Hybrid Self Adaptive Learning Scheme for Simple and Multiple Drift-like Fault Diagnosis in Wind Turbine Pitch Sensors

Houari Toubakh and **Moamar Sayed-Mouchaweh**
IMT Lille Douai, Univ. Lille, Unite de Recherche Informatique Automatique, F-59000 Lille, France
e-mail: houari.toubakh@imt-lille-douai.fr

Abstract

This paper presents a hybrid dynamic data-driven approach to achieve simple and multiple drift like fault detection of pitch system sensors. This approach considers the system evolving in non-stationary environments and switching between several control modes. This switching is entailed by changes in the system environments. In each control mode, the system has a different dynamical behavior. The latter is described in a feature space sensitive to normal operating conditions in the corresponding control mode. These operating conditions are represented by restricted zones in the feature space called classes. The latter are characterized by a set of parameters representing their statistical properties, e.g. gravity center and variance-covariance matrix. The occurrence of an incipient fault entails a drift in the system operating conditions until the failure takes over completely. This drift manifests as a progressive change in the classes parameters in each control mode over time. The proposed approach monitors normal classes parameters in order to detect a drift in their characteristics. This drift detection allows achieving the fault in its early stages. It uses two drift indicators. The first indicator detects the drift and the second indicator confirms it. Both indicators are based on the observation of changes in the normal operating conditions characteristics over time. A wind turbine simulator is used to validate the performance of the proposed approach.

1 INTRODUCTION

The search for alternative clean energy is undoubtedly becoming more and more important in modern societies. The growing interest in wind energy production has led to the design of sophisticated wind turbines (WTs). Like every other complex and heterogeneous system, WTs are faced to the occurrence of faults that can impact their performance as well as their security. Therefore, it is crucial to design a reliable automated diagnostic system in order to achieve fault detection and isolation in early stage.

Fault diagnosis of WTs is a challenging task because of the high variability of the wind speed and the confusion between faults and noises as well as outliers. However, the fault diagnosis of pitch system is particularly a challenging task because of (i) the occurrence of pitch system faults in power optimization zone in which the fault consequences are hidden and (ii) the actions of the control feedback which compensate the fault effects. The role of the pitch system is to adjust the pitch of a blade by rotating it depending on the pitch angle position reference provided by the controller. The latter decides the pitch angle position reference according to the wind speed in order to allow an optimum energy production.

In the literature, there are several methods [6],[9],[11],[12],[1],[4],[15] that are used to achieve fault diagnosis in WTs. They achieve the fault diagnosis by reasoning over differences between desired or expected behavior, defined by a model, and observed behavior provided by sensors. They can be classified into two main categories of methods: internal and external methods. The internal methods [17],[18],[20] use a mathematical or structural model to represent the relationships between measurable variables by exploiting the physical knowledge or/and experimental data about the system dynamics. These variables represent the internal parts of the wind turbine. The response of the mathematical model is compared to the observed values of variables in order to generate indicators used as a basis for the fault diagnosis. Generally, the model is used to estimate the system state, its output or its parameters. The difference between the system and the model responses is monitored. Then, the trend analysis of this difference can be used to detect changing characteristics of the system resulting from a fault occurrence. The internal methods used to achieve the fault diagnosis of wind turbines are divided into three main categories: parameter estimation [8],[19], observer and state estimation based [3],[23] and signal analysis or feature based [7],[21] approaches. These methods were applied successfully to achieve the diagnosis of faults impacting the pitch system [19],[3],[16], the generator [16],[14], the converter [25],[14], and the gearbox [26],[16].

The major advantages of these methods are their ability to detect both the abrupt and progressive failures via trend analysis, and they give a precise decision or isolation of a failure. However, they suffer from the necessity to depth information about system behavior and failures which is hard to obtain for complex and strong non-stationary systems as wind turbines.

An alternative to overcome this problem is the external methods [17],[22],[9]. The external methods consider the system as a black box, in other words, they do not need any mathematical model to describe the system dynami-

cal behaviours. They use exclusively a set of measurements or/and heuristic knowledge about system dynamics to build a mapping from the measurement space into a decision space. They include expert systems and machine learning and data mining techniques. These methods are suitable for systems that are difficult to model, they are simple to implement and require short processing time. However, since the obtained models are not transparent, the obtained results are hard to be interpreted and demonstrated. There are several machine learning and data mining methods used to achieve the fault diagnosis of wind turbines. Such methods are described and successfully applied in [24],[2].

Few approaches have been proposed to achieve early fault diagnosis of WTs, in particular pitch sensors. This is due to the fact that modeling component degradation in strong non-linear and complex non-stationary environments is very hard task. Examples of these methods, we can cite genetic algorithm [10], neural network, the boosting tree algorithm, and support vector machine [9]. These methods do not integrate a mechanism to detect a drift by analyzing the characteristics of incoming data and to update the model parameters and structure in response to this drift. Therefore, they do not achieve a reliable early diagnosis. Consequently, the diagnosis performance (diagnosis delay) is decreased significantly for faults occurring in WT critical subsystems as pitch systems ones.

This paper presents a new data-driven based approach in order to achieve a reliable drift monitoring and diagnosis of simple and multiple drift-like faults that can affect wind turbine pitch sensors. This approach takes into account the different dynamical behaviors of WTs according to the wind speed. The goal is to detect a drift from normal operating conditions using only the recent and useful data. Initial offline modeling allows constructing initial classes based on the historical data set. These classes characterize the operating conditions of the pitch system (normal/faulty) and are represented by restricted zones in the feature space. The latter is formed by sensitive features to pitch sensor operating conditions in order to distinguish any drift from normal to fault operating conditions. The modeling tool is an algorithm called AuDyC (Auto-Adaptive Dynamical Clustering) used to initialize the classes that will be dynamically updated.

In this work, two-dimensional feature space is constructed, for the sensor faults. The faulty classes, representing the failure operating conditions of pitch sensor, are considered to be a priori unknown. There is one known class in advance.The class represents the pitch sensor normal operating conditions. It considers gradual degradations in pitch sensor operating condition as a drift in the characteristics of normal class over time. Detecting and following this drift can help to predict the occurrence of pitch sensor failure.

The drift-like fault is monitored using two drift indicators: one to detect a drift and the second one to confirm it. When the drift is detected by the first indicator, a warning is emitted to human operators. Then, the second drift indicator confirms this drift in order to inform human operators of the necessity to react by taking the adequate correction actions.

The proposed data-driven approach is composed of five main steps: processing and data analysis, clustering and classification, drift monitoring, updating and interpretation steps.

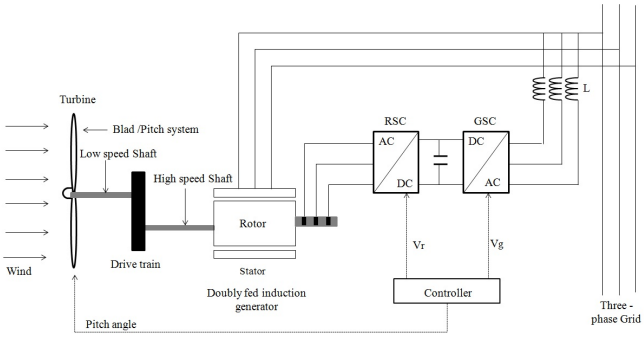

Figure 1: Wind turbine components.

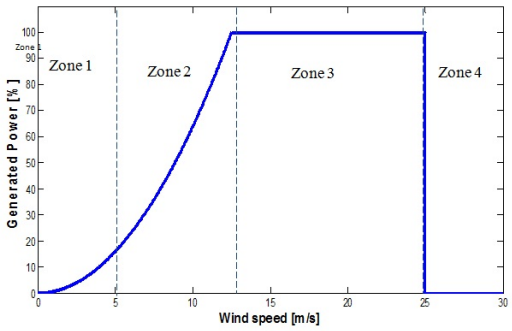

Figure 2: Reference power curve for the WT depending on the wind speed.

2 Pitch system within wind turbines

The wind turbine model under study is composed of five principal parts: the blades, the drive train, the generator with the converter, and the controller (see Figure 1). It can be seen that the blades are fixed to the main axis, which in turn is connected to the generator through the drive train. The generator is electrically connected to the converter, which in turn is connected to a transformer. The blades are pitched by the pitch actuators.

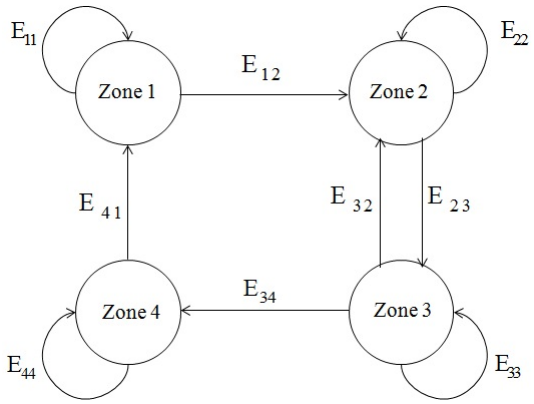

Figure 3: Controller operating zones modeled by a finite state automaton.

The controller operates in four zones (see Figure 2). Zone 1 is the start-up of the turbines, zone 2 is power optimization, zone 3 is constant power production and zone 4 is no power production due to a too high wind speed.

In order to handle transitions between the control modes,

the controller checks the operating zone in which the WT is by observing the wind speed. The transitions between the control modes change the dynamics of the pitch system. Each control mode is active in one zone thus it is modeled by a finite state automaton. Each zone is represented by a state in which a specific control mode or strategy is defined. According to the wind speed, the control mode changes by switching from one mode or state to another mode or state. This switching between control modes is achieved by discrete events. As an example, if the WT was initially in control mode related to the zone 1, as long as the wind speed is less than a predefined threshold (5 m/s in Figure 2) E_{11} will be generated. E_{11} keeps the WT in control mode 1. If the wind speed is greater than the predefined threshold for zone 1 (5 m/s in Figure 2), The event E_{12} is generated leading to switch the WT from the control mode related to zone 1 to the control mode related to zone 2 (see Figure 3). Same reasoning can be applied for the other events.

The focus of this benchmark model is on the operation of WT in zones 2 and 3. Two control strategies are applied to optimize the energy production and keep it constant at its optimal value: the converter torque control in zone 2 and the blades angle control in zone 3 (see Figure 4). In zone 2, the WT is controlled so that it produces as much energy as possible. To do so, the blades angle is maintained equal to 0° and the tip speed ratio is kept constant at its optimal value. The latter is regulated by the rotating speed control by tuning the converter torque. Once the optimal power production is achieved, the blades angle control maintains the converter torque constant and adjusts the rotating speed by controlling the blades angle. The latter modifies the transfer of the aerodynamic power of the wind on the blades. In this work, the controller modes are modeled by a finite state automaton containing two states (see Figure 4). In the following, zones 2 and 3, respectively, correspond to control modes 1 and 2:

Control Mode 1 In this control mode, the power optimum value is achieved by setting the pitch reference to zero $\beta[t] = 0$ and the reference torque to the converter $\tau_{g,r}$ as follows:

$$\tau_{g,r} = K_{opt} \times \left(\frac{\omega_g [t]}{N_g} \right)^2 \quad (1)$$

N_g is the gear ratio and n is the sampling time.
Where

$$K_{opt} = \frac{1}{2} \rho A R^3 \frac{C_{P_{\max}}}{\lambda_{opt}^3} \quad (2)$$

with ρ the air density, A the area swept by the turbine blades, $C_{P_{max}}$ the maximum value of power coefficient, and λ_{opt} the optimal value of λ is found as the optimum point in the power coefficient C_P mapping of the WT. The power coefficient mapping characterizes the efficiency of energy and it depend on λ and β.

Control Mode 2 In this mode, the major control actions are handled by the pitch system using a Proportional Integral (PI) controller trying to keep $\omega_g[t]$ at ω_g.

$$\beta_r(t) = \beta_r(t-1) + k_p.e(t) + (k_i.T_s.k_p).e(t-1) \quad (3)$$

When $e(t) = \omega_r(t) - \omega_{nom}$. In this case the converter reference is used to suppress fast disturbances:

$$\tau_{g,r}(t) = \frac{P_r(t)}{\omega_t(t)} \quad (4)$$

The control mode should switch from mode 1 to mode 2 if the following condition is satisfied:

$$E_{23} : \omega_g(t) \geq \omega_{nom} \quad (5)$$

The satisfaction of this condition generates a discrete event, E_{23}, allowing the switching from control mode 1 to control mode 2. The goal to obtain P_g equal to P_r. This condition is satisfied when the wind speed is greater than predefined threshold for zone 2 (12.5 m/s in Figure 2). Likewise, the control mode should switch from control mode 2 to control mode 1 if the following condition is satisfied:

$$E_{32} : \omega_g(t) < \omega_{nom} - \omega_\Delta \quad (6)$$

Where ω_{nom} is the nominal generator speed and ω_Δ is a small offset subtracted from the nominal generator speed to introduce some hysteresis in the switching scheme, thereby avoiding that the control modes are switching all the time [15]. The satisfaction of this condition generates a discrete event, E_{32}, allowing the switching from control mode 2 to control mode 1. This condition is satisfied when the wind speed is less than the wind speed threshold defined for zone 3 (12.5 m/s in Figure 2).

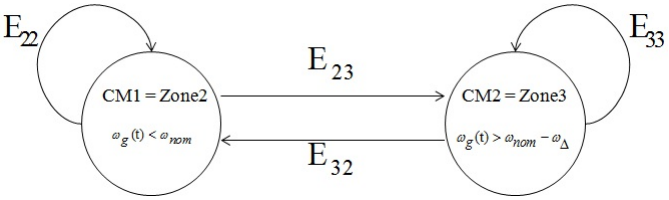

Figure 4: Controller modes modeled by a finite state automaton

As we said before, the benchmark model allows simulating the WT behavior in two power zones: 1) zone 2 (power optimization) where τ_g is controlled and β_r is equal to zero and; 2) zone 3 (optimal energy production) where τ_g is kept β_r constant and is controlled. In this paper, we focus on pitch sensor faults as it is discussed in subsection 2.

3 Pitch system description

The considered WT is horizontal-axis based with three blades. Each blade is equipped with an actuator. The role of the pitch actuator is to adjust the pitch of a blade by rotating it; Each actuator is provided by the same pitch angle reference β_r. The pitch angle of a blade is measured on the cylinder of the pitch actuator, each pitch position (angle) β_{m_i} where $i \in \{1, 2, 3\}$ is measured with two sensors where index m_i represents the i^{th} sensor of the corresponding variable (see Figure 5). The pitch system feedback β_f is an internal variable used to model the pitch position error caused by sensor faults:

$$\beta_f = \beta_r - \frac{1}{2}(\beta_{k,m1} + \beta_{k,m2}) \quad (7)$$

The controller is fed by the mean value of the readings of the two sensors. Hence, this sensor fault is modeled as a change

in the pitch references, meaning that a sensor fault resulting in changed mean value should also change the pitch reference accordingly [15].

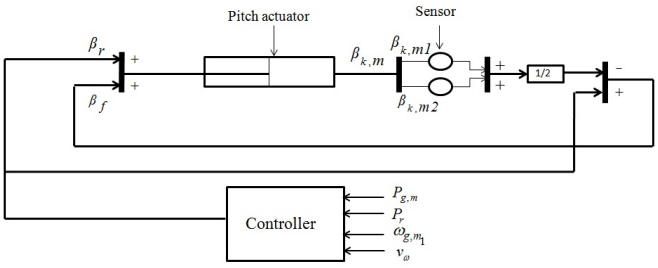

Figure 5: Block diagram of pitch system for the blade k, $(k = 1, 2, 3)$

4 Pitch system modeling

The hydraulic pitch system is modeled in the benchmark as a closed loop of dynamic system. The state representation of the nominal pitch system dynamics is defined as follows [15]:

$$\begin{aligned} \dot{x}_p &= A_p x_p + B_p (\beta_r + \beta_f) \\ y_p &= C_p x_p \\ A_p &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \\ B_p &= \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} \\ C_p &= \begin{bmatrix} 0 & 1 \end{bmatrix} \end{aligned} \tag{8}$$

The state vector $x_p = \begin{bmatrix} \dot{\beta}_k & \beta_k \end{bmatrix}^T$ is composed of pitch angular speed $\dot{\beta}_k$, and position β_i for each blade k : $(k = 1, 2, 3)$. y_p is the measured pitch position, β_r is the pitch angle position reference provided by the controller, and β_r is the feedback pitch system (see Figure 5). ω_n, ζ are the parameters of the pitch system where ω_n represent the natural frequencies and ζ is the damping ratio.

The pitch system represent a hybrid dynamic system and especially it belongs to the class of Discretely Controlled Jumping Systems (DCJS), In these systems, the continuous state variables change discontinuously under the influence of an external action (e.g., a command) as the case for electromagnetic systems with pulse inputs [**?**]. The pitch system state variable $x_p = \begin{bmatrix} \dot{\beta}_k & \beta_k \end{bmatrix}^T$ changes discontinuously under the influence of an external action defined by Equation 5 and 6.

5 Pitch system drift-like fault scenarios generation

In this paper the types of fault which are considered in this work are simple and multiple drift-like fault in pitch sensors. The following subsections detail the generation of several scenarios representing drift-like faults with three different speeds in pitch sensor β_{m1} and pitch sensor β_{m2}, and in both pitch sensors β_{m1} and β_{m2}.

5.1 Sensor drift-like fault

Each blade is equipped with an actuator. Each actuator is provided by the same pitch angle reference β_r. In addition, each pitch position, (angle) β_{mi} is measured with two sensors where index i represents the i^{th} sensor of the corresponding variable. The fault scenarios related to simple drift-like fault in pitch sensor $n°1$ and sensor $n°2$ and multiple drift-like fault in both pitch position sensor $n°1$ and sensor $n°2$ in blade $n°3$ are summarized respectively in Table 1, Table 2 and Table 3. The state representation of the pitch system after the integration of a fault in sensor β_{mi}, $i \in \{1, 2\}$ is defined as follow:

$$\begin{aligned} \dot{x_p} &= Ax_p + Bu \\ y_p &= Cx_p + f(t) \\ f(t) &= \lambda_i . (t_b - t_e) \end{aligned} \tag{9}$$

Therefore the parameter λ_i, $i \in \{1, 2\}$ is used in the simulation to generate a fault in sensor β_{mi} during the time period $(t_b - t_e)$ where t_b is the start time and t_e is the end time of sensor drift-like fault.

Simple drift-like fault in sensor β_{m1}

In this paper the simple drift-like fault scenarios in pitch sensor 1 (β_{m1}) scenarios are modeled as a gradual change in the coefficient λ_1 of pitch sensor $n°1$ in blade $n°3$ where t_b is the beginning of the drift and t_e is the end of the drift. Nine scenarios for simple sensor drift-like fault are generated in order to simulate slow, moderate and high degradation speeds represented by slow, moderate and high drift speeds (see Figure 6). Each drift speed scenario is generated at three different time instances. Thus, parameter λ_1 is changed linearly from λ_{1N} to λ_{1F} in a period of 30s, 60s and 90s, corresponding respectively to high, moderate and slow drift speeds. Then, the fault remains active for 200s. Finally the parameter λ_1 decreases again to return to its initial value λ_{1N} (see Figure 6 for the case of high drift speed in sensor 1 (β_{m1})).

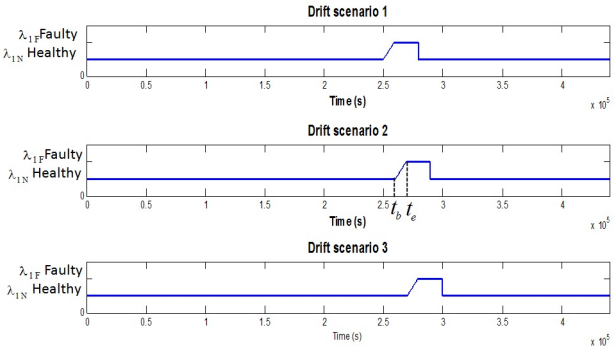

Figure 6: Simple drift-like fault scenarios in pitch sensor 1 (β_{m1}), corresponding to high drift speed in 3 different time instances t_b is the beginning time of the drift and t_e is the end of the drift.

Simple drift-like fault in sensor β_{m2}

The simple drift-like fault scenarios in pitch sensor 2 (β_{m2}) scenarios are modeled as a gradual change in the coefficient

Fault N°	Drift speed	Simple drift-like fault in pitch sensor β_{m1}	Period
F_{4h}	30s (High)	$\lambda_{1N} \rightarrow \lambda_{1F}$	2500s -2730s
F_{4m}	60s (Medium)	$\lambda_{1N} \rightarrow \lambda_{1F}$	2500s -2760s
F_{4s}	90s (Slow)	$\lambda_{1N} \rightarrow \lambda_{1F}$	2500s -2790s
F_{5h}	30s	$\lambda_{1N} \rightarrow \lambda_{1F}$	2600s -2830s
F_{5m}	60s	$\lambda_{1N} \rightarrow \lambda_{1F}$	2600s -2830s
F_{5s}	90s	$\lambda_{1N} \rightarrow \lambda_{1F}$	2600s -2890s
F_{6h}	30s	$\lambda_{1N} \rightarrow \lambda_{1F}$	2700s -2930s
F_{6m}	60s	$\lambda_{1N} \rightarrow \lambda_{1F}$	2700s -2960s
F_{6s}	90s	$\lambda_{1N} \rightarrow \lambda_{1F}$	2700s -2990s

Table 1: Simple drift-like fault scenarios in pitch sensor 1 (β_{m1}).

λ_2 of pitch sensor $n^\circ 2$ in blade $n^\circ 3$ where t_b is the beginning of the drift and t_e is the end of the drift. As for the case of simple drift-like fault in pitch sensor β_{m1} scenarios, nine scenarios for simple sensor drift-like fault are generated in order to simulate slow, moderate and high degradation speeds represented by slow, moderate and high drift speeds (see Figure 7). Each drift speed scenario is generated at three different time instances. Thus, parameter λ_2 is changed linearly from λ_{2N} to λ_{2F} in a period of 30s, 60s and 90s, corresponding respectively to high, moderate and slow drift speeds. Then, the fault remains active for 200s. Finally the parameter λ_2 decreases again to return to its initial value λ_{2N} (see Figure 7 for the case of high drift speed in sensor 2, (β_{m2})).

Fault N°	Drift speed	Simple drift-like fault in pitch sensor β_{m2}	Period
F_{7h}	30s (High)	$\lambda_{2N} \rightarrow \lambda_{2F}$	2800s -3030s
F_{7m}	60s (Medium)	$\lambda_{2N} \rightarrow \lambda_{2F}$	2800s 3060s
F_{7s}	90s (Slow)	$\lambda_{2N} \rightarrow \lambda_{2F}$	2800s- -3090s
F_{8h}	30s	$\lambda_{2N} \rightarrow \lambda_{2F}$	2900s -3130s
F_{8m}	60s	$\lambda_{2N} \rightarrow \lambda_{2F}$	2900s -3130s
F_{8s}	90s	$\lambda_{2N} \rightarrow \lambda_{2F}$	2900s -3190s
F_{9h}	30s 30s	$\lambda_{2N} \rightarrow \lambda_{2F}$	3000s -3230s
F_{9m}	60s	$\lambda_{2N} \rightarrow \lambda_{2F}$	3000s -3260s
F_{9s}	90s	$\lambda_{2N} \rightarrow \lambda_{2F}$	3000s -3290s

Table 2: Simple drift-like fault scenarios in pitch sensor 2 (β_{m2}).

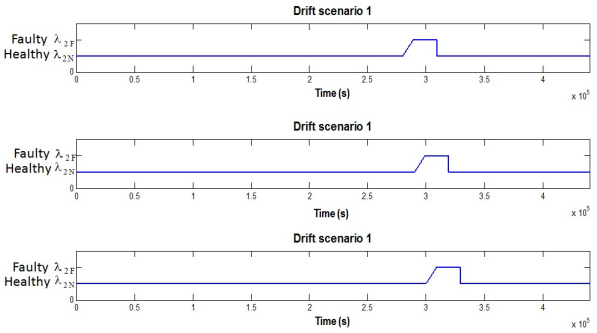

Figure 7: Simple drift-like fault scenarios in pitch sensor 2 (β_{m2}), corresponding to high drift speed in 3 different time instances.

Multiple sensor drift-like fault

In this chapter the generated scenarios of the multiple drift-like fault in pitch sensor 1 (β_{m1}) and sensor 2 (β_{m2}) are modeled as a gradual change at the same time in the drift coefficient (λ_1 and λ_2) of both pitch sensors $n^\circ 1$ and pitch sensors $n^\circ 2$ in blade $n^\circ 3$. As for the case of simple drift-like fault in pitch sensor scenarios, nine scenarios for multiple sensor drift-like fault are generated in order to simulate slow, moderate and high degradation speeds representing by slow, moderate and high drift speeds (see Table 3). Each drift speed scenario is generated at three different time instances. Thus, parameters λ_1 and λ_2 are changed linearly from λ_{1N} and λ_{2N} to λ_{1F} and λ_{2F} in a period of 30s, 60s and 90s, corresponding respectively to high, moderate and slow drift speeds. Then, the fault remains active for 200s. Finally the parameter decreases again to return to their initial values (see Figure 8 for the case of high drift (degradation) speed in both sensor 1 (β_{m1}) and sensor 2 (β_{m2})).

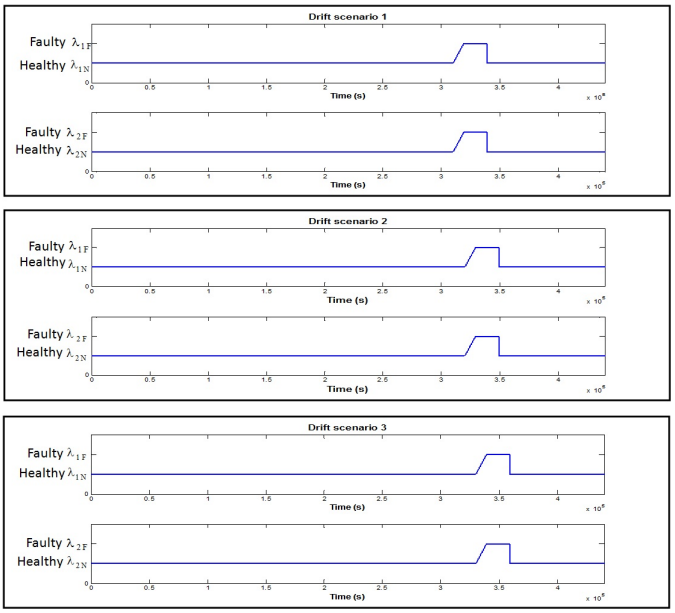

Figure 8: Multiple sensor drift-like fault scenarios in sensors (β_{m1}) and (β_{m2}) corresponding to high drift speed in 3 different time instances.

Fault N°	Drift speed	Multiple drift-like fault in in pitch sensors β_{m2} and β_{m2}	Period
F_{10h}	30s (High)	$\lambda_1 N \rightarrow \lambda_{1F}$ and $\lambda_{2N} \rightarrow \lambda_{2F}$	3100s-3330s
F_{10m}	60s (Medium)	$\lambda_1 N \rightarrow \lambda_{1F}$ and $\lambda_{2N} \rightarrow \lambda_{2F}$	3100s-3360s
F_{10s}	90s (Slow)	$\lambda_1 N \rightarrow \lambda_{1F}$ and $\lambda_{2N} \rightarrow \lambda_{2F}$	3100s-3390s
F_{11h}	30s	$\lambda_{1N} \rightarrow \lambda_{1F}$ and $\lambda_{2N} \rightarrow \lambda_{2F}$	3200s-3430s
F_{11m}	60s	$\lambda_{1N} \rightarrow \lambda_{1F}$ and $\lambda_{2N} \rightarrow \lambda_{2F}$	3200s-3460s
F_{11s}	90s	$\lambda_{1N} \rightarrow \lambda_{1F}$ and $\lambda_{2N} \rightarrow \lambda_{2F}$	3200s-3490s
F_{12h}	30s	$\lambda_{1N} \rightarrow \lambda_{1F}$ and $\lambda_{2N} \rightarrow \lambda_{2F}$	3300s-3530s
F_{12m}	60s	$\lambda_{1N} \rightarrow \lambda_{1F}$ and $\lambda_{2N} \rightarrow \lambda_{2F}$	3300s-3560s
F_{12s}	90s	$\lambda_{1N} \rightarrow \lambda_{1F}$ and $\lambda_{2N} \rightarrow \lambda_{2F}$	3300s-3590s

Table 3: Multiple drift-like fault scenarios in pitch sensors (β_{m1}) and (β_{m2}).

6 Proposed approach

In this section, hybrid dynamic data-driven approach is developed in order to achieve condition monitoring and drift like fault detection of pitch sensor. It performs predictive diagnosis by detecting a drift of the system operating conditions from normal to faulty modes. The proposed approach is based on 5 steps developed in the following subsections (see Figure 9).

6.1 Processing and data analysis

This step aims at finding the features that are sensitive to the system operating conditions in order to construct the feature space. A feature space representing the operating conditions of each assembly of WT is defined, this feature space will be responsible of the detection and isolation of faults impacting this components. The research of sensitive features is based on the signals provided by the pitch sensors as well as the prior knowledge about the system dynamics. These features are chosen in order to maximize the discrimination between operating conditions in the feature space. In this paper, two-dimension feature space is constructed for the sensor fault. The goal of the feature space use, at the level of component, is to facilitate the drift-like fault isolation and to enhance the diagnosis robustness.

The position of the pitch actuators is measured by two redundant sensors for each of the three pitch positions $\beta_{k,mi}$, $k = 1, 2, 3$, $i = 1, 2$, with the same reference angle β_r provided to each of them. In order to enhance the robustness against noise, the measurements are filtered by a first order filter using time constant $\tau = 0.06$.

For the drift like fault detection and isolation of the sensor faults, we propose to explore the physical redundancy in order to generate residuals as follows:

$$\Delta\beta_{s1} = |\beta_r + \beta_f - \beta_{m1}| \quad (10)$$

$$\Delta\beta_{s2} = |\beta_r + \beta_f - \beta_{m2}| \quad (11)$$

To do so, the residual $\Delta\beta_{sn}$, $n = 1, 2$, is generated by the comparison between the pitch angle measurement β_{mi}, $i = 1, 2$, $m = 1, 2, 3$ and the command computed by the sum of the desired value of the pitch angle β_r and the feedback pitch system β_f (see Figure 5). The residual is computed within a time window which is tuned to be several times the actuator time response.

The evolution of these residuals with respect to each of the two sensors is considered as meaningful features. Indeed, the residual $\Delta\beta_{s1}$ respectively $\Delta\beta_{s2}$, is equal to zero when the corresponding sensor β_{m1} respectively β_{m2}, is in normal operating conditions. When, the sensor β_{m1} respectively β_{m2}, is in faulty operating conditions, the residual $\Delta\beta_{s1}$, $\Delta\beta_{s2}$ will be different of zero because this sensor will not measure the new value of command $(\beta_r + \beta_f)$ (see Figure 5). Indeed, the command $(\beta_r + \beta_f)$ will change in order to compensate the difference between the two sensors due to the fault of sensor β_{m1} respectively β_{m2}.

6.2 Classifier learning and updating

The clustering looks to determine the number of classes contained in the learning set and to initialize their parameters. The classification aims at designing a classifier able to assign a new pattern to one of the learnt classes in the feature space. A new pattern characterizes the actual operating conditions (normal or faulty in response to the occurrence of a certain fault) of the system. Examples of these approaches are present in [5] as well as in the references of this paper.

Auto-adaptive Dynamical Clustering Algorithm (AuDyC) [13] is selected in this work in order to achieve both clustering and classification. AuDyC computes the parameters of initial classes based on the statistical properties of data which are the mean and the variance-covariance matrix. These classes characterize the normal operating conditions of pitch sensors. AuDyC was chosen because it is unsupervised classification method and is able to model streams of patterns since it always reflects the final distribution of patterns in the features space. It uses a technique that is inspired from the Gaussian mixture model [13]. Let E^d be a d-dimensional feature space. Each feature vector $x \in E^d$ is called a pattern. The patterns are used to model Gaussian prototypes P^j characterized by a center $\mu_{P^j} \in R^{d\times 1}$ and a covariance matrix $\sum_{P^j} \in R^{d\times d}$. Each Gaussian pro-

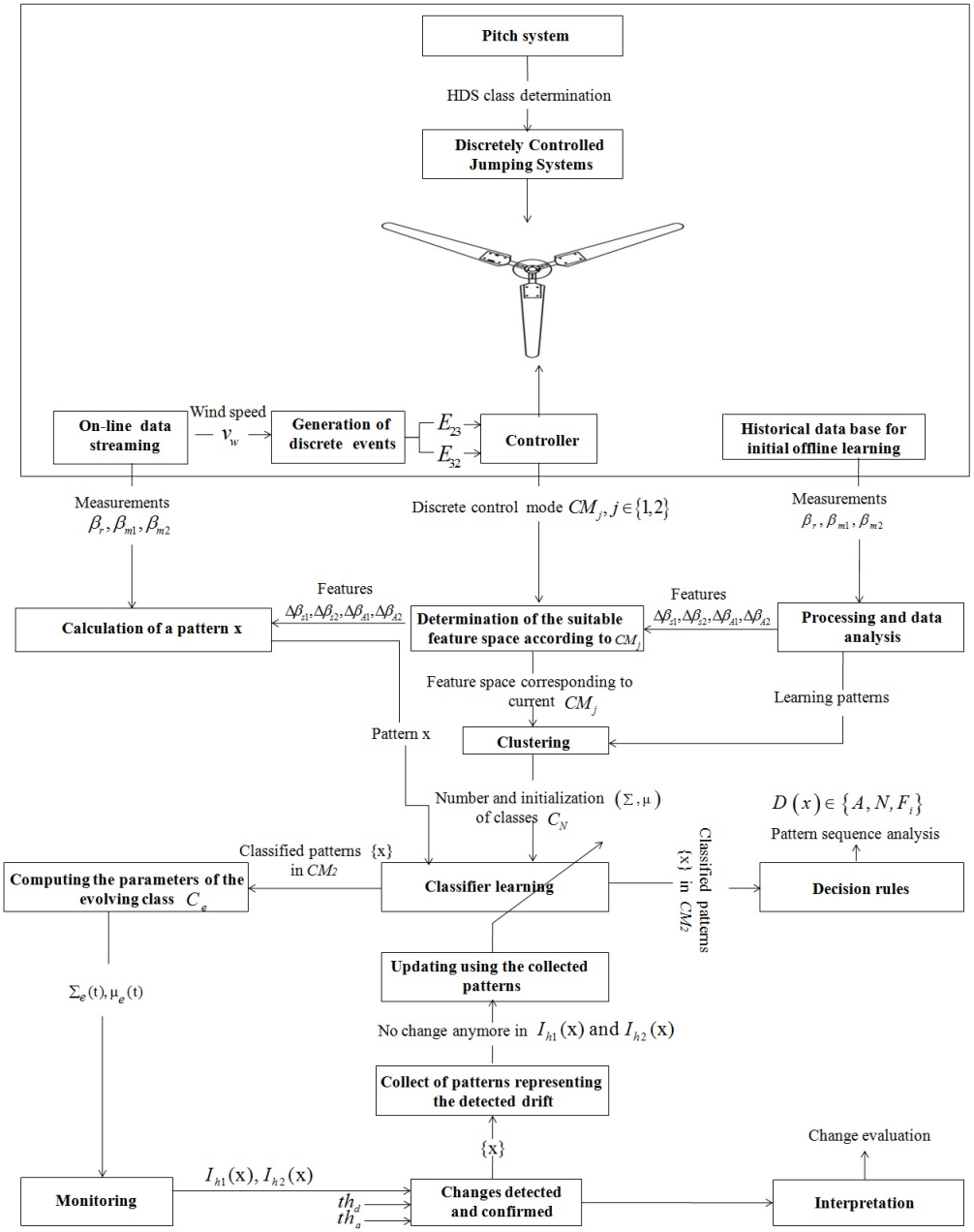

Figure 9: Proposed on-line adaptive scheme steps.

totype characterizes a class. A minimum number of N_{win} patterns are necessary to define one prototype, where N_{win} is a user-defined threshold. A class models operating conditions and gathers patterns that are similar one to each other. The similarity criterion that is used is the Gaussian membership degree. Faults will affect directly this distribution and this will be seen through the continuously updated parameters. More details about AuDyC related to merging classes, splitting classes, rules of recursive adaptation, similarity criteria, etc., can be found in [13].

In the sensor feature space, four classes are considered: the fault of sensor 1, β_{m1} , the fault of sensor 2, β_{m2}, the fault of both sensor 1,β_{m1} and sensor 2 β_{m2}, and the normal functioning. Figure 10 shows the classes representing normal and failure operating conditions of pitch sensor in the feature space constituted by the two residuals defined by Equation 10 and 11. In zone 2, the effects of this fault are hidden because the actuators are not operated. Moreover, it is strongly difficult to distinguish the fault occurrence to the noise in the case of small angles. Therefore an overlapping region is created between the normal and failure classes (see Figure 10 and Figure 15).

In order to answer the challenges inherent to the system operation, the normal and failure classes are split into five classes and the pitch actuator dynamics are represented by two different control modes. The first one corresponds to the case of zone 2 low wind speed; while the second control mode represents the case of zone 3 high wind speed (see Figure 16). Class 1 is the ambiguity class. It gathers the patterns representing pitch sensor normal or faulty operating conditions. This class represents the control mode 1. Class 2 represents the normal operating conditions class in

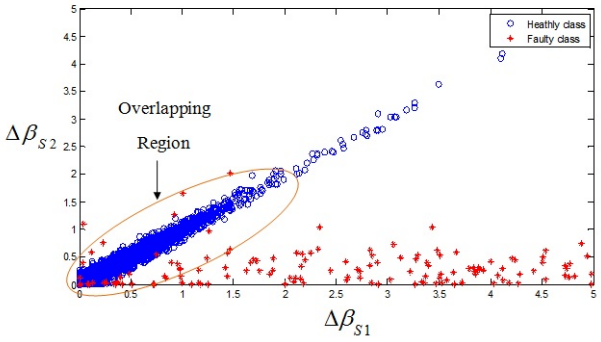

Figure 10: Large view of overlapping region for the pitch sensor normal and failure operating conditions in case of simple fault in pitch sensor 1, (β_{m1}).

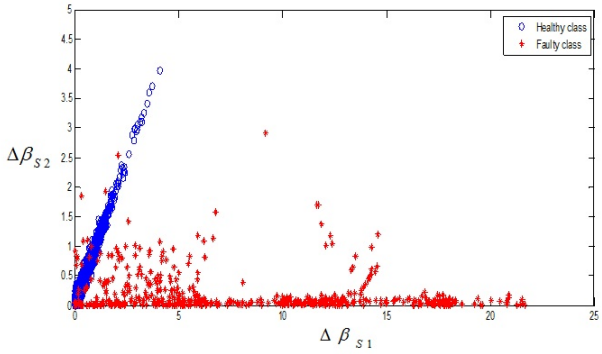

Figure 11: Feature space of the pitch sensor normal and failure operating conditions in case of simple fault in pitch sensor 1, (β_{m1}).

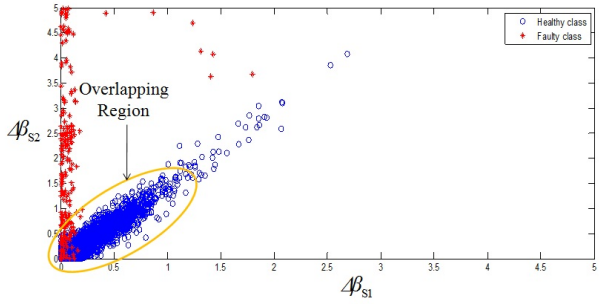

Figure 12: Large view of overlapping region for the pitch sensor normal and failure operating conditions in case of simple fault in pitch sensor 2, (β_{m2}).

control mode 2. Class 3 represents failure class caused by simple drift-like fault in pitch sensor 1, β_{m1} in control mode 2, class 4 represents failure class caused by simple drift-like fault in pitch sensor 2, β_{m2} in control mode 2 and class 5 represents failure class caused by multiple drift-like fault in pitch sensor 1, β_{m1} and sensor 2, β_{m2} in control mode 2.

The updating step aims at reacting to the changes in classes characteristics in the feature space. AuDyC continuously updates the classes parameters by using the recursive adaptation Rules 12 and 13. In such a way, its validity and

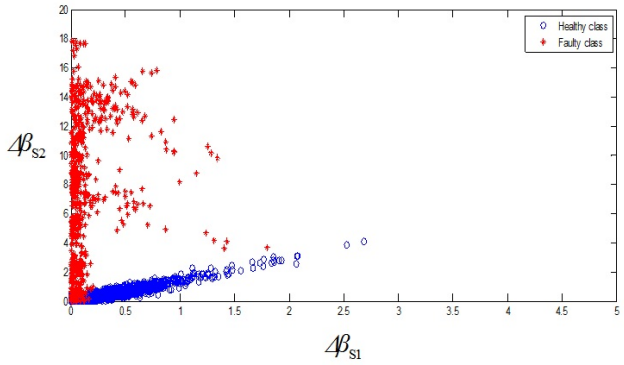

Figure 13: Feature space of the pitch sensor normal and failure operating conditions in case of simple fault in pitch sensor 2, (β_{m2}).

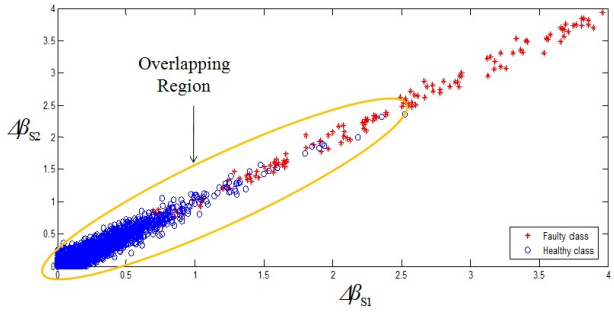

Figure 14: Large view of overlapping region for the pitch sensor normal and failure operating conditions in case of multiple fault in pitch sensor 1, (β_{m1}) and pitch sensor 2, β_{m2}.

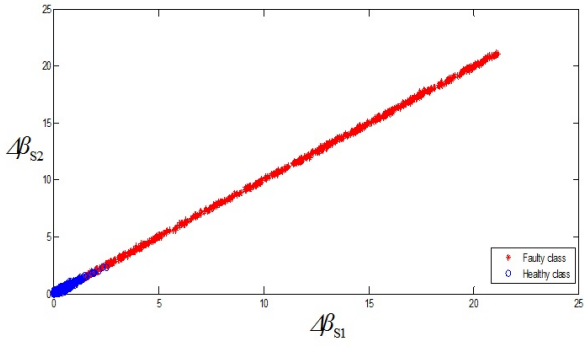

Figure 15: Feature space of the pitch sensor normal and failure operating conditions in case of multiple fault in sensor β_{m1} and β_{m2}.

performance over time is preserved.

$$\mu_e(t) = \mu_e(t-1) + f(\mu_e(t-1), x^{new}, x^{old}, N_{win}) \quad (12)$$

$$\sum_e(t) = \sum_e(t-1) + g(\sum_e(t-1), \mu_e(t-1), x^{new}, x^{old}, N_{win}) \quad (13)$$

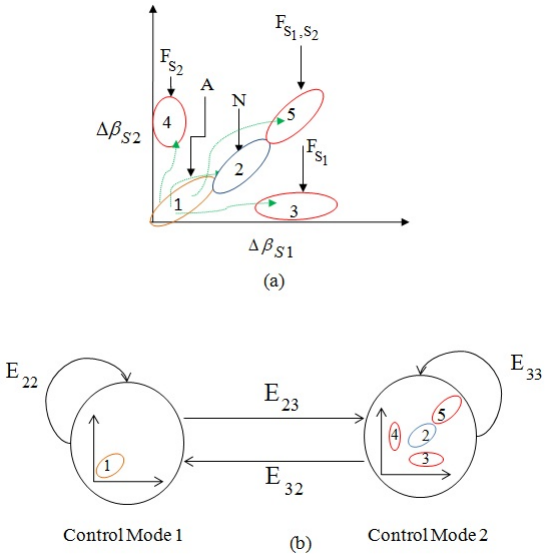

Figure 16: (a) Sensor decision space. (b) Control modes 1 and 2 modeled by a finite state automaton.

where x^{new} and x^{old} are respectively, the newest and the oldest arrived pattern in the time window N_{win} .

Initial off-line modeling allows the construction of initial classes that characterize knowledge from historical data. The historical data are usually sensor data that are saved. AuDyC is used to initialize the parameters of classes that will be dynamically updated. Knowledge of failure modes given from (labeled) historical data can help building a classification scheme for fault diagnosis. However, in reality, these data are hard to obtain.

In this work, we suppose that only data corresponding to normal operating conditions (normal classes) are known in advance. The training of the process by applying AuDyC is made based on features that are extracted from historical sensor data once finished; the class corresponding to normal operating conditions is retained. We denote this class by $C_N = (\mu_N, \Sigma_N)$.

In on-line functioning, the parameters of C_N are dynamically updated by AuDyC for each new pattern arrived in control mode 2. This yields changes in the class parameters which continuously reflect the distribution of the newest arriving patterns. We denote by $C_e = (\mu_e, \Sigma_e)$ the evolving classes in feature space. We have $C_e(t = 0) = (\mu_e, \Sigma_e) = C_N$.

In control mode 1 of pitch system, pitch sensor normal and faulty behaviors cannot be distinguished. Thus, in the proposed approach, the decisions about the status (normal/faulty) of patterns located in this region are delayed. Therefore in this case, the classifier will not be updated in order to avoid integrating in the drift time window useless patterns. In order to detect the drift as soon as possible, AuDyC updates the classes parameters by using a window that contains only the patterns belonging to control mode 2. AuDyC is dynamic by nature in the sense that it continuously updates the parameters of the classes as new patterns arrive.

6.3 Pattern decision analysis

When a new pattern is classified in the ambiguity class (A), in sensor feature space, assigning it to normal or failure operating conditions is a risky decision since normal and failure classes are overlapped in this region of the feature space. In order to reduce this risk, the decision about the status (normal or faulty) of any pattern classified in this region is delayed by assigning the label (A) (ambiguity decision). Then, this ambiguity can be removed by analyzing the past and future decisions of this pattern. The analysis of the pattern decision sequence is achieved by using a set of decision rules allowing assigning to ambiguity patterns label (N) or label (F) (normal or faulty) as follows. Let us suppose that $X_A = \{x_t, x_{t+1}, \ldots, x_{t+n}\}$ is a set of patterns associated with decision (A). Let x_{t-1} be the previous pattern arrived just before x_t. Let $D(x_{t-1}) \in \{A, N, F_i\}$ be the decision of this pattern. Let x_{t+n+1} the pattern arrived just after x_{t+n}. Let $D(x_{t+n+1}) \in \{A, N, F_i\}$ be the decision for this pattern. Then, the decision can be updated as follows:

$$D(x_{t-1}) = N \wedge D(x_{t+n+1}) = N \Rightarrow D(x) = N, \forall x \in X_A \quad (14)$$

$$D(x_{t-1}) = F \wedge D(x_{t+n+1}) = F \Rightarrow D(x) = F, \forall x \in X_A \quad (15)$$

$$D(x_{t-1}) = N \wedge D(x_{t+n+1}) = F \Rightarrow D(x) = A, \forall x \in X_A \quad (16)$$

$$D(x_{t-1}) = F \wedge D(x_{t+n+1}) = N \Rightarrow D(x) = A, \forall x \in X_A \quad (17)$$

Where $\wedge$ refers to And logical operation.

Rule 16 signifies that the fault has occurred somewhere in control mode 1 where its consequences on the pitch system dynamical behavior can be observed. Rule 17 indicates that the failure has disappeared in the control mode 1 either because of maintenance actions or because the fault is intermittent.

6.4 Drift monitoring and interpretation

The key problem of drift monitoring is to distinguish between variations due to stochastic perturbations and variations caused by unexpected changes in a system's state. If the sequence of observations is noisy, it may contain some inconsistent observations or measurements errors (outliers) that are random and may never appear again. Therefore, it is reasonable to monitor a system and to process observations within time windows in order to average and reduce the noise influence. Moreover, the information about possible structural changes within time windows can be interpreted and processed more easily. As a result, a more reliable classifier update can be achieved by monitoring within time windows. The latter must include enough of patterns representing the drift.

To distinguish the useful patterns, the pitch sensor dynamics are represented by two different control modes. In the control mode 2, the degradation consequences of pitch sensor can be observed. Therefore, all patterns in this mode are useful to be analyzed and to be included in the drift time window. In the control mode 1, the degradation consequences are masked. Patterns representing normal operating conditions cannot be distinguished from patterns representing pitch sensor degradations. Therefore in this case, no decision (normal/drift) will be taken in order to avoid integrating in the drift time window useless patterns.

The proposed scheme makes use of classes parameters (Mean, Variance-covariance matrix) which are dynamically

updated at each time but only with the patterns belonging to control mode 2. Drift indicators are defined based on these parameters and the detection of faults inception will be made based on their values. We define two drift indicators $I_{h1}(x), I_{h2}(x)$ as follows:

$$I_{h_1}(x) = d_{Mah}(C_N, \mu_e) \tag{18}$$

$$I_{h_2}(x) = d_E(\mu_N, \mu_e) \tag{19}$$

Where d_{Mah} and d_E are, respectively, the Mahalanobis and Euclidean metrics.

Euclidean metric computes the distance between the center μ_n of the normal class C_N and the center μ_e of evolving class C_e; on the other side Mahalanobis metric computes the distance between the normal class C_N and the evolving class center μ_e. Therefore, these two distances are calculated as follows:

$$d_{Mah}(C_N, \mu_e) = \sqrt{(\mu_N - \mu_e) \quad \Sigma_N^{-1} \quad (\mu_N - \mu_e)^T} \tag{20}$$

$$d_E(\mu_N, \mu_e) = \sqrt{(\mu_N - \mu_e) \times (\mu_N - \mu_e)^T} \tag{21}$$

The drift is detected when the Mahalanobis indicator $I_{h_1}(x)$, defined by Equation 18, exceeds a certain threshold th_d:

$$I_{\mathrm{h1}}(x) > th_d \Rightarrow \text{drift is detected} \tag{22}$$

After the drift detection, the drift is confirmed when Euclidean indicator $I_{h2}(x)$ defined by Equation 19, exceeds th_d as follows:

$$I_{\mathrm{h2}}(x) > th_d \Rightarrow \text{drift is confirmed} \tag{23}$$

The selection of th_d is motivated statically by taking three σ (standard deviations) of the data in the normal operating conditions.

In the case of pitch sensor faults, three scenarios may appear in the sensor feature space: fault impacting sensor 1 (β_{m1}), fault impacting sensor 2 (β_{m2}) or fault impacting both sensors (β_{m1} and β_{m2}) at the same time. The direction of the evolving class in the sensor feature space depends on which of these scenarios happened. Therefore, for sensor fault isolation, we use a drift direction indicator in order to monitor the direction of the evolving class. This will allow to determine which of these three scenarios happened and hence to isolate the abnormal drift source. When drift occurs, the evolving class will migrate from normal operating condition to failure. The direction indicator Dr and direction isolation DI are used to isolate the sensor which caused the drift-like fault. The idea is to consider the angle θ_1 respectively θ_2, between the vector μ_e relating the center of the evolving class and the origin of the feature space, and the vector μ_{e1} respectively μ_{e2} relating the origin with the projection of the center of the evolving class according to feature 1 respectively feature 2, of the feature space. These angles define the movement direction of the evolving class.

In order to calculate θ_1 and θ_2, the scalar products between $\overrightarrow{\mu_{e1}}$ and $\overrightarrow{\mu_e}$ and between $\overrightarrow{\mu_{e2}}$ and $\overrightarrow{\mu_e}$ are calculated as follows:

$$\overrightarrow{\mu_e}(x) \cdot \overrightarrow{\mu_{e1}}(x) = \|\mu_e(x)\| \cdot \|\mu_{e1}(x)\| \cdot \cos\theta_1 \tag{24}$$

$$\overrightarrow{\mu_e}(x) \cdot \overrightarrow{\mu_{e2}}(x) = \|\mu_e(x)\| \cdot \|\mu_{e2}(x)\| \cdot \cos\theta_2 \tag{25}$$

If the drift is detected and confirmed by the two drift indicators $I_{h1}(x)$ and $I_{h2}(x)$, then the drift isolation (to determine if sensor 1 or sensor 2 or both is the source of this drift) is achieved as follows:

$$\text{If } Dr = \theta_1 - \theta_2 > th_a \text{ and } \theta_1 > \theta_2 \Rightarrow DI = 1 : \text{fault in sensor 1 } (\beta_{m1}) \tag{26}$$

$$\text{If } Dr = \theta_1 - \theta_2 > th_a \text{ and } \theta_2 < \theta_1 \Rightarrow DI = 2 : \text{fault in sensor 2 } (\beta_{m2}) \tag{27}$$

$$\text{If } Dr = \theta_1 - \theta_2 < th_a \Rightarrow DI = 3 : \text{fault in both sensors } (\beta_{m1} \text{and} \beta_{m2}) \tag{28}$$

where th_a is the angle threshold. th_a is defined according to the variation of patterns within the normal class C_N. Therefore, th_a is determined experimentally using the patterns belonging to C_N.

The interpretation step aims at interpreting the detected changes within the classifier parameters and structure. This interpretation is then used as a prediction about the tendency of the future development of the WT current situation. This prediction is useful to formulate a control or maintenance action.

7 Experimentation and obtained results

The failures of pitch sensors are caused by a continuous degradation of its performance over time. This degradation can be seen as a continuous drift of the normal operating conditions characteristics (normal class) of the pitch sensor. Detecting and following this drift can help to predict the occurrence of the pitch sensor failures. The two monitoring indicators defined by Equation 18 and Equation 19 are used to detect and to confirm this drift for the twenty-seven scenarios of simple and multiple drift-like fault in pitch sensors are defined in section 2.

7.1 Simple drift-like fault in sensor β_{m1}

Figure 18 and Figure 19 represent, respectively, first and second residuals used in the pitch sensor feature space in presence of an abnormal drift in pitch sensor 1, β_{m1}. We can see in the case of an abnormal drift in pitch sensor 1, β_{m1}, that only residual $\Delta\beta_{s1}$ is impacted, while residual $\Delta\beta_{s2}$ has similar behavior as the one without abnormal drift in β_{m1}.

Table 4 show the values of the drift indicators $I_{h_1}(x)$ and $I_{h_2}(x)$ for the nine defined drift-like fault scenarios. These values represent the required time (starting from the drift beginning) to detect and confirm the drift occurrence. Thus, they can be used as an evaluation criterion to measure the time delay to detect a drift before its end.

Figures 20 and 21 show the obtained results using the two drift detection indicators $I_{h_1}(x)$ and $I_{h_2}(x)$, for simple drift-like fault in pitch sensor β_{m1}. The degradation is observed when the pitch actuator operate in control mode 2, the drift like fault in pitch sensor is successfully detected by

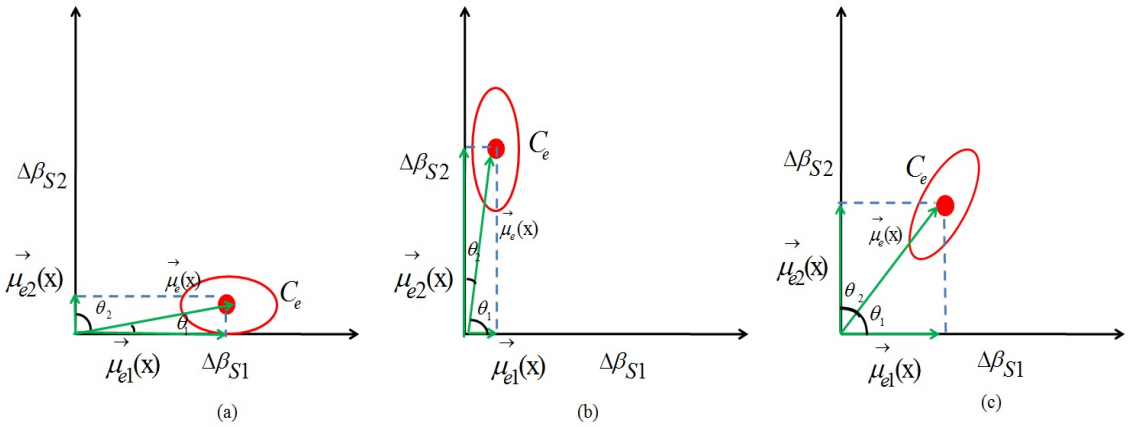

Figure 17: Drift direction angles in the pitch sensor feature space in the case of (a) simple drift-like fault in pitch sensor 1 (β_{m1}), (b) simple drift-like fault in pitch sensor 2 (β_{m2}), (c) multiple drift-like fault in both pitch sensors (β_{m1}) and (β_{m2}).

Fault N	Drift speed	I_{h1}	I_{h2}	Period
F_{4h}	30s(High) (High)	5.25s	11.00s	2500s -2730s
F_{4m}	60s(Medium) (Medium)	8.60s	18.70s	-2760s -2760s
F_{4s}	90s (Slow)	14s	26.30s	2500s -2790s
F_{5h}	30s	6.90s	13.30s	2600s -2830s
F_{5m}	60s	11.50s	20.20s	2600s -2860s
F_{5s}	90s	14.25s	27.10s	2600s -2890s
F_{6h}	30s	6.05s	11.90s	2700s -2930s
F_{6m}	60s	12.60s	23.50s	2700s -2960s
F_{6s}	90s	15.10s	29.40s	2700s -2990s

Table 4: Results of simple drift-like fault detection and confirmation in pitch sensor 1 (β_{m1}), for the nine drift scenarios.

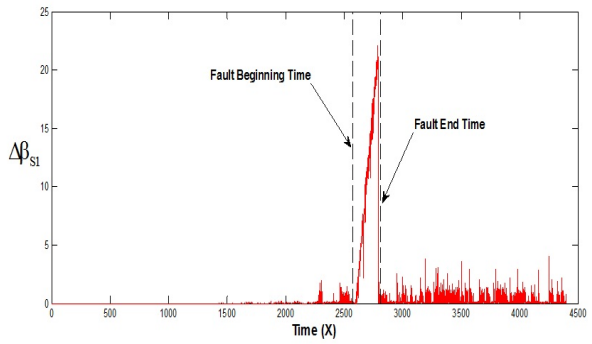

Figure 18: First residual used in the pitch sensor feature space in the case of the simple drift-like fault in pitch sensor 1 (β_{m1}).

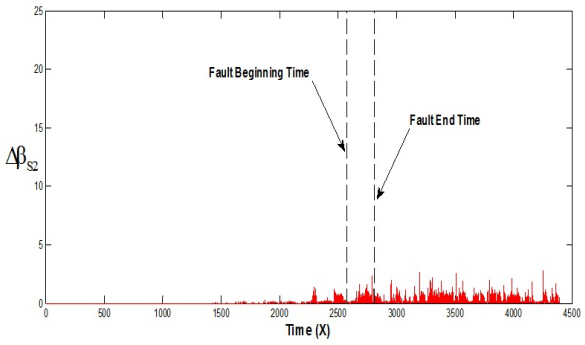

Figure 19: Second residual used in the pitch sensor feature space in the case of the simple drift-like fault in pitch sensor 1 (β_{m1}).

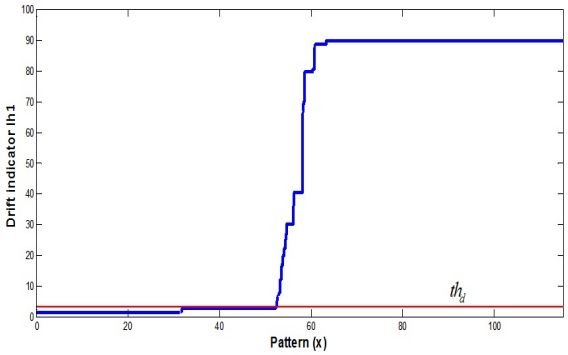

Figure 20: Drift indicator $I_{h_1}(x)$ based on Mahalanobis distance of the simple drift-like fault in pitch sensor 1 (β_{m1}).

both indicator $I_{h_1}(x)$ and $I_{h_2}(x)$, for all drift speeds (see Figure 20 and Figure 21).

The drift-like fault in pitch sensor 1 (β_{m1}), is detected in early stage before the end of this drift (arriving to the failure mode due to drift fault in pitch sensor). As an example, in the case of a drift of slow speed (F6s) (see Table 4), the

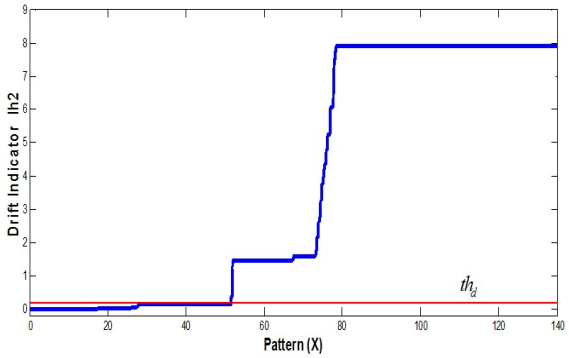

Figure 21: Drift indicator $I_{h_2}(x)$ based on Euclidean distance of the simple drift-like fault in pitch sensor 1 (β_{m1}).

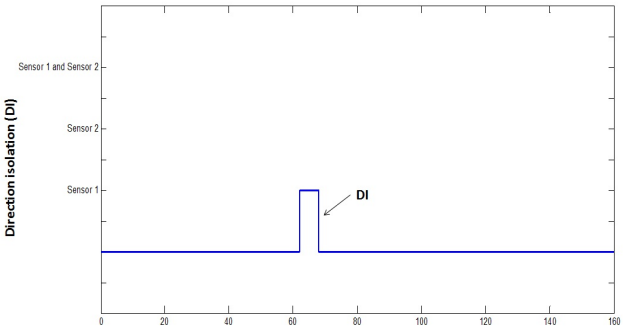

Figure 23: Direction isolation DI of the simple drift-like fault in pitch sensor 1 (β_{m1}).

pitch sensor reaches the failure mode resulting from a drift-like fault in λ_1 (degradation in λ_1) after 90 seconds of the beginning of the drift. In the proposed approach, this drift is detected 15.10 seconds and confirmed 29.40 seconds after its beginning. Therefore, the drift like fault in pitch sensor is confirmed 60 seconds before its end. This enables to achieve an early fault diagnosis and therefore helps the human operators of supervision to take efficiently the right actions.

Figure 22 and Figure 23 represent, respectively, evolving class angle and the direction indicator of the pitch sensor fault. These figures show the obtained results in presence of simple drift-like fault in pitch sensor 1, based on Figure 22 and Figure 23 the sensor 1 (β_{m1}), fault is successfully isolated by the direction indicator. Indeed, the direction angle shows that the evolving class exceeds the angle threshold (see Figure 17.a). Based on Equation 26, the drift-like fault in sensor 1 (β_{m1}), is isolated (see Figure 29).

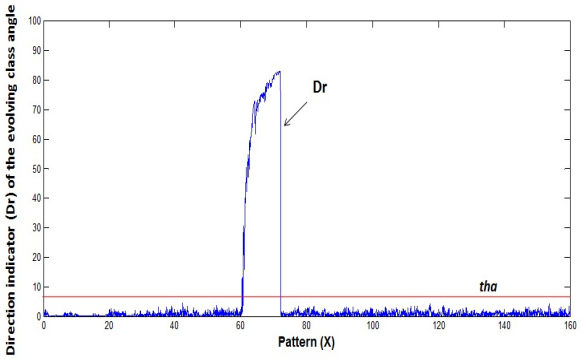

Figure 22: Direction indicator Dr of the evolving class angle of the simple drift-like fault in pitch sensor 1 (β_{m1}).

7.2 Simple drift-like fault in sensor β_{m2}

Figure 24 and Figure 25 represent, respectively, first and second residuals used in the pitch sensor feature space in presence of an abnormal drift in pitch sensor sensor 2, β_{m2}. We can see in the case of an abnormal drift in pitch sensor 2, β_{m2}, that only residual $\Delta\beta_{s2}$ is impacted, while residual $\Delta\beta_{s1}$ has similar behavior as the one without abnormal drift in β_{m2}.

Table 5 show the values of the drift indicators $I_{h_1}(x)$ and $I_{h_2}(x)$ for the nine defined drift-like fault scenarios. These values represent the required time (starting from the drift beginning) to detect and confirm the drift occurrence. Thus, they can be used as an evaluation criterion to measure the time delay to detect a drift before its end.

Fault N	Drift speed	I_{h1}	I_{h2}	Period
F_{7h}	30s (High)	6.07s	12.15s	2800s -3030s
F_{7m}	60s (Medium)	8.90s	19.05s	2800s -3060s
F_{7s}	90s (Slow)	14.20s	27s	2800s -3090s
F_{8h}	30s	5.70s	11.80s	2900s -3130s
F_{8m}	60s	8.25s	18.40s	2900s -3160s
F_{8s}	90s	13.70s	26.18s	2900s -3190s
F_{9h}	30s	6.90s	12.70s	3000s -3230s
F_{9m}	60s	9s	20.30s	3000s 3260s
F_{9s}	90s	14.90s	28.10s	3000s 3290s

Table 5: Results of simple drift-like fault detection and confirmation in pitch sensor 2(β_{m2}), for the nine drift scenarios.

Figures 26 and 27 show the obtained results using the two drift detection indicators $I_{h_1}(x)$ and $I_{h_2}(x)$, for simple drift-like fault in pitch sensor 2 (β_{m2}). The degradation is observed when the pitch actuator operate in control mode 2, the drift-like fault in pitch sensor 2 is successfully detected by both indicators $I_{h_1}(x)$ and $I_{h_2}(x)$ for all drift speeds (see Figure 26 and Figure 27).

The drift-like fault in pitch sensor 2 (β_{m2}), is detected in early stage before the end of this drift (arriving to the failure mode due to drift fault in pitch sensor). As an example, in the case of a drift of slow speed (F9s) (see Table 5), the pitch sensor reaches the failure mode resulting from a drift-like fault in λ_2 (degradation in λ_2) after 90 seconds of the beginning of the drift. In the proposed approach, this drift is detected 14.90 seconds and confirmed 28.10 seconds after

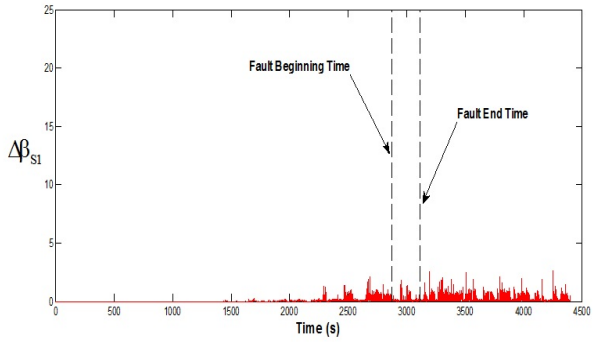

Figure 24: First residual used in the pitch sensor feature space in the case of the simple drift-like fault in pitch sensor 2 (β_{m2}).

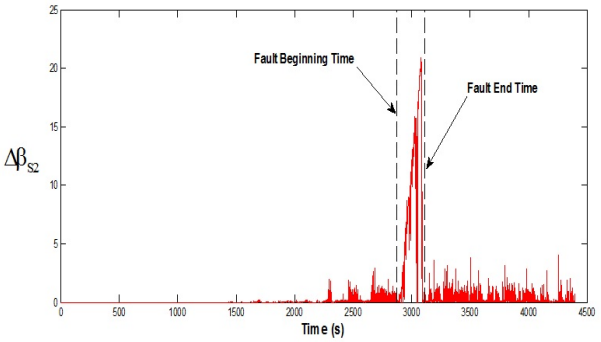

Figure 25: Second residual used in the pitch sensor feature space in the case of the simple drift-like fault in pitch sensor 2 (β_{m2}).

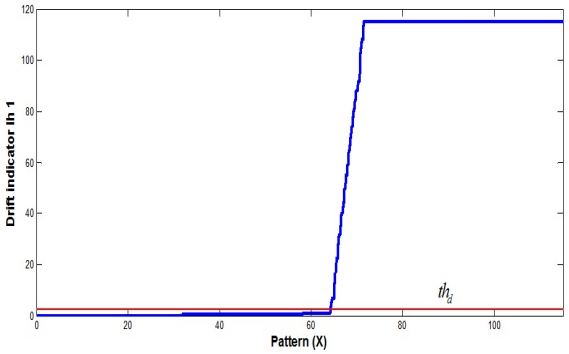

Figure 26: Drift indicator $I_{h_1}(x)$ based on Mahalanobis distance of the simple drift-like fault in pitch sensor 2 (β_{m2}).

its beginning. Therefore, the drift like fault in pitch sensor is confirmed 60 seconds before its end. This enables to achieve an early fault diagnosis and therefore helps the human operators of supervision to take efficiently the right actions.

For the drift isolation, Figure 28 and Figure 29 are used. They represent, respectively, evolving class angle and the direction indicator of the pitch sensor fault. These figures

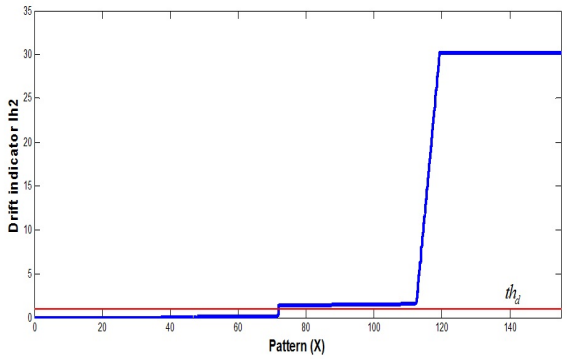

Figure 27: Drift indicator $I_{h_2}(x)$ based on Euclidean distance of the simple drift-like fault in pitch sensor 2 (β_{m2}).

show the obtained results in presence of simple drift-like fault in pitch sensor 2, based on Figure 28 and Figure 29 the sensor 2 (β_{m2}), fault is successfully isolated by the direction indicator. Indeed, the direction angle shows that the evolving class exceeds the angle threshold (see Figure 17.b). Based on Equation 27, the drift-like fault in sensor 2 (β_{m2}), is isolated (see Figure 29).

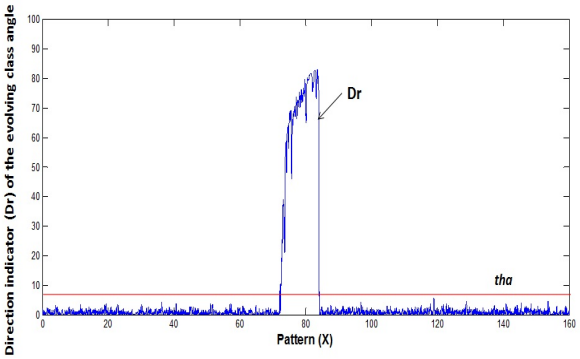

Figure 28: Direction indicator Dr of the evolving class angle of the simple drift-like fault in pitch sensor 2 (β_{m2}).

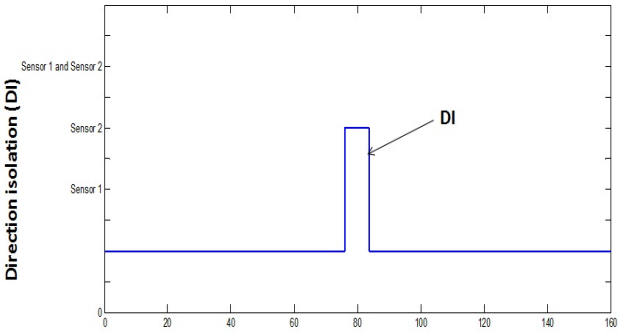

Figure 29: Direction isolation DI of the simple drift-like fault in pitch sensor 2 (β_{m2}).

7.3 Multiple drift-like fault in sensors β_{m1} and β_{m2}

Figure 30 and Figure 31 represent, respectively, first and second residuals used in the pitch sensor feature space in presence of an abnormal drift in both pitch sensor β_{m1} and β_{m2} at the same time. We can see that both residual $\Delta\beta_{s1}$ and $\Delta\beta_{s2}$ are impacted by the occurrence of the abnormal drift in β_{m1} and β_{m2}.

Table 6 show the values of the drift indicators $I_{h_1}(x)$ and $I_{h_2}(x)$ for the nine defined drift-like fault scenarios. These values represent the required time (starting from the drift beginning) to detect and confirm the drift occurrence. Thus, they can be used as an evaluation criterion to measure the time delay to detect a drift before its end.

Fault N	Drift speed	I_{h1}	I_{h2}	Period
F_{10h}	30s (High)	5.04s	10.9s	3100s -3330s
F_{10m}	60s (Medium)	9s	19.04s	3100s -3360s
F_{10s}	90s (Slow)	13.68s	26.23s	3100s -3390s
F_{11h}	30s	6.55s	15.50s	3200s -3430s
F_{11m}	60s	10.05s	19.30s	3200s -3460s
F_{11s}	90s	13.80s	27.50s	3200s -3490s
F_{12h}	30s	7.10s	16.10s	3300s -3530s
F_{12m}	60s	9.55s	22.80s	3300s -3560s
F_{12s}	90s	14.70s	28.25s	3300s -3590s

Table 6: Results of multiple drift-like fault detection and confirmation in pitch sensor 1 (β_{m1}), and pitch sensor 2 (β_{m2}), for the nine drift scenarios.

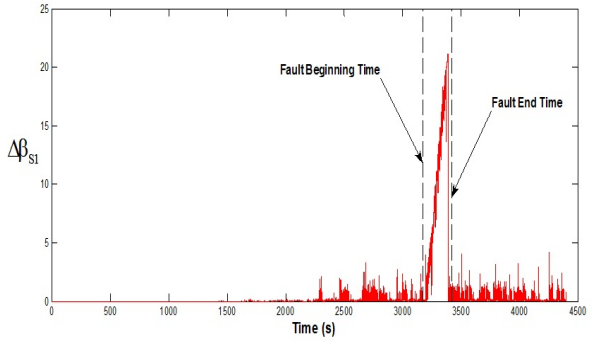

Figure 30: First residual used in the pitch sensor feature space in the case of the multiple drift-like fault in pitch sensor 1 (β_{m1}), and sensor 2 (β_{m2}).

Figures 32 and 33 show the obtained results using the two drift detection indicators $I_{h_1}(x)$ and $I_{h_2}(x)$, for multiple pitch sensor fault. The degradation is observed when the pitch actuator operate in control mode 2. The drift like

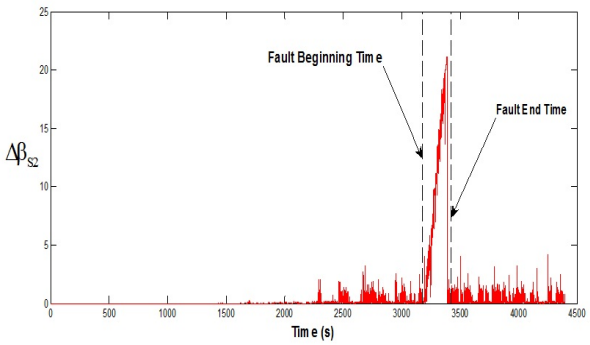

Figure 31: Second residual used in the pitch sensor feature space in the case of the multiple drift-like fault in pitch sensor 1 (β_{m1}), and sensor 2 (β_{m2}).

fault in pitch sensor is successfully detected by both indicator $I_{h_1}(x)$ and $I_{h_2}(x)$ for all drift speeds in both sensors (see Figure 32 and Figure 33).

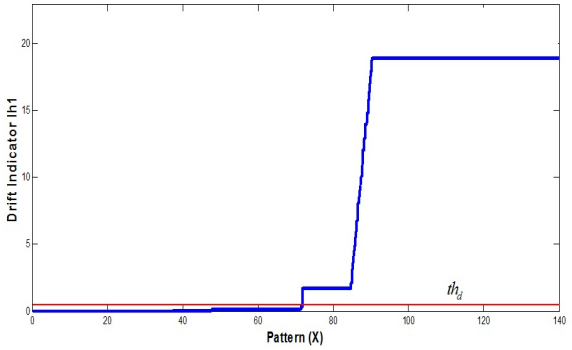

Figure 32: Drift indicator $I_{h_1}(x)$ based on Mahalanobis distance of the multiple drift-like fault in both pitch sensor 1 (β_{m1}), and sensor 2 (β_{m2}).

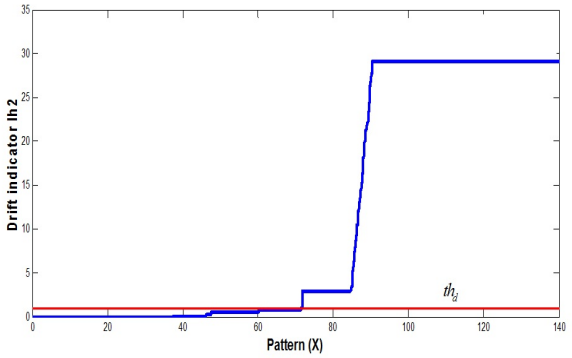

Figure 33: Drift indicator $I_{h_2}(x)$ based on Euclidean distance of the multiple drift-like fault in both pitch sensor 1 (β_{m1}), and sensor 2 (β_{m2}).

The multiple drift-like faults in pitch sensors are detected in early stage before the end of these drifts (arriving to the failure mode due to drift fault in both pitch sensors). As an

example, in the case of a drift of slow speed (F12s) (see Table 6), the pitch sensors reache the failure mode resulting from a drift-like fault in λ_1 and λ_2 (degradation in λ_1 and λ_2) after 90 seconds of the beginning of the drift. In the proposed approach, this drift is detected 14.70 seconds and confirmed 28.25 seconds after its beginning. Therefore, the multiple drift-like fault in pitch sensor is confirmed 60 seconds before its end. This enables to achieve an early fault diagnosis and therefore helps the human operators of supervision to take efficiently the right actions.

For the drift isolation, Figure 34 and Figure 35 are used. They represent, respectively, evolving class angle and the direction indicator of the pitch sensor fault. These figures show the obtained results in presence of a multiple drift-like fault in both pitch sensors β_{m1} and β_{m2}, as we can see in Figure 34 and Figure 35 the fault is successfully isolated by the direction indicator. Indeed, the direction angle shows that the evolving class evolve within the axe of the normal class (see Figure 17.c). Based on Equation 27, the multiple drift-like isolation in both pitch sensors is isolated (see Figure 35).

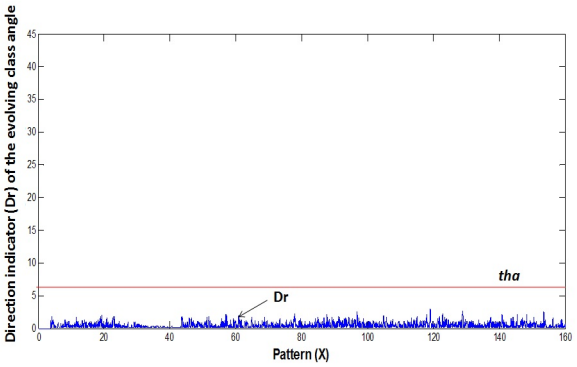

Figure 34: Direction indicator Dr of the evolving class angle of the multiple drift-like fault in both pitch sensor 1 (β_{m1}), and pitch sensor 2 (β_{m2}).

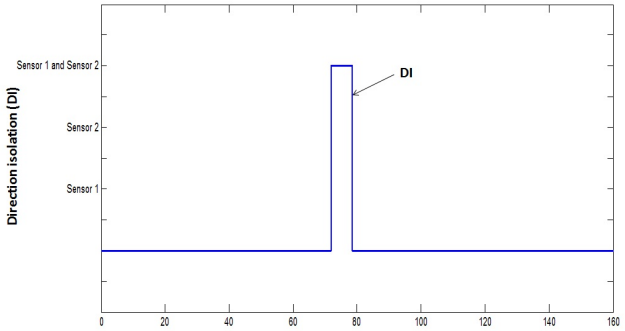

Figure 35: Direction isolation DI of the multiple drift-like fault in both pitch sensor 1 (β_{m1}), and sensor 2 (β_{m2}).

8 CONCLUSIONS

In this paper, an approach of condition monitoring and drift-like fault detection was developed. It is based on the use of a classifier able to achieve a reliable drift monitoring and early diagnosis of simple and multiple pitch sensors faults. This approach considers the system switching between several control modes. This approach based on the monitoring of the drift of the characteristics of classes representing the normal operating conditions of pitch system in each control mode. These characteristics are described by the mean and variance covariance matrix of these classes. They are monitored using two indicators in order to monitor and follow the drift. Both are defined based on the computation of the distance between the class representing normal operating conditions and the evolving class. The first indicator is based on the Mahalanobis distance and is used to detect the drift; while the second indicator is based on Euclidean distance and is used to confirm the drift. The drift indicators have detected successfully all drift scenarios of three speeds in early stage before the end of this drift for the case of simple and multiple drift-like faults in pitch system.

Future work will focus on the drift like fault of other wind turbine critical components as the generator and drive train as well as the use of other indicators to detect drifts of other types or natures.

References

[1] Bouthaina Abichou, Diana Flórez, Moamar Sayed-Mouchaweh, Houari Toubakh, Bruno François, and Nicolas Girard. Fault diagnosis methods for wind turbines health monitoring: a review. In *European Conference of the Prognostics and Health Management Society*, 2014.

[2] J.C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 1981.

[3] W. Chen, S. X. Ding, A. Haghani, A. Naik, A. Q. Khan, and S. Yin. Observer-based fdi schemes for wind turbine benchmark. pages 7073–7078, 2011.

[4] MA Djeziri, H Toubakh, and M Ouladsine. Fault prognosis based on fault reconstruction: Application to a mechatronic system. In *Systems and Control (ICSC), 2013 3rd International Conference on*, pages 383–388. IEEE, 2013.

[5] D.Kolev, P.P. Angelov, G. Markarian, M.Suvorov, and S.Lysanov. Arfa: Automated real-time flight data analysis using evolving clustering, classifiers and recursive density estimation. In *Evolving and Adaptive Intelligent Systems (EAIS), 2013 IEEE Conference on*, pages 91–97, April 2013.

[6] J. M. P. PÃl'rez M. Papaelias F. P. G. MÃąrquez, A. M. Tobias. Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46:169–178, 2012.

[7] Fredrik Gustafsson and Fredrik Gustafsson. *Adaptive filtering and change detection*, volume 1. Wiley New York, 2000.

[8] R. Hallouzi. *Multiple-model based diagnosis for adaptive fault-tolerant control*. TU Delft, Delft University of Technology, 2008.

[9] A. Kusiak and W. Li. The prediction and diagnosis of wind turbine faults. *Renewable Energy*, 36:16–23, 2011.

[10] A. Kusiak and A. Verma. A data-mining approach to monitoring wind turbines. *Sustainable Energy, IEEE Transactions on*, 3(1):150–157, Jan 2012.

[11] N. Laouti, S. Othman, M. Alamir, and N.Sheibat-Othman. Combination of model-based observer and support vector machines for fault detection of wind turbines. *International Journal of Automation and Computing*, 11:274–287, 2015.

[12] X. Luo and X. Huang. Fault diagnosis of wind turbine based on elmd and fcm. *The Open Mechanical Engineering Journal*, 8:716–720, year= 2014,.

[13] M.Traore, E. Duviella, and S. Lecoeuche. Comparison of two prognosis methods based on neuro fuzzy inference system and clustering neural network. In *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 91–97, 2009.

[14] P. F. Odgaard and J. Stoustrup. Unknown input observer based scheme for detecting faults in a wind turbine converter. pages 161–166, 2009.

[15] P.F. Odgaard, J. Stoustrup, and M. Kinnaert. Fault-tolerant control of wind turbines: A benchmark model. *Control Systems Technology, IEEE Transactions on*, 21(4):1168–1182, July 2013.

[16] Ahmet Arda Ozdemir, Peter Seiler, and Gary J Balas. Wind turbine fault detection using counter-based residual thresholding. *IFAC Proceedings Volumes*, 44(1):8289–8294, 2011.

[17] R. Precup, P. Angelov, B. S. Jales Costa, and M. Sayed-Mouchaweh. An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. *Computers in Industry*, 2015.

[18] S. Simani, S. Farsoni, and P. Castaldi. Residual generator fuzzy identification for wind turbine benchmark fault diagnosis. *machines*, 2:275–298, 2014.

[19] Silvio Simani, Paolo Castaldi, and Marcello Bonfe. Hybrid model–based fault detection of wind turbine sensors. *IFAC Proceedings Volumes*, 44(1):7061–7066, 2011.

[20] S. Tabatabaeipour, P.F. Odgaard, T. Bak, and J. Stoustrup. Fault detection of wind turbines with uncertain parameters: a set-membership approach. *Energies*, 5(7):2424–2448, 2012.

[21] C. Tsai, C. Hsieh, and S. Huang. Enhancement of damage-detection of wind turbine blades via cwt-based approaches. *Energy Conversion, IEEE Transactions on*, 21(3):776–781, Sept 2006.

[22] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin. A review of process fault detection and diagnosis part iii: Process history based methods. *Computers and Chemical Engineering*, 27:327–346, 2003.

[23] S.Zhao R.M. Ferrari M.M. Polycarpou X.Zhang, Q. Zhang and T. Parisini. Fault detection and isolation of the wind turbine benchmark: An estimation-based approach. In *Proceedings of IFAC world congress*, volume 2, pages 8295–8300, 2011.

[24] W. Yang, R. Court, and J. Jiang. Wind turbine condition monitoring by the approach of scada data analysis. *Renewable Energy*, 53:365–376, 2013.

[25] X. Youa and W. Zhangb. Fault diagnosis of frequency converter in wind power system based on som neural network. *Procedia Engineering*, 29:3132âĂŞ3136, 2012.

[26] Y. Zhi-Ling, W. Bin, D. Xing-Hui, and L. Hao. Expert system of fault diagnosis for gear box in wind turbine. *Systems Engineering Procedia*, 4:189–195, 2012.