theta}_{\text{limit}}$	10°/s

on all three blades. The flaps are considered to produce a lift increase of 10% for every 3° increase in flap angle. This value was chosen in order to provide adequate realistic control authority for SPRC flap control. A baseline controller for the wind turbine has been designed as per [13], where a basic treatment of the components of the control system of a wind turbine is also given.

This baseline controller for a commercial wind turbine may also address load reduction by adding damping to structural modes. However, the most significant dynamic loading component experienced by the wind turbine occurs at frequencies corresponding to the rotor speed (1P) and its harmonics (2P, 3P, ...). These loads arise due to wind shear, rotational sampling of turbulence and tower shadow, but are not addressed by the baseline controller. So, an RC controller is set up for a period corresponding to 1P in order to address these loads. This is formulated in the next section.

III. Subspace Predictive Repetitive Control

A. Problem Formulation

As motivated in Section I, an ideal controller for flap control would be able to target periodic loads, adapt online to changes in the dynamics and produce smooth actuation signals. This can be achieved by the SPRC framework described in [17], which is adapted for flap control in this section. The aim of the controller is to identify system dynamics from turbine input-output data and implement a learning control law based on the identified dynamics, which yields an optimal feedforward sequence to address periodic turbine loads. The steps required for this controller formulation are outlined below.

B. Step 1: Predictor Definition

In this step, the output of the system is predicted over one period, so that it can be minimised over the control horizon. A wind turbine can be approximated by an LTI (linear, time-invariant) system with periodic disturbances [3]:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ed_k + Ke_k && (1)\\ y_k &= Cx_k + Du_k + Fd_k + e_k && (2) \end{aligned}$$

In the predictor form, this is given by:

$$\begin{aligned} x_{k+1} &= \tilde{A}x_k + \tilde{B}u_k + \tilde{E}d_k + Ky_k && (3)\\ y_k &= Cx_k + Du_k + Fd_k + e_k && (4) \end{aligned}$$

where $x_k \in \mathbb{R}^n$ is the state vector. The control input $u_k \in \mathbb{R}^r$ consists of the three flap deployment angles. The output load vector is $y_k \in \mathbb{R}^\ell$; the blade root out-of-plane moments. The signal $d_k \in \mathbb{R}^m$ represents the periodic component of the loading at the blade root. As stated earlier, the period of the disturbance is taken equal to rotor speed, P. The aperiodic component of loading is given by $e_k \in \mathbb{R}^\ell$. The system matrices A, B, E, $\tilde{A}$, $\tilde{B}$, C, D, K, $\tilde{E}$, F are unknown and have the correct dimensions. The exact size of the state vector, n is unknown; it is, however, assumed to be much smaller than the period of the disturbance, P.

The system matrices are not constant in practice, on account of non-linearities caused due to effects like changes in the local flap angle. For brevity, the dependence of the system matrices on time k and other parameters has not been explicitly stated in the notation.

To eliminate d_k, a periodic difference operator δ is defined:

$$\delta y_k = y_k - y_{k-P}, \qquad \delta d_k = d_k - d_{k-P} = 0$$

It is to be noted that the wind turbine is assumed to be operating at a wind speed above its rated value, where most of the fatigue damage takes place. In this operational region, the rotor speed is held constant by the baseline controller and the value of the period P is hence also almost constant over time. Applying the δ operator to the system (3) - (4), d_k disappears:

$$\begin{aligned} \delta x_{k+1} &= \tilde{A}\delta x_k + \tilde{B}\delta u_k + K\delta y_k && (5)\\ \delta y_k &= C\delta x_k + D\delta u_k + \delta e_k && (6) \end{aligned}$$

For RC control, the output has to be predicted over one period, in order to minimise the predicted error. Such a predictor is given here, but it should be noted that the system matrices herein are unknown. The differenced state is:

$$\delta x_{k+P} = \tilde{A}^P \delta x_k + \begin{bmatrix} \mathscr{K}_u^{(P)}, & \mathscr{K}_y^{(P)} \end{bmatrix} \begin{bmatrix} \delta U_k^{(P)} \\ \delta Y_k^{(P)} \end{bmatrix} \quad (7)$$

where the stacked vectors are defined as:

$$\delta U_k^{(P)} = \begin{bmatrix} u_k - u_{k-P} \\ u_{k+1} - u_{k+1-P} \\ \vdots \\ u_{k+s-1} - u_{k+s-1-P} \end{bmatrix}$$

and a similar vector $\delta Y_k^{(P)}$ for the signal y_k. The extended controllability matrices are defined as:

$$\begin{aligned} \mathscr{K}_u^{(P)} &= \begin{bmatrix} \tilde{A}^{P-1}\tilde{B}, & \tilde{A}^{P-2}\tilde{B}, & \ldots, & \tilde{B} \end{bmatrix} \\ \mathscr{K}_y^{(P)} &= \begin{bmatrix} \tilde{A}^{P-1}K, & \tilde{A}^{P-2}K, & \ldots, & K \end{bmatrix} \end{aligned}$$

It is assumed now that $\tilde{A}^j \approx 0$ for all $j \geq P$. In [14] it is shown that if the system in (3)-(4) is stable, the approximation error is small for a sufficiently large P. Here, it is assumed that the predictor form (3)-(4) is stabilised by the presence of K. Thus:

$$\delta x_{k+P} \approx \begin{bmatrix} \mathscr{K}_u^{(P)}, & \mathscr{K}_y^{(P)} \end{bmatrix} \begin{bmatrix} \delta U_k^{(P)} \\ \delta Y_k^{(P)} \end{bmatrix} \quad (8)$$

For repetitive control, the output is to be predicted over P and hence the output equation is lifted:

$$\delta Y_{k+P}^{(P)} = \tilde{\Gamma}^{(P)} \delta x_{k+P} + \begin{bmatrix} \tilde{H}^{(P)}, & \tilde{G}^{(P)} \end{bmatrix} \begin{bmatrix} \delta U_{k+P}^{(P)} \\ \delta Y_{k+P}^{(P)} \end{bmatrix} \quad (9)$$

As this is a predictor, the white noise sequence δe_k is omitted. Substituting (8) in this equation:

$$\begin{aligned} \delta Y_{k+P}^{(P)} \approx & \begin{bmatrix} \tilde{\Gamma}^{(P)} \mathscr{K}_u^{(P)}, & \tilde{\Gamma}^{(P)} \mathscr{K}_y^{(P)} \end{bmatrix} \begin{bmatrix} \delta U_k^{(P)} \\ \delta Y_k^{(P)} \end{bmatrix} \\ & + \begin{bmatrix} \tilde{H}^{(P)}, & \tilde{G}^{(P)} \end{bmatrix} \begin{bmatrix} \delta U_{k+P}^{(P)} \\ \delta Y_{k+P}^{(P)} \end{bmatrix} \end{aligned} \quad (10)$$

Here, the Toeplitz matrix $\tilde{H}^{(P)}$ is defined as:

$$\tilde{H}^{(P)} = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ C\tilde{B} & D & 0 & \cdots & 0 \\ C\tilde{A}\tilde{B} & C\tilde{B} & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{P-1}\tilde{B} & C\tilde{A}^{P-2}\tilde{B} & C\tilde{A}^{P-3}\tilde{B} & \cdots & 0 \\ 0 & C\tilde{A}^{P-1}\tilde{B} & C\tilde{A}^{P-2}\tilde{B} & \cdots & 0 \\ 0 & 0 & C\tilde{A}^{P-1}\tilde{B} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & D \end{bmatrix}$$

Replacing $\tilde{A}$ by A and $\tilde{B}$ by B, the matrix $H^{(P)}$ can be defined. To obtain $\tilde{G}^{(P)}$ and $G^{(P)}$, $\tilde{B}$ and B are replaced both by K. The extended observability matrix, $\tilde{\Gamma}^{(P)}$ is defined as:

$$(\tilde{\Gamma}^{(P)})^T = \begin{bmatrix} C^T, (C\tilde{A})^T, \ldots, (C\tilde{A}^{P-1})^T, 0^T, \ldots, 0^T \end{bmatrix} \quad (11)$$

The matrix $\Gamma^{(P)}$ can be defined by replacing $\tilde{A}$ by A. The matrices $\tilde{\Gamma}^{(P)} \mathscr{K}_*^{(P)}$ in (10) are given by:

$$\tilde{\Gamma}^{(P)} \mathscr{K}_u^{(P)} \approx \begin{bmatrix} C\tilde{A}^{P-1}\tilde{B} & C\tilde{A}^{P-2}\tilde{B} & \cdots & C\tilde{B} \\ 0 & C\tilde{A}^{P-1}\tilde{B} & \cdots & C\tilde{A}\tilde{B} \\ \vdots & & \ddots & \\ 0 & & & C\tilde{A}^{P-1}\tilde{B} \end{bmatrix}$$

A similar expression can be generated for $\tilde{\Gamma}^{(P)} \mathscr{K}_y^{(P)}$. These matrices, and the Toeplitz matrices above, need to be identified online. From the first block row of (10):

$$\delta y_{k+P} = \begin{bmatrix} C\mathscr{K}_u^{(P)}, & C\mathscr{K}_y^{(P)} \end{bmatrix} \begin{bmatrix} \delta U_k^{(P)} \\ \delta Y_k^{(P)} \end{bmatrix} \quad (12)$$

If input-output data is available, the matrix of coefficients $\begin{bmatrix} C\mathscr{K}_u^{(P)}, & C\mathscr{K}_y^{(P)} \end{bmatrix}$ can be estimated. This matrix contains the information required to construct the $\tilde{\Gamma}^{(P)} \mathscr{K}^{(P)}$ and the Toeplitz matrices. Hence, the regression problem constituted by (12) is to be solved in the next step.

C. Step2: Identification

In this step, the system parameters are identified online, so that the control law can be formulated based on these parameters. The identification problem is: given the input sequence u_k and the output sequence y_k available over a time horizon, estimate recursively the set of Markov parameters Ξ that uniquely define the system (1)-(2), given by:

$$\Xi = \begin{bmatrix} C\mathscr{K}_u^{(P)}, & C\mathscr{K}_y^{(P)} \end{bmatrix} \quad (13)$$

To minimise interference in the regular operation of the turbine, it is desirable to perform this identification in closed-loop, online and recursively.

This regression problem is stated mathematically below. If the differenced output is given by:

$$\delta y_{k+P} = \Xi \begin{bmatrix} \delta U_k^{(P)} \\ \delta Y_k^{(P)} \end{bmatrix} + \delta e_k \quad (14)$$

compute the matrix Ξ, recursively, at every k, as:

$$\Xi = \arg\min_{\Xi} \sum_{k=0}^{\infty} \left\| \delta y_{k+P} - \Xi \begin{bmatrix} \delta U_k^{(P)} \\ \delta Y_k^{(P)} \end{bmatrix} \right\|_2^2 \quad (15)$$

For a unique solution, the matrix $[(\delta U_k^{(P)})^T, (\delta Y_k^{(P)})^T]^T$ should have full rank at all k and hence the control input is required to be persistently exciting of a sufficiently high order. For the recursive implementation, the adaptive forgetting factor technique from [15] is used. At time k, an estimate of the Markov parameters is available:

$$\hat{\Xi}_k = \begin{bmatrix} \widehat{C\tilde{A}^{(P-1)}\tilde{B}} & \cdots & \hat{D} & \widehat{C\tilde{A}^{P-1}K} & \cdots & \widehat{CK} \end{bmatrix}_k \quad (16)$$

By partitioning and rearranging the matrix $\hat{\Xi}_k$ the matrices $(\widehat{\tilde{\Gamma}^{(P)} \mathscr{K}_u^{(P)}})_k$, $(\widehat{\tilde{\Gamma}^{(P)} \mathscr{K}_y^{(P)}})_k$, $\hat{\tilde{H}}_k^{(P)}$ and $\hat{\tilde{G}}_k^{(P)}$ can be created.

The period P is not perfectly constant over time, as has been assumed earlier. In the implementation, the incoming data is resampled such that the stacked vectors always span one full period. This leads to approximation errors during the identification. It is expected that these errors are bounded and small in magnitude due to the action of the speed controller. It is also assumed that the RC control is adequately robust to compensate for the errors.

D. Step3: Infinite Horizon Repetitive Control

In this step, the repetitve control law to minimise the output (error) is formulated based on the system parameters obtained in the last step. Repetitive Control involves the implementation of a feedforward sequence which is updated once every period, based on the error signal over the previous period. For this, the formulation of an ILC law with basis functions [6] is extended to RC. The general RC law has the form:

$$\theta_{j+1} = \alpha \theta_j + \beta \begin{bmatrix} \delta x_j \\ \varepsilon_{j-1} \end{bmatrix}$$

with

$$U_k^{(P)} = U_f \theta_j$$

Here, j is the iteration number (or number of periods elapsed). To control the shape of the actuation signal, the input and output are parameterised by projection onto a subspace spanned by basis function vectors. Here, θ_j is the projected input. The projection matrix is:

$$U_f = \begin{bmatrix} \phi_0, & \phi_1, & \cdots, & \phi_b \end{bmatrix}$$

where each $\phi_i \in \mathbb{R}^r$ is a basis vector and $b \leq r$, and typically b is much smaller than r. For $U_f = I_{r\times r}$ the "traditional" RC controller with a full-dimensional input space is recovered, thus traditional RC can be considered a special case of RC with basis functions.

The change in the input sequence, is a function of the new initial state of the system δx_j and the disturbance rejection error ε_{j-1}. The term $\beta \in \mathbb{R}^{b\times(n+\ell)}$ is the learning gain, and the term α is a Q-filter (a similar ILC law is given in [5]).

A repetitive control law is needed to obtain the input that minimises the predicted vibration output, over an infinite horizon. An infinite horizon ensures closed-loop stability when the true system parameters are identified in the previous step. The predictor from (10) is restated here, using estimated system matrices:

$$\delta Y_{k+P}^{(P)} = \left[(\widehat{\tilde{\Gamma}^{(P)}\mathscr{K}_u^{(P)}})_k, \quad (\widehat{\tilde{\Gamma}^{(P)}\mathscr{K}_y^{(P)}})_k\right]\begin{bmatrix}\delta U_k^{(P)}\\ \delta Y_k^{(P)}\end{bmatrix} + \left[\hat{\tilde{H}}_k^{(P)}, \quad \hat{\tilde{G}}_k^{(P)}\right]\begin{bmatrix}\delta U_{k+P}^{(P)}\\ \delta Y_{k+P}^{(P)}\end{bmatrix} \tag{17}$$

Using the equalities $(I-\hat{\tilde{G}}_k^{(P)})^{-1}\hat{\tilde{H}}_k^{(P)} = \hat{H}_k^{(P)}$ and $(I-\hat{\tilde{G}}_k^{(P)})^{-1}\hat{\tilde{\Gamma}}_k^{(P)} = \hat{\Gamma}_k^{(P)}$, this reduces to:

$$\delta Y_{k+P}^{(P)} = \left[(\widehat{\Gamma^{(P)}\mathscr{K}_u^{(P)}})_k, \quad (\widehat{\Gamma^{(P)}\mathscr{K}_y^{(P)}})_k\right]\begin{bmatrix}\delta U_k^{(P)}\\ \delta Y_k^{(P)}\end{bmatrix} + \hat{H}_k^{(P)}\delta U_{k+P}^{(P)} \tag{18}$$

To penalise the absolute output, $\delta Y_{k+P}^{(P)}$ is expanded:

$$Y_{k+P}^{(P)} = \left[(\widehat{\Gamma^{(P)}\mathscr{K}_u^{(P)}})_k, \quad (\widehat{\Gamma^{(P)}\mathscr{K}_y^{(P)}})_k, \quad I_{P\ell}\right]\begin{bmatrix}\delta U_k^{(P)}\\ \delta Y_k^{(P)}\\ Y_k^{(P)}\end{bmatrix} + \hat{H}_k^{(P)}\delta U_{k+P}^{(P)}$$

This equation is recast into an LQ form such that a state-feedback controller can be synthesised [17], which guarantees stability if the true system parameters are estimated:

$$\underbrace{\begin{bmatrix}Y_k\\ \delta U_k\\ \delta Y_k\end{bmatrix}}_{\mathscr{X}_{j+1}} = \underbrace{\begin{bmatrix}I_{P\ell} & (\Gamma\mathscr{K}_u)_{k-P} & (\Gamma\mathscr{K}_y)_{k-P}\\ 0_{Pr\times P\ell} & 0_{Pr} & 0_{Pr\times P\ell}\\ 0_{P\ell} & (\Gamma\mathscr{K}_u)_{k-P} & (\Gamma\mathscr{K}_y)_{k-P}\end{bmatrix}}_{\mathscr{A}_j}\underbrace{\begin{bmatrix}Y_{k-P}\\ \delta U_{k-P}\\ \delta Y_{k-P}\end{bmatrix}}_{\mathscr{X}_j} + \underbrace{\begin{bmatrix}H_{k-P}\\ I_{Pr}\\ H_{k-P}\end{bmatrix}}_{\mathscr{B}_j}\delta U_k \tag{19}$$

where the stacking superscripts and estimate notations have been dropped for brevity. Also, $\mathscr{A}_j$ and $\mathscr{B}_j$ are taken to be functions of the iteration number j since it is deemed adequate to update them every iteration instead of once every time step. The input and output is now expressed as a linear combination of the basis functions, by the transformations $U_k^{(P)} = U_f\theta_j$ and $\bar{Y}_j = U_f^\dagger Y_k^{(P)}$, where the $\dagger$ symbol indicates the pseudo-inverse of a matrix. Thus, (19) becomes:

$$\underbrace{\begin{bmatrix}\bar{Y}_j\\ \delta\theta_j\\ \delta\bar{Y}_j\end{bmatrix}}_{\bar{\mathscr{X}}_{j+1}} = \underbrace{\begin{bmatrix}I_{b\ell} & U_f^\dagger(\Gamma\mathscr{K}_u)_{k-P}U_f & U_f^\dagger(\Gamma\mathscr{K}_y)_{k-P}U_f\\ 0_{br\times b\ell} & 0_{br} & 0_{br\times b\ell}\\ 0_{b\ell} & U_f^\dagger(\Gamma\mathscr{K}_u)_{k-P}U_f & U_f^\dagger(\Gamma\mathscr{K}_y)_{k-P}U_f\end{bmatrix}}_{\bar{\mathscr{A}}_j} \times \underbrace{\begin{bmatrix}\bar{Y}_{j-1}\\ \delta\theta_{j-1}\\ \delta\bar{Y}_{j-1}\end{bmatrix}}_{\bar{\mathscr{X}}_j} + \underbrace{\begin{bmatrix}U_f^\dagger H_{k-P}U_f\\ I_{br}\\ U_f^\dagger H_{k-P}U_f\end{bmatrix}}_{\bar{\mathscr{B}}_j}\delta\theta_j \tag{20}$$

For the RC law, the weighted norm of $\bar{Y}_k$ over an infinite horizon is to be minimised. Thus, the cost criterion is:

$$J = \sum_{j=0}^{\infty}||(\bar{\mathscr{X}}_{j+1})^T Q_f\bar{\mathscr{X}}_{j+1} + (\delta\theta_j)^T R_f\delta\theta_j||_2^2 \tag{21}$$

where Q_f and R_f are user-defined LQR style weighting matrices for weighting the extended state and input in (20). The SPRC problem is now similar to the LQR problem formulation, the difference being that SPRC involves lifting the system over one period P and projecting the input-output signals onto a basis function subspace. The state feedback gain K_f can be obtained by solving the SPRC problem (21), using this similarity. The discrete algebraic Riccati equation (DARE) is solved online to obtain the optimal state feedback gain. This is done by assuming an initial estimate of $P_{R,j}$, the positive definite solution to the DARE obtained online, and then updating it using the DARE iteratively:

$$P_{R,j+1} = Q_f + \bar{\mathscr{A}}_j^T(P_{R,j} - P_{R,j}\bar{\mathscr{B}}_j^T(R_f + \bar{\mathscr{B}}_j^T P_{R,j}\bar{\mathscr{B}}_j)^{-1}\bar{\mathscr{B}}_j^T P_{R,j})\bar{\mathscr{A}}_j$$

and the state feedback gain $K_{f,j}$ is then:

$$K_{f,j} = (R_f + \bar{\mathscr{B}}_j^T P_{R,j}\bar{\mathscr{B}}_j)^{-1}\bar{\mathscr{B}}_j^T P_{R,j}\bar{\mathscr{A}}_j$$

With this control law, the control input over the next period is given by:

$$\delta U_{k+P}^{(P)} = U_f K_{f,j} U_f^\dagger\begin{bmatrix}Y_k^{(P)}\\ \delta U_k^{(P)}\\ \delta Y_k^{(P)}\end{bmatrix} \tag{22}$$

The implementation with sinusoidal basis functions is shown in Fig. 1. In place of a fixed U_f matrix, the output is synthesised online using the rotor azimuth, making the method robust to speed variations. A convergence proof for this adaptive law is not given, however if the estimation error in the identification is small and the parameters vary slowly, the LQ formulation should yield a stabilising controller. With different wind fields and initial estimates, the method shows good convergence properties in simulations.

Thus, an adaptive control law to reduce periodic loads using wind turbine flaps has been formulated, and will be validated in the next section.

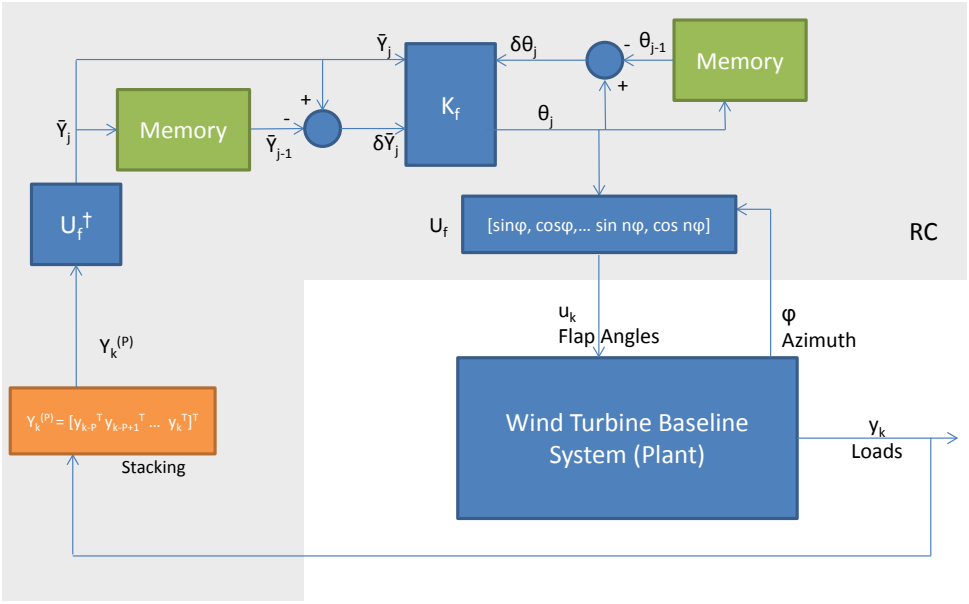

Fig. 1: RC implementation, corresponding to Eq. (22)

IV. Simulation Study

The simulation environment and wind turbine used for validating SPRC are described in Section II. A baseline controller is designed for this turbine for speed control, using the paradigm followed for commercial wind turbine controllers. The SPRC controller for flap control, as described above, is then used with this closed-loop baseline controlled turbine.

A mean wind speed of 18 m/s is chosen for the simulations, as this corresponds to relatively high turbine loads and a relatively high probability of occurrence at a typical wind turbine site. A zero-turbulence case is considered, in order to understand the behaviour of the controller in an idealised scenario. The turbine simulation is run for 800 seconds.

For wind turbine loading, most of the energy is concentrated at 1P and 2P [1], where 1P refers to the rotor speed, and 2P is twice the rotor speed. Hence, the basis functions are chosen to be sinusoidal, with frequencies 1P and 2P:

$$U_f = \begin{bmatrix} \sin(2\pi/P), & \cos(2\pi/P), & \sin(4\pi/P), & \cos(4\pi/P) \\ \sin(4\pi/P), & \cos(4\pi/P), & \sin(8\pi/P), & \cos(8\pi/P) \\ \vdots, & \vdots, & \vdots, & \vdots \\ \sin(2\pi), & \cos(2\pi), & \sin(4\pi), & \cos(4\pi) \end{bmatrix}$$

It is seen that the identification algorithm is able to arrive at a close approximation of the Markov parameters in 120 seconds, or 19 iterations. This can be seen in Figure 2, where the quality of the estimate is measured in terms of variance accounted for (vaf) [9]. After the identification exceeds 90% vaf, the persistency of excitation is artificially switched off; hence the vaf does not improve further. An RC law is calculated online from the identified Markov parameters and it converges quickly to the required optimal feedforward sequence. Within 180 seconds or 28 iterations, the learning law converges to a stable feedforward flap signal that reduces

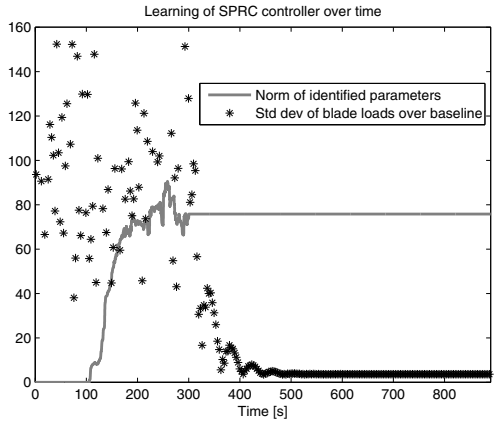

Fig. 2: Identification algorithm convergence and repetitive control law convergence over time: wind speed 18 m/s and 0% turbulence

the blade dynamic loading to 20% of its baseline value in this idealised case.

For more realistic results, the model is tested with an SPRC controller for a wind field of mean wind speed 18 m/s and turbulence intensity 3.75%. This cases has increased variations in the generator rpm, but SPRC is expected to be robust to these variations. The reduction in the blade root out-of-plane loads can be seen in Figure 3; the load is reduced by 12.4%. Since 1P and 2P sinusoidal basis functions are used, dynamic load reduction is obtained at these frequencies. Further, similiar to the results of IPC [1], flap control also has a collateral effect of load reduction on stationary components, as evidenced by the tower side-side loading results in Figure 4. The mean of the tower loading reduces by 6.3%.

The flap actuation signal is shown in Figure 5. It can be

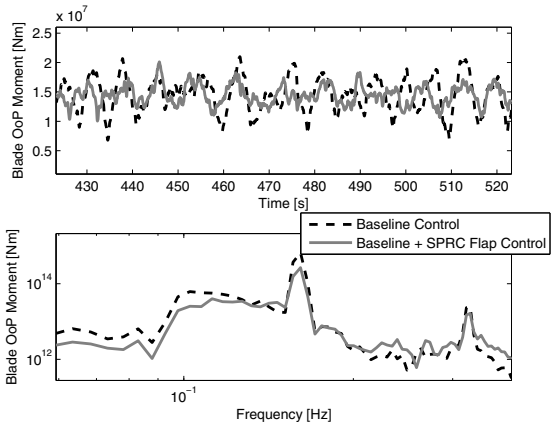

Fig. 3: SPRC achieves blade load reductions at 1P and 2P frequencies: wind speed 18 m/s and 3.75% turbulence

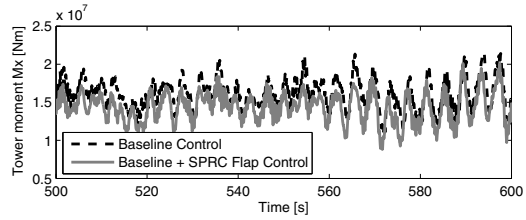

Fig. 4: SPRC achieves mean tower side-side load reduction: wind speed 18 m/s and 3.75% turbulence

seen that it has frequency content only at the 1P and 2P frequencies, as dictated by the sinusoidal basis functions.

Finally, there is a slight increase in the pitch duty cycle as compared to the baseline. This is because the flap action tends to raise rotational speed variations, which are counteracted by collective pitch action. However, the pitch demand is still an order of magnitude lower than that demanded by IPC.

V. Conclusions and Future Work

The SPRC paradigm presented in this paper combines many of the desirable qualities for a wind turbine controller for trailing-edge flaps and other non-traditional actuators. It is seen that the adaptive control law is able to identify system parameters online, from zero a priori knowledge of the system. Further, a feedforward sequence is learned by the controller to alleviate periodic loading of the wind turbine, which would be optimal for the case of perfect system identification. Finally, the actuation signals are precisely controlled in terms of frequency content using basis functions to enhance durability of the flap actuator. The inherently multivariable nature of SPRC implies that IPC can also be included in the algorithm, in order to obtain the best trade-off between flap and pitch control. Further, it is also possible to add hard constraints to the optimisation problem, an important characteristic for control of non-traditional actuators. Future work is expected to consist of these extensions, and their validation in a full turbulent environment.

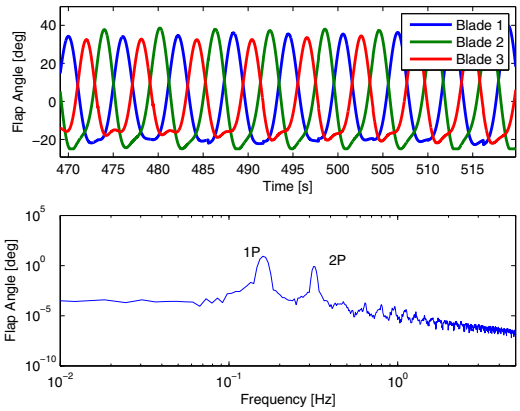

Fig. 5: SPRC flap actuation signal, for wind speed 18 m/s, 3.75% turbulence

References

[1] E. A. Bossanyi. Individual blade pitch control for load reduction. Wind Energy 2003, 6:119-128.

[2] P.B. Andersen, M. Gaunaa, C. Bak, T. Buhl. Load alleviation on wind turbine blades using variable aerofoil geometry. Proceedings of the European Wind Energy Association Conference, Brussels, Belgium, 2006.

[3] J.W. van Wingerden, A.W. Hulskamp, T.K. Barlas, I. Houtzager, H.E.N. Bersee, G.A.M. van Kuik, M. Verhaegen. Two-degree-of-freedom active vibration control of a prototyped 'smart' rotor. Journal of IEEE Transactions on Control System Technology 2011, 19:284-296.

[4] S. Arimoto, S. Kawamura, F. Miyazaki. Bettering operation of robots by learning. Journal of Robotic Systems 1984, 1:123-140.

[5] D.A. Bristow, M. Tharayil, A.G. Alleyne. A survey of iterative learning control. IEEE Control Systems Magazine, 2006, 96-114.

[6] J. van de Wijdeven, O.H. Bosgra. Using basis functions in iterative learning control: analysis and design theory. International Journal of Control 2010, 83:661-675.

[7] J.A. Frueh, M.Q. Phan. Linear quadratic optimal learning control (LQL). International Journal of Control 2000, 10:832:839.

[8] Y. Ye, D. Wang. DCT basis function learning control. IEEE/ASME Transactions on Mechatronics 2005; 10:449-454.

[9] M. Verhaegen and V. Verdult. Filtering and System Identification: An introduction. Cambridge University Press, 2007.

[10] A.L. Lackner and G.A.M. van Kuik. A comparison of wind turbine smart rotor control approaches using trailing edge flaps and individual pitch control. Wind Energy 2010, 13:117-134.

[11] J.F. Manwell, J.G. McGowan, A.L. Rogers. Wind energy explained: Theory, design and application. John Wiley and Sons Ltd., Chichester, 2002.

[12] C. Bak, F. Zahle, R. Bitsche, T. Kim, A. Yde, L.C. Henriksen, A. Natarajan, M. Hansen, Description of the DTU 10 MW Reference Wind Turbine 2013, DTU Wind Energy Report-I-0092.

[13] E.A. Bossanyi. The design of closed loop controllers for wind turbines. Wind Energy 2000, 3:149-163.

[14] A. Chiuso. The role of vector auto regressive modelling in predictor based subspace identification. Automatica 2007, 43:1034:1048.

[15] R. Kulhavy. Restricted exponential forgetting in real-time identification. Automatica 1987, 23:589-600.

[16] B.G. Dijkstra, O.H. Bosgra. Extrapolation of optimal lifted system ILC solution, with application to wafer stage. Proceedings of the American Control Conference, Anchorage, USA, 2002.

[17] S.T. Navalkar, J.W. van Wingerden, E. van Solingen, T. Oomen, E. Pasterkamp, G.A.M van Kuik. Subspace Predictive Repetitive Control to mitigate periodic loads on large scale wind turbines. Mechatronics, 2014, http://dx.doi.org/10.1016/j.mechatronics.2014.01.005.

[18] Y. Shi, R. W. Longman, M. Nagashima. Small gain stability theory for matched basis function repetitive control. Acta Astronautica 2014, 95:260-271.