2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

A novel online controller redesign approach to fault accommodation in Wind Turbine Systems

Tushar Jain[1], Joseph J. Yamé[2], Dominique Sauter[2], and Samir Aberkane[2]

***Abstract*— This paper presents a real-time mechanism to tolerate faults occurring in a Wind Turbine (WT) system. This system is a FAST coded simulator designed by the U.S. National Renewable Energy Laboratory. The demonstrated mechanism lies under the taxonomy of Active Fault-tolerant Control (FTC) systems, namely the online redesign based approach. In the proposed approach, we do not use any *a priori* information about the model of the turbine in real-time. In fact, we use online measurements generated by the WT. Based on the given control specifications, and the observed measurement an occurred fault is accommodated by redesigning the controller online such that the WT generates rated power even under faulty conditions. Secondly, no explicit fault diagnosis (FD) module is used in this approach. As a result, issues of model uncertainty, false alarms, etc. associated with an integrated FD and controller reconfiguration approach to FTC systems are not experienced here.**

I. INTRODUCTION

Renewable sources of energy are always considered to be on the priority level of energy policies. Seeing to an enormous increase in world population, and to a large-scale utilization of polluting sources of energy, wind power is recognized as one of the valuable non-polluting renewable sources of energy. In view of this, today many Variable Speed Wind Turbines (VSWT) are installed offshore, which contribute to a larger part of world's power production. Discovering the advantages of VSWT of such a large scale, it has some disadvantages as well from the industrial point of view, which includes their high manufacturing and maintenance costs.

To deal with unpredictable weather or gravely environmental conditions associated with natural wind supply, the prime goal is to maximize the power production from WT systems. A special attention during the last decade has been given to fulfilling this goal that led to the development of robust and reliable control systems for WT. Such control schemes are designed to deliver the performance specifications within a limited operating range of WT. It becomes a point of major concern whenever this operating range of the WT is violated. Serious sources of this violation are the appearance or occurrence of unknown malfunctions, termed as *faults* that may appear in actuators, sensors or other system components. No doubt, an effectively designed conventional control scheme may result in an unsatisfactory performance, or even instability in the event of faults. The real-time control of systems under such scenarios is known as Active Fault-Tolerant Control (AFTC). Generally, an AFTC system is composed of two cascaded working modules, namely Fault Diagnosis (FD), and Fault Accommodation (FA) or Controller Reconfiguration (CR) [1]. The former module is used to detect, isolate, and identify faults, while the latter module reconfigures controller based on a precisely obtained FD information so that the system can still deliver the specified performance. One of the biggest challenges in this cascaded structure is to handle effectively the model uncertainties appearing during FD operation, which can lead to *false alarms*. In addition, a strong dynamical interaction between the FD module and the FA module, which also imposes some difficulties considering the real-time constraints [4].

In recent years, a significant amount of work has been done that deal mainly in diagnosing a fault, see [10] and the references therein. A little work in controlling a WT system is addressed in the literature [9]. The worth noting point is that neither FD nor FTC is much demonstrated on the benchmark model considered in this paper apart from the work reported by the authors in [6]. We use the benchmark model of WT [9], where the integrated FAST-coded block is designed by the U.S. National Renewable Energy Laboratory's (NREL) National Wind Turbine Center.

Stating precisely, apart from the aforementioned contribution within the paper, the other contribution is to present a measurement based solution to deal with a problem of controlling the system subjected to unknown faults, where no mathematical model of the WT is used in real-time. In addition, any explicit FD module that extracts in online the information about the system to diagnose a fault is not used. Instead, whenever a fault occurs in the system, the controller is reconfigured online solely based on the real-time measurements generated by the NREL's WT guaranteeing the fault accommodation. To present this novel AFTC technique, we use the mathematical framework of behavioral system theory [11], in which the interconnection between two dynamical systems via system variables plays the central role.

II. BENCHMARK MODEL DESCRIPTION

The driving force in a wind turbine system is unregulated variable wind speed. The basic functionality includes the two-step energy conversion. In the first step, the wind energy is converted into the mechanical energy that later converted into the electrical energy.

[1]T. Jain is with Laboratory of Process Control and Automation, School of Chemical Technology, Aalto University, Kemistintie 1, FI-00076 Aalto, Finland. tushar.jain at aalto.fi

J. J. Yamé, D. Sauter, and S. Aberkane are with Centre de Recherche en Automatique de Nancy, CRAN-CNRS, UMR 7039, Université de Lorraine, Campus Sciences, 54506 Vandoeuvre-lès-Nancy, France. firstname.lastname at univ-lorraine.fr

978-3-952-41734-8/©2013 EUCA

A. Dynamics of the system

The system is composed of various sub-systems, namely Blade & Pitch System (BPS), Drive Train (DT), and Generator & Converter (GC). The BPS model is the combination of an aerodynamic model, and a pitch model. The latter model is treated as an actuator within the system and will be discussed later together with other actuators. The aerodynamic properties of the wind turbine are affected by the pitch angles of the blades $\beta(t)$, the speed of the rotor $\omega_r(t)$, and the wind speed $v_w(t)$. This aerodynamic torque T_a is applied to the rotor and is expressed by

$$T_a(t) = \frac{\rho \pi R^3 C_q(\lambda(t), \beta(t)) v_w(t)^2}{2}, \tag{1}$$

where $\lambda(t) = \frac{\omega_r(t)R}{v_w(t)}$ is the tip-speed ratio with ρ, R, and C_q denoting the air density, radius of the rotor, and torque coefficient, respectively.

B. Actuator Model

In this benchmark WT model, three actuators for the pitch, generator, and yaw systems are modeled and implemented externally, i.e. apart from the embedded FAST Simulink code. In this paper, we mainly concentrate on the other two actuators but the yaw system. Actually, the yaw actuator model and the associated yaw controller, which is conceived as an overall yaw mechanism is used to orient the wind-turbine upright to the wind direction. The FAST implementing non-linear WT model requires a yaw angular velocity and yaw angular position as one of the inputs. At all time, it is assumed that a yawing system exists, which keep the wind direction perpendicular to the rotor plane.

The hydraulic pitch system consists of three identical pitch actuators, which is modeled as a differential equation between the pitch angle β and its reference β_r. In principle, it is a piston servo-system which can be expressed by a second-order differential system [9]:

$$\frac{d^2}{dt^2}\beta(t) + 2\zeta\omega_n \frac{d}{dt}\beta(t) + \omega_n^2 \beta(t) - \omega_n^2 \beta_r(t) = 0, \tag{2}$$

where ζ and ω_n denotes the damping factor and the natural frequency, respectively. The differential equation (2) is associated with each of the three pitch actuators.

In the GC system, the converter loads the generator producing the electric power with a certain torque. The dynamics of the converter can be approximated by a first-order differential system [9], which is given by

$$\frac{d}{dt}T_g(t) + \alpha_{gc} T_g(t) - \alpha_{gc} T_{g,r}(t) = 0, \tag{3}$$

where T_g and $T_{g,r}$ represents the generated torque and the reference generated torque, respectively, with a constant model parameter $\alpha_{gc} = 50$. The power produced by the generator P_g is given by $P_g(t) = \eta_g \omega_g(t) T_g(t)$, where η_g and ω_g denotes the efficiency of the generator and the generator speed, respectively.

C. Fault Scenario

Various types of faults are addressed in the literature [9]. However, in this paper, we consider a fault that causes an abrupt power drop in the hydraulic pressure. This power drop fault affects the dynamics of the pitch system by changing the parameters, ζ and ω_n from their nominal or healthy-mode values ζ_n and $\omega_{n,n}$ to their values in faulty-mode ζ_f and $\omega_{n,f}$. The faulty dynamics of the pitch system can be described by the following second-order differential system

$$\frac{d^2}{dt^2}\beta(t) + 2\zeta(\Theta_f(t))\omega_n(\Theta_f(t))\frac{d}{dt}\beta(t) + \omega_n^2(\Theta_f(t))\beta(t) - \omega_n^2(\Theta_f(t))\beta_r(t) = 0, \tag{4}$$

where

$$\omega_n^2(\Theta_f(t)) = (1-\Theta_f(t))\omega_{n,n}^2 + \Theta_f(t)\omega_{n,f}^2$$
$$2\zeta(\Theta_f(t))\omega_n(\Theta_f(t)) = 2(1-\Theta_f(t))\zeta_n\omega_{n,n} + 2\Theta_f(t)\zeta_f\omega_{n,f}$$

with $\Theta_f(t) \in [0,1]$ representing the various operating modes of the WT.

D. Fault-Tolerant Control Objectives

The wind turbine principally operates inside four regions or control-zones depending on the speed of wind $v_w(t)$, namely startup zone, partial load zone, full load zone, and cut-off zone. In this paper, we are mainly interested in operating the WT in the full load region where $v_w(t) \in [11.4, 25] m/s$. Particularly, we consider the wind speed at a mean value of 17 m/s, where the control objective is to design controllers such that the generated power $P_g(t)$ can track the rated power P_{rated} around its mean value of 5 MW. Nevertheless, under the fault occurrence, satisfying the above requirement can no longer be guaranteed. Consequently, the fault-tolerant control objective is to design a real-time controller reconfiguration mechanism such that the aforementioned control requirement on the WT can be fulfilled. In addition, suppressing large transients during accommodating an occurring fault is another requirement from the practical implementation point of view.

III. Real-time Fault Accommodation

We view a dynamical system as an exclusion law that indicates which trajectories are admissible for the system. A trajectory is a vector-valued function $s : \mathbb{T} \to \mathbb{S}, t \mapsto s(t)$ that take its values in the signal space $\mathbb{S}$ where $\mathbb{T} \subseteq \mathbb{R}$ is the time axis, $\mathbb{S} \subseteq \mathbb{R}^{\mathtt{s}}$ with $\mathtt{s}$ denoting the dimension of $s(t)$. The behavior of such systems can be expressed by the set of solutions of a system of linear, constant-coefficient differential equations. The system is defined by a linear differential equation

$$R_0 s + R_1 \frac{d}{dt}s + R_2 \frac{d^2}{dt^2}s + \ldots + R_{\mathtt{n}} \frac{d^{\mathtt{n}}}{dt^{\mathtt{n}}}s = 0, \tag{5}$$

where $R_i, i = 0, 1, 2, \ldots, \mathtt{n}$ are real constant matrices belonging to $\mathbb{R}^{\bullet \times \mathtt{s}}$ with finite number of rows and $\mathtt{s}$ columns. The set of all linear differential systems with $\mathtt{s}$ variables will be denoted by $\mathfrak{L}^{\mathtt{s}}$. Equation (5) can compactly be written as $R\left(\frac{d}{dt}\right)s = 0, \quad R(\xi) \in \mathbb{R}^{\bullet\times \mathtt{s}}[\xi]$, with $R(\xi) = R_0 + R_1\xi + R_2\xi^2 + \ldots + R_{\mathtt{n}}\xi^{\mathtt{n}}$ where $\mathbb{R}^{\bullet\times \mathtt{s}}[\xi]$ denotes the set of $\bullet \times \mathtt{s}$ polynomial matrices with real coefficients and indeterminate ξ. Then the behavior $\mathcal{B}$ is given by the set $\mathcal{B} = \{s \in (\mathbb{R}^{\mathtt{s}})^{\mathbb{R}} | s \text{ satisfies (5)}\}$. The representation in (5) is called the kernel representation of $\mathcal{B}$, and we often write it as $\mathcal{B} = \mathtt{ker}(R(\frac{d}{dt}))$. From the above, clearly a dynamical system is now represented by a set of operating signals.

A. Feedback Interconnection of Dynamical Systems

The concept of interconnection plays the central role in modeling and control of system within the behavioral framework. By an interconnected system, we mean a system that consists of interacting subsystems. Here, we deal within the special type of interconnection, termed as the feedback interconnection [3]. Let $\mathcal{P} \in \mathfrak{L}^{\mathtt{w}+\mathtt{c}}$ denotes the full behavior of the plant and $\mathcal{C} \in \mathfrak{L}^{\mathtt{c}}$ denotes the behavior of the controller, where $s = \mathtt{col}(w,c)$ with $w = \mathtt{col}(r,y)$ and $c = \mathtt{col}(e,u)$, whose values lies in the signal space $\mathbb{S}$ having the dimension $\mathtt{s} = \mathtt{r}+\mathtt{y}+\mathtt{e}+\mathtt{u}$. Acquiring the behavioral point of view, we can now define the trajectory-based dynamical system for the plant and the controller by $\Sigma_{\mathcal{P}} = (\mathbb{T},\mathbb{S},\mathcal{P})$, and $\Sigma_{\mathcal{C}} = (\mathbb{T},\mathbb{S},\mathcal{C})$ respectively, where $\mathbb{T} \subseteq \mathbb{R}$, $\mathbb{S} \subseteq \mathbb{R}^{\mathtt{r}+\mathtt{y}+\mathtt{e}+\mathtt{u}}$, $\mathcal{P} \subseteq \mathbb{S}^{\mathbb{T}}$, and their behaviors in the following way.

$$\mathcal{P} = \left\{ s = \mathtt{col}(w,c) \in \mathbb{S}^{\mathbb{T}} \middle| \begin{bmatrix} R\left(\frac{d}{dt}\right) & -M\left(\frac{d}{dt}\right) \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0 \right\}, \tag{6}$$

$$\text{where} \quad R(\xi) = \begin{bmatrix} I_{\mathtt{r}} & -I_{\mathtt{y}} \\ 0_{\mathtt{r}} & D_p(\xi) \end{bmatrix}, \quad M(\xi) = \begin{bmatrix} I_{\mathtt{e}} & 0_{\mathtt{u}} \\ 0_{\mathtt{e}} & N_p(\xi) \end{bmatrix}$$

with $D_p(\xi) \in \mathbb{R}^{\bullet\times \mathtt{y}}[\xi], N_p(\xi) \in \mathbb{R}^{\bullet\times \mathtt{u}}[\xi]$ being co-prime polynomials, and $0_{\bullet}$, and $I_{\bullet}$ representing the zero matrix, and the identity matrix of suitable dimension. From the input/output point of view, y is considered as the output of the plant and u as the input. With this partition of inputs and outputs, together with [11, Definition 3.3.1], evidently $D_p(\xi)^{-1}N_p(\xi) = G(\xi)$ defines a proper rational matrix with $D_p(\xi) \neq 0$. In a similar way, the behavior of the controller $\Sigma_{\mathcal{C}}$ is given by

$$\mathcal{C} = \left\{ c \in \mathbb{S}^{\mathbb{T}} \middle| C\left(\frac{d}{dt}\right) c = 0 \right\}, \tag{7}$$

$$\text{where} \quad C(\xi) = \begin{bmatrix} N_c(\xi) & -D_c(\xi) \end{bmatrix}$$

with $D_c(\xi) \in \mathbb{R}^{\bullet\times \mathtt{u}}[\xi]$, $N_c(\xi) \in \mathbb{R}^{\bullet\times \mathtt{y}}[\xi]$ being co-prime polynomials, and $D_c(\xi)^{-1}N_c(\xi) = H(\xi)$ representing a proper rational matrix with $D_c(\xi) \neq 0$. In this controller configuration, u is the output of the controller, and e is the input. Whenever the above two system interconnects, the controller imposes some restrictions on the behavior of the plant. The imposed restrictions on $\mathcal{P}$ by $\mathcal{C}$ yields the full controlled behavior, which is given by

$$\mathcal{K}_{\mathtt{full}} = \{ s = \mathtt{col}(w,c) \,|(w,c) \in \mathcal{P} \text{ and } c \in \mathcal{C} \}. \tag{8}$$

Generally, the interest lies in controlling the behavior of the manifest variables in the controlled system. The controlled behavior in terms of the manifest variables in the full interconnected system, defined in (8), can be obtained by using the elimination theorem [11, Theorem 6.2.6], which is then given by

$$\mathcal{K} = \{ w \in \mathbb{S}^{\mathbb{T}} | \exists c \in \mathcal{C} \text{ such that } (w,c) \in \mathcal{P} \}. \tag{9}$$

B. Effects of fault

The real-time notion of controlling a faulty system is that the operating plant must achieve the control objectives at anytime, i.e. regardless of any occurrence of a fault. In this respect, we can single out a subset of plants' behavior as desirable. We call it the desired behavior, denoted by $\mathcal{D}$, which is provided by an effective Failure Mode and Effective Analysis (FMEA). Indeed, this analysis procedure is a mandatory prerequisite to study fail-safe systems [1]. FMEA's objective is to forecast systematically how fault effects on elements relate to faults at inputs, or outputs within the elements, and what reactions should be imposed on the system whenever certain faults appear. We termed this phase as the Analysis & Development (AD) phase that aims at providing a complete coverage of possible occurring faults into the system as well as the achievable or *implementable* desired behavior. An approach to perform this analysis procedure is presented in [8]. The desired behavior $\mathcal{D}$ will, indeed, be defined in terms of the manifest variables, which is given by

$$\mathcal{D} = \left\{ w \in \mathbb{S}^{\mathbb{T}} | D\left(\frac{d}{dt}\right) w = 0 \right\}, \tag{10}$$

$$\text{where } D(\xi) = \begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix}.$$

with $D_r(\xi) \in \mathbb{R}^{\bullet\times \mathtt{r}}[\xi], D_y(\xi) \in \mathbb{R}^{\bullet\times \mathtt{y}}[\xi]$ as the co-prime polynomials, and $D_y(\xi)^{-1}D_r(\xi)$ representing a set of proper rational matrices with $D_y(\xi) \neq 0$.

The International Federation of Automatic Control (IFAC) SAFEPROCESS Technical Committee defines a fault as an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition. Faults affect the dynamics of the system in a way that the control specifications are not satisfied. With the above consideration, we define a fault as

Definition 1 (Occurring Faults [3]): A fault is said to be occurring in the system whenever $\mathcal{K} \nsubseteq \mathcal{D}$.

The real-time FTC problem we are dealing with can now be posed in the following way. Given a vector space of time dependent signals $\mathbb{S}^{\mathbb{T}}$, and the desired behavior $\mathcal{D}$, the problem is to "synthesize" an appropriate controller $\mathcal{C}$, without using any mathematical model of the plant in real-time, which have the suitable control actions such that the controlled behavior $\mathcal{K}$ satisfy the desired behavior $\mathcal{D}$ at anytime.

C. Design and Implementation of Real-time FTC mechanism

The implementability of the desired behavior plays a key role in an online design of the controller. Otherwise, if the desired behavior is not achievable or implementable, then no controller exists that can guarantee the fault tolerance in the sense that a faulty system satisfies the specified performance as it was satisfied in the fault-free operating mode. Roughly speaking, the faults for which the desired behavior is not implementable can be termed as "intolerable faults". To support the implementability of $\mathcal{D}$, we posit the "Willems' Theorem" [13].

Theorem 1 (Willems' Theorem): Let $\mathcal{P}$ be a behavior of the plant, and let $\mathcal{D}$ be a desired behavior. Then the following statements are equivalent:

1) $\mathcal{D}$ is achievable or implementable w.r.t. the plant.
2) There exists a controller $\mathcal{C}$ that implements $\mathcal{D}$.
3) $\mathcal{N} \subseteq \mathcal{D} \subseteq \mathcal{P}_w$, where $\mathcal{P}_w$ is the manifest plant behavior and $\mathcal{N}$ is the hidden plant behavior, defined by

$$\mathcal{N} = \{w \in \mathbb{S}^{\mathbb{T}} | (w, 0) \in \mathcal{P}\},$$
$$\mathcal{P}_w = \{w \in \mathbb{S}^{\mathbb{T}} | \exists c \in (\mathbb{R}^{\mathtt{c}})^{\mathbb{R}} \text{ such that } (w, c) \in \mathcal{P}\}.$$

Based on the above implementability theorem, van der Schaft [12] gives a "general behavioral description" of the existing controller, irrespective of any particular control configuration, that can implement the desired behavior.

Theorem 2: [12] Let $\mathcal{P}$ be a behavior of the plant, and let $\mathcal{D}$ be the implementable desired behavior. Then the controller, defined as $\mathcal{C} = \{c \in (\mathbb{R}^{\mathtt{c}})^{\mathbb{R}} | \exists \tilde{w} \text{ such that } (\tilde{w}, c) \in \mathcal{P} \text{ and } \tilde{w} \in \mathcal{D}\}$, implements the desired behavior $\mathcal{D}$.

The controller defined in Theorem 2 is termed as the canonical controller. Basically, this controller is constructed by the interconnection of the plant (with reversed terminal) and the desired behavior. For determining the kernel representation of the above controller $\mathcal{C}$, we will now use the implementability theorem. From the first inclusion of Theorem 1, i.e. $\mathcal{N} \subseteq \mathcal{D}$, there exists a polynomial matrix, say $L(\xi)$ such that

$$D(\xi) = L(\xi)R(\xi) \tag{11}$$

where $\mathcal{D} = \mathtt{ker}(D(\xi))$, and $\mathcal{N} = \mathtt{ker}(R(\xi))$. The full behavior of the plant is given by the following kernel representation $R(\xi)w = M(\xi)c$. Pre-multiplying the last differential equation by $L(\xi)$, we get $L(\xi)R(\xi)w = L(\xi)M(\xi)c$. From the above, it follows that $D(\xi)w = L(\xi)R(\xi)w = 0$. This yields the kernel representation of the canonical controller, which is given by

$$\mathcal{C} \equiv L(\xi)M(\xi)c = 0. \tag{12}$$

From the above, clearly the controller is constructed for general systems without imposing any *realizability* requirements. This issue is of utmost practical importance for a possible implementation of the controller in the closed-loop. Theoretically, in [7, Theorem 16], the so-called regularity of interconnection is imposed for the design of the canonical controller. By construction, the control configuration considered in this paper guarantees that whatever be the controller, it will always make a regular interconnection with the plant. In addition, giving a closer look to the kernel representation of the controller given in (12), it includes the plant's embedded knowledge within the $M(\xi)$ matrix, which has to be available in real-time while synthesizing an online controller. As we mentioned before, we do not have any mathematical model of the plant in real-time, i.e. $R(\xi)$ and $M(\xi)$ matrices are not available during the controller reconfiguration process, therefore, we cannot use the above equation to compute the controller's polynomials.

The main result of this section is given in the following proposition where we directly compute the controller polynomials using the real-time measurements observed from the plant. First, we define the "filtered" plant signals, denoted by $(\bar{u}, \bar{y})$, which are given as

$$\bar{u} = D_r(\xi)u, \quad \bar{y} = (D_r(\xi) - D_y(\xi))y, \tag{13}$$

together with polynomials $D_r(\xi), D_y(\xi)$ considering to take the form as $D_r(\xi) = d_r(\xi)I_{\mathtt{r}}, D_y(\xi) = d_y(\xi)I_{\mathtt{y}}$, where $d_r(\xi) \in \mathbb{R}^{1\times 1}[\xi], d_y(\xi) \in \mathbb{R}^{1\times 1}[\xi]$.

Proposition 1: Given a vector space of time-dependent signals $(\mathbb{T} \times \mathbb{S})$, the implementable desired behavior $\mathcal{D}$, for any closed-loop controller $\mathcal{C}$ if an unknown fault occurs into the system then the following statements are equivalent:

1) The system is a real-time fault-tolerant control system.
2) The trajectories $(\bar{u}, \bar{y})$ belongs to the controller $\mathcal{C}$, which is equivalent to saying that the following differential equation holds.

$$N_c(\xi)\bar{y} + D_c(\xi)\bar{u} = 0. \tag{14}$$

Proof: (1) $\Longrightarrow$ (2): Since the desired behavior $\mathcal{D}$ is implementable, from Theorem 1 it follows that there exists a controller $\mathcal{C}$ that implements $\mathcal{D}$. Therefore, we now only required to synthesize the kernel representation of that controller $\mathcal{C}$ without using any *a priori* information of the plant's model to guarantee the fault tolerance. For the considered feedback configuration, substitute the explicit kernel representation of $\mathcal{D}$ and $\mathcal{N}$ into (11), which gives

$$\begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix} = L(\xi) \begin{bmatrix} I_{\mathtt{r}} & -I_{\mathtt{y}} \\ 0_{\mathtt{r}} & D_p(\xi) \end{bmatrix} \tag{15}$$

In the following discussion, the dimension of the identity and zero matrix will be avoided whenever it is clear from context. The matrix $R(\xi)$ can be factored as

$$\begin{bmatrix} I & -I \\ 0 & D_p(\xi) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & D_p(\xi) \end{bmatrix} \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}. \tag{16}$$

Putting (16) in (15), we get

$$\begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix} = L(\xi) \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & D_p(\xi) \end{bmatrix} \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}.$$

Since the matrix $D_p(\xi)$ is invertible, so we can write the last equation in the following form

$$\begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix} = L(\xi) \begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix} \begin{bmatrix} D_p^{-1}(\xi) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}$$

$$\begin{bmatrix} D_r(\xi) & -D_y(\xi) \end{bmatrix} \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} D_p^{-1}(\xi) & 0 \\ 0 & I \end{bmatrix}^{-1} = L(\xi) \begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix}$$

$$\begin{bmatrix} D_r(\xi)D_p(\xi) & D_r(\xi)-D_y(\xi) \end{bmatrix} = L(\xi)\begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix}.$$

Here, knowing that the matrix $\begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix}$ is a diagonal matrix, we can write the right hand side of the above equation as $\begin{bmatrix} D_p(\xi) & 0 \\ 0 & D_p(\xi) \end{bmatrix} L(\xi)$ and assign it to $L'(\xi)$. From (12), it follows that

$$L(\xi)M(\xi)c = 0 \implies L'(\xi)M(\xi)c = 0.$$

Accordingly, the kernel representation of the controller (still in terms of plant's parameters) can be written as

$$\mathcal{C} \equiv L'(\xi)M(\xi)c = 0. \quad (17)$$

Writing it explicitly, we have

$$\begin{bmatrix} D_r(\xi)D_p(\xi) & D_r(\xi)-D_y(\xi) \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & N_p(\xi) \end{bmatrix} \begin{bmatrix} e \\ u \end{bmatrix} = 0 \quad (18)$$

$$D_r(\xi)D_p(\xi)e + (D_r(\xi)-D_y(\xi))N_p(\xi)u = 0. \quad (19)$$

Pre-multiply the last equation by $D_p^{-1}(\xi)$, and re-arranging it yields

$$D_r(\xi)e + (D_r(\xi)-D_y(\xi))D_p^{-1}(\xi)N_p(\xi)u = 0. \quad (20)$$

From the structure of the feedback configuration, we have the relation $y = D_p^{-1}(\xi)N_p(\xi)u$. Also, $e = N_c^{-1}(\xi)D_c(\xi)u$. From the above, we obtain

$$N_c^{-1}(\xi)D_c(\xi)D_r(\xi)u + (D_r(\xi)-D_y(\xi))y = 0. \quad (21)$$

Pre-multiplying above by $N_c(\xi)$, it gives

$$D_c(\xi)D_r(\xi)u + N_c(\xi)(D_r(\xi)-D_y(\xi))y = 0 \Leftrightarrow$$
$$N_c(\xi)\bar{y} + D_c(\xi)\bar{u} = 0. \quad (22)$$

(2) $\implies$ (1): The proof of this implication is trivial, which can be obtained by substituting the filtered plant signals (13) in the kernel representation of the controller $\mathcal{C}$ (7). ■

One of the deep consequences of the above proposition is that the observed signals $\bar{w} = \mathtt{col}(\bar{u}, \bar{y})$ is independent of any particular setting in the feedback configuration, i.e. one can collect these signals with any arbitrary controller working in the closed-loop. Therefore, the controller synthesized in the above manner is a pure "data-driven online controller", i.e. a controller which is directly synthesized without a mathematical model of the plant but solely on the basis of the desired behavior and the any experimental input/output data produced by the plant. In this way, we can use the signal $\bar{w}$, which amounts to the measurements of the physical plant signals $\mathtt{col}(u,y)$, to design an online controller "which-when" makes an interconnection with the plant subject to faults yields the desired behavior. Interestingly, solving the equation (14) is a continuous-time system identification problem, which can be solved using various methods listed in the literature [2]. Note that on fixing (or if we know) the degree of controller's polynomials at the outset, the controller synthesized using the tools borrowed from the system identification community becomes an approximated controller that implements the desired behavior.

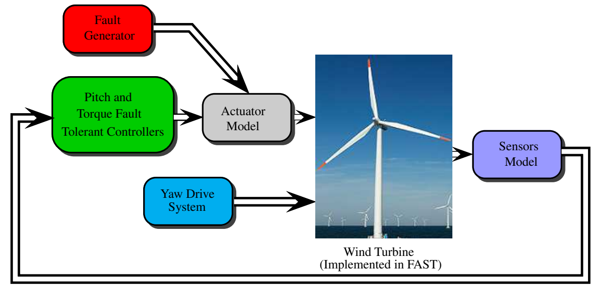

Fig. 1. Block diagram of the Simulink-based Wind Turbine FTC Systems

IV. Simulations

The model described in section II has just been discussed to show how the dynamics of the WT system evolves. However, we do not use any of this knowledge to demonstrate the proposed real-time fault-tolerant control mechanism. The two control inputs to the wind turbine are the generated torque $T_{g,r}$, and the blade pitch angle β_r. Since, we are working in Zone-3, the control objective is to track the generator power at its rated value of 5 MW. As suggested in [9], the main control scheme is developed in the torque control and the pitch controls from the industry standpoint. FAST also requires a yaw angular velocity and a yaw angular position as inputs. Here no yaw system is installed within the benchmark model since the direction of the wind is considered to be perpendicular to the blades. The torque controller is a nonlinear controller which depends on the generator speed and the pitch angle, and the pitch controller is a PI (Proportional + Integral) controller. In the proposed FTC design, the control objective can be achieved by reconfiguring only the PI pitch controller, described by

$$\mathcal{C}_{\Theta_f(t)} \equiv C_e^{\Theta_f(t)}(\xi)u = C_u^{\Theta_f(t)}(\xi)(r-y) \quad \text{where} \quad (23)$$
$$u = \beta_r, r = \omega_{g,ref}, y = \omega_{g,m}$$
$$C_e^{\Theta_f(t)}(\xi) = \xi, C_u^{\Theta_f(t)}(\xi) = k_p^{\Theta_f(t)}\xi + k_i^{\Theta_f(t)}.$$

The nonlinear torque controller, embedded within the benchmark model, will not be reconfigured. With this consideration, the closed-loop system can be viewed as a single-input single-output system. The structural description of the online redesign based Fault-tolerant Wind Turbine Control is shown in Fig. 1.

In the AD phase, the parameter space $\Theta(t) \in [0,1]$ was gridded with a 0.1 step yielding eleven points. For testing the proposed FTC scheme, we will focus on two significant faulty dynamics given by the grid values of $\Theta(t) \in \{0, 0.9\}$. The AD phase also provides the implementable desired behavior $\mathcal{D}$ covering the parameter space $\Theta(t)$. To collect the filtered signals, we only need the polynomials D_r and D_y of the desired behavior. These polynomials are given as $D_r(\xi) = 49, D_y(\xi) = \xi^2 + 12.6\xi + 49$ and a filtered measurement set is observed during every interval of length $\tau = 10sec$.

The experimental setup considers the wind profile with aerodynamics around the mean speed of 17 m/s, which is

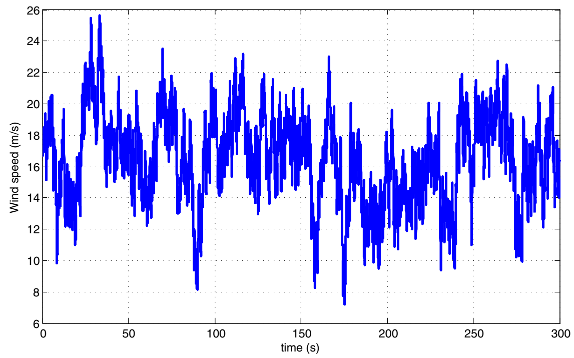

Fig. 2. Wind profile used for simulation

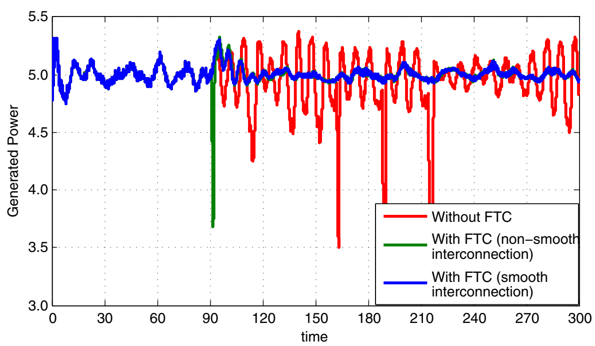

Fig. 3. Closed-loop Signals of Fault-Tolerant Wind Turbine System

illustrated Fig. 2, together with fault scenarios as discussed in subsection II-C, where the parametric values of the pitch system are taken as $\zeta_n = 0.6, \omega_{n,n} = 11.11, \zeta_f = 0.1, \omega_{n,f} = 1$. An experiment is run with an initial value of pitch actuator parameters as $\omega_n(\Theta(t) = 0), \zeta(\Theta(t) = 0)$. The parameters of the initial controller operating in the closed-loop are computed as

$$k_p^0 = 2.746 \times 10^{-3}, \quad k_i^0 = 6.76 \times 10^{-2}. \tag{24}$$

A pressure power drop fault appears within the WT system at time 80 *sec*, which changes the value of pitch parameters to $\omega_n(0.9), \zeta(0.9)$. Based on the theory developed in previous sections, a new controller is determined at interval of length $n \times \tau, n = 1, 2, \ldots$. Thus, a controller $\mathcal{C}_{0.9}$ is then computed online and switched into the closed-loop at time 90 *sec*. The computed parameters of $\mathcal{C}_{0.9}$ are given by

$$k_p^{0.9} = 5.113 \times 10^{-4}, \quad k_i^{0.9} = 7.044 \times 10^{-3}. \tag{25}$$

The generated power by the WT-FTC system is shown in Fig. 3. It has been shown here that without any fault-tolerant strategy, the dynamics of the system oscillates that might damage some internal components of the wind turbine. However, with the proposed real-time FTC strategy, the behavior of the system satisfies the desired behavior at anytime.

Indeed, the online synthesized controller is introduced in the closed-loop through a one-time switch. From the practical implementation point of view, avoiding the appearance of large transients during the switching of controllers is of utmost importance. These unpermitted transients due to an instant switching deteriorates the system performance. To tackle with this issue, we have used an approach proposed in [5, Section 4] that takes care of these undesirable transients appearing during the fault accommodation. In that approach, a real-time *smooth interconnection* between the unknown plant and the controller is guaranteed using the mathematical framework of behavioral theory. Clearly, the effect of real-time smooth interconnection can be visualized in the illustrated figure.

V. Conclusion

In this paper, we have demonstrated a successful implementation of a AFTC system to deal with occurring faults in the NREL's 5 MW Wind Turbine benchmark model such that the generator can produce the rated power. The novelty in the demonstrated FTC mechanism lies in its trajectory-based viewpoint derived from the behavioral theory. The fault accommodation delay is the time taken by an FTC algorithm to accommodate a fault from the time it appears in the system. In the integrated FD-FA approach, normally a significant amount of time is utilized during the fault diagnosis operation, and later the controller reconfiguration is initiated. No doubt the primary aim of any FTC system is to accommodate the fault as soon as possible. It has been shown here that an occurring fault is accommodated without using an explicit FD module. Clearly, the aforementioned prevailing issue with an integrated-FD FTC mechanism is not experienced here. As a consequence, the fault accommodation delay in the proposed approach might be smaller than the delay as experienced in the traditional FTC architecture.

References

[1] M. Blanke, M. Kinnaert, M. Staroswiecki, and J. Lunze. *Diagnosis and Fault tolerant control*. Springer-Verlag, 2003.

[2] H. Garnier and L. Wang, editors. *Identification of Continuous-time Models from Sampled Data*. Springer-Verlag London Limited, 2008.

[3] T. Jain. *Behavioral system theoretic approach to fault tolerant control*. PhD thesis, Université de Lorraine, 2012.

[4] T. Jain, J. J. Yamé, and D. Sauter. Synergy of canonical control and unfalsified control concept to achieve fault tolerance. In *World Congress*, volume 18, pages 14832–14837, 2011.

[5] T. Jain, J. J. Yamé, and D. Sauter. On implementing on-line designed controller for smooth interconnection in the behavioral framework. In *Robust Control Design*, volume 7, pages 424–429, 2012.

[6] T. Jain, J. J. Yamé, and D. Sauter. A real-time projection-based approach for fault accommodation in NRELs 5MW wind turbine systems. In *IEEE American Control Conference*, 2013.

[7] A.A. Julius, J. C. Willems, M. N. Belur, and H. L. Trentelman. The canonical controllers and regular interconnection. *Systems & Control Letters*, 54(8):787–797, 2005.

[8] E. Mazars, I. M. Jaimoukha, and Z. Li. Computation of a reference model for robust fault detection and isolation residual generation. *Journal of Control Science and Engineering*, Volume 2008, 2008.

[9] P. Odgaard and K. E. Johnson. Wind turbine fault detection and fault tolerant control - a second challenge. submitted, 2012.

[10] P. Odgaard, J. Stoustrup, R. Nielsen, and C. Damgaard. Observer based detection of sensor faults in wind turbines. In *European Wind Energy Conference (EWEA)*, 2009.

[11] J. W. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer-Verlag, 1997.

[12] A.J van der Schaft. Achievable behavior of general systems. *Systems & Control Letters*, 49(2):141–149, 2003.

[13] J. C. Willems and H. L. Trentelman. Synthesis of dissipative systems using quadratic differential forms: Part I. *IEEE Transactions On Automatic Control*, 47:53–69, 2002.