2013

Wind Turbine Fault Detection and Isolation Using Support Vector Machine and a Residual-Based Method

Jianwu Zeng  
University of Nebraska-Lincoln, jzeng@huskers.unl.edu

Dingguo Lu  
University of Nebraska-Lincoln, stan1860@huskers.unl.edu

Yue Zhao  
University of Nebraska-Lincoln

Zhe Zhang  
University of Nebraska-Lincoln

Wei Qiao  
University of Nebraska-Lincoln, wqiao@engr.unl.edu

See next page for additional authors

Follow this and additional works at: http://digitalcommons.unl.edu/electricalengineeringfacpub

Part of the Computer Engineering Commons, and the Electrical and Computer Engineering Commons

Zeng, Jianwu; Lu, Dingguo; Zhao, Yue; Zhang, Zhe; Qiao, Wei; and Gong, Xiang, "Wind Turbine Fault Detection and Isolation Using Support Vector Machine and a Residual-Based Method" (2013). Faculty Publications from the Department of Electrical and Computer Engineering. 329.  
http://digitalcommons.unl.edu/electricalengineeringfacpub/329
Authors
Jianwu Zeng, Dingguo Lu, Yue Zhao, Zhe Zhang, Wei Qiao, and Xiang Gong
Abstract—This paper proposes a novel scheme combining support vector machines (SVM) and a residual-based method for wind turbine fault detection and isolation (FDI). SVMs with radius basis function kernels are used for detecting and identifying sensor stuck and offset faults, where binary codes of fault types are used as the outputs of the SVMs to minimize the number of SVMs being used. The same output of a SVM may correspond to different types of faults and the final decision is made by all SVMs instead of one SVM. Moreover, a residual-based fault detection method using a time-variant threshold is developed to identify the abrupt change and scaling faults. Monte Carlo simulations are carried out in MATLAB to test the effectiveness and robustness of the proposed FDI methods using a wind turbine FDI benchmark model. Results show that the proposed methods can always detect the faults successfully within the required time limits.

I. INTRODUCTION

As the number of wind turbines continuously grows, fault detection and isolation (FDI) has become a more and more important and urgent issue in modern wind turbine operation, where control systems play a vital role in satisfying power capture and load alleviation [1]. Sensors and actuators are key components in a wind turbine control system. A faulty sensor or actuator may cause process performance degradation, process shutdown, or even a fatal accident. Early FDI can provide necessary information for the control system and, therefore, helps reduce the cost of wind energy and increase penetration of wind power into electrical grids.

The purpose of fault detection is to generate symptoms, which indicate the difference between nominal and faulty conditions. Fault isolation is then performed to localize the fault and to identify the fault type based on the observed analytical and heuristic symptoms [2].

Existing FDI techniques can be broadly classified into two major categories, including model-based methods and signal processing-based methods [3]. For model-based FDI, the system model could be mathematical- or knowledge-based [4]. Faults are detected and isolated based on the residual generated by state variable or model parameter estimation [5]-[8]. For signal processing-based FDI, mathematical or statistical operations [9]-[11] are performed on the measurements, or artificial intelligence (AI) techniques [12], [13] are applied to the measurements to extract the information about the faults.

Among all the AI techniques, support vector machine (SVM) is a widely used method in binary classification application [14]. Recently, SVM has been utilized to detect and isolate sensor and actuator faults in wind turbine control system [12], [13]. However, in these papers, one SVM was used to detect each type of fault. Therefore, \( n \) SVMs were required in order to detect \( n \) types of faults. This requires significant computational load and increases the complexity of the problem.

This paper proposes a method of using SVMs for FDI of wind turbines. To reduce the number of SVMs, binary codes of fault types are generated as the output of the SVMs. The final decision is then made by all SVMs instead of one SVM for fault detection. Furthermore, a residual-based fault detection (RFD) method is proposed as well. Compared to other residual-based methods, the threshold in the proposed method is a time-variant variable rather than a constant. The detection law is also more general than those counter-based methods since it can be used for signals in different forms, e.g., different orders of derivatives.

II. FAULT DESCRIPTION AND ANALYSIS

This paper investigates two categories of faults, i.e., sensor faults and actuator faults, addressed in the second challenge call on wind turbine fault detection and fault tolerant control [15]. Table I shows the 10 fault scenarios in [15].

A. Sensor Faults

Sensor faults (i.e., Faults 1-6) include the faults in sensor measurements that are stuck, scaled from the true values, or offset from the true values. The details are listed as follows:

Fault 1: the blade root bending moment sensor measurement at Blade 2 (\( M_{BL2} \)) is scaled by a factor of 0.95.

Fault 2: an offset of -0.5 m/s\(^2\) on the tower top accelerometer in both the fore-aft (\( x_\) \)) and side-to-side (\( y_\) \)) directions.

Fault 3: the generator speed sensor (\( \omega_g \)) is scaled by a factor of 0.95.

Fault 4: Blade 1 has a stuck pitch angle (\( \beta_1 \)) sensor, which holds a constant value of one degree.

Fault 5: the generator power sensor (\( P_g \)) is scaled with a factor of 1.1.
**TABLE I: FAULT SCENARIOS.**

<table>
<thead>
<tr>
<th>No.</th>
<th>Time Range (s)</th>
<th>Fault Component</th>
<th>Fault Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[20, 45]</td>
<td>Blade root bending moment sensor ($M_b$)</td>
<td>Scaling</td>
</tr>
<tr>
<td>2</td>
<td>[75, 100]</td>
<td>Tower top accelerometer ($\dot{X}_t$)</td>
<td>Offset</td>
</tr>
<tr>
<td>3</td>
<td>[130, 155]</td>
<td>Generator speed sensor ($\omega_g$)</td>
<td>Scaling</td>
</tr>
<tr>
<td>4</td>
<td>[185, 210]</td>
<td>Pitch angle sensor ($\beta_i$)</td>
<td>Stuck</td>
</tr>
<tr>
<td>5</td>
<td>[240, 265]</td>
<td>Generator power sensor ($P_g$)</td>
<td>Scaling</td>
</tr>
<tr>
<td>6</td>
<td>[295, 320]</td>
<td>Low speed shaft position encoder ($\phi$)</td>
<td>Bit error</td>
</tr>
<tr>
<td>7</td>
<td>[350, 410]</td>
<td>Pitch actuator ($\beta_1$, $\beta_2$, $\beta_3$)</td>
<td>Slow change in dynamics</td>
</tr>
<tr>
<td>8</td>
<td>[440, 465]</td>
<td>Pitch actuator ($\beta_1$, $\beta_2$, $\beta_3$)</td>
<td>Abrupt change in dynamics</td>
</tr>
<tr>
<td>9</td>
<td>[495, 520]</td>
<td>Pitch actuator ($\beta_1$, $\beta_2$, $\beta_3$)</td>
<td>Abrupt change in dynamics</td>
</tr>
<tr>
<td>10</td>
<td>[550, 575]</td>
<td>Yaw drive ($\Omega_e$)</td>
<td>Stuck</td>
</tr>
</tbody>
</table>

**B. Actuator Faults**

Faults 7-10 belong to actuator faults, which include the faults of the blade pitch actuators, the generator torque converter, and the yaw drive.

Fault 7: slow change in Blade 2 pitch angle dynamics, which is introduced linearly from 350 s to 370 s, fully active from 370 s to 390 s, and linearly outpaced from 390 s to 410 s.

Fault 8: abrupt change in Blade 3 pitch actuator, which is active from 440 s to 465 s, and linearly introduced and outpaced within one second.

Fault 9: offset of 1000 Nm on the generated generator torque ($T_g$).

Fault 10: yaw actuator is stuck at zero rad/s.

**C. Fault Analysis**

Consider a sensor dataset $X$ consisting of $m$ variables. In this paper, $X^{(i)}$ is the $i^{th}$ ($i = 1, \cdots, m$) column of the matrix $X$ and represents a time series of the $i^{th}$ variable. $X^{(i)}(t)$ represents the value of $X^{(i)}$ at time $t$. The features used for FDI are the $q^{th}$ ($q = 0, \cdots, 3$) order derivatives of the sensor data calculated as follows:

$$X_q^{(i)}(t) = \frac{d^q}{dt^q} X^{(i)}(t)$$

(1)

where $X_q$ represents the features used for FDI. Particularly, when $q = 0$, $X_q^{(i)} = X^{(i)}$.

Let $Y \in \{+1, -1\}$ be a matrix with $n$ columns that represents the state of the system. The positive and negative values of $Y$ indicate that the system is in fault or normal state, respectively. $Y^{(j)}$ is the $j^{th}$ ($j = 1, \cdots, n$) column of the matrix $Y$ and $Y^{(j)}(t)$ represents the system state of the $j^{th}$ fault at time $t$. Then the FDI problem becomes how to construct the system fault matrix $\hat{Y}$ with the use of features $X_q$.

**III. FAULT DETECTION AND ISOLATION**

Two techniques are proposed for FDI in this paper, including a SVM method and an RFD method. The SVM is used for detecting the Faults 2, 4, 6 and 10, while the rest faults are detected by the RFD method.

**A. Support Vector Machine**

SVM has been successfully applied to classification problems, especially in binary classification applications. In this paper, SVM is used for FDI with the use of the measured signals. Since the ranges of different signals are different, they are normalized as follows:

$$\bar{X}_q^{(i)}(t) = \frac{X_q^{(i)}(t) - \mu_{X_q^{(i)}}}{\sigma_{X_q^{(i)}}}$$

(2)

where $\mu_{X_q^{(i)}}$ and $\sigma_{X_q^{(i)}}$ are the mean and standard deviation of $X_q^{(i)}$, respectively. Then a classifier is constructed as follows:

$$\hat{Y}^{(j)}(t) = \text{sgn}(w^T \cdot \varphi(x_i) + b)$$

(3)

where $\hat{Y}^{(j)}(t)$ represents the detection result of the $j^{th}$ fault at time $t$; $\text{sgn}(\cdot)$ is the sign function; $w$ and $b$ are the weights and bias of SVM, respectively; $\varphi(\cdot)$ is a nonlinear mapping function; $x_i = [\bar{X}_q^{(i)}(t), \bar{X}_q^{(i)}(t-1), \cdots, \bar{X}_q^{(i)}(t-\delta + 1)]$ is the current and historical values of the time series $\bar{X}_q^{(i)}$, where $\delta$ represents the dimension of the vector $x_i$.

The key issue is to find the optimal values of the SVM parameters $w$ and $b$. This can be done by solving the following constrained optimization problem.

$$\min \frac{1}{2} w^T w + \gamma \sum_{i=1}^{N} \xi^2(t)$$

$$\text{s.t.} \begin{cases} Y^{(j)}(t) \cdot \text{sgn}(w^T \varphi(x_i) + b) = 1 - \xi(t) \\ \xi(t) \geq 0, \quad t = 1, 2, \cdots, N \end{cases}$$

(4)

where $Y^{(j)}(t)$ is the observed value of $\hat{Y}^{(j)}(t)$; $\xi(t)$ is a slack variable; $\gamma$ is a regularization parameter, which balances the fitting in the training stage and generalization in the implementation stage. Equation (4) can be solved by using Lagrange multipliers and the solution is expressed in its dual form. Then (3) can be rewritten as follows:

$$\hat{Y}^{(j)}(t) = \sum_{k=1}^{N} \alpha_k K(x_k, x_k) + b$$

(5)

where $\alpha_k$ ($k = 1, \cdots, N$) is the nonnegative Lagrange multiplier of (3); and $K(x_k, x_k) = \varphi(x_k)\varphi(x_k)$ is a positive-definite radial basis function (RBF) kernel function defined as follows:

$$K(x_k, x_k) = \exp\left(-\frac{||x_k - x_k||^2}{\sigma^2}\right)$$

(6)

where $||\cdot||$ represents the operation of the Frobenius norm and $\sigma$ is the width of the RBF kernel.

In order to identify multiple faults, the SVM-based method is usually implemented by combining several SVMs with either a one-versus-all or one-versus-one method [1], [16]. Therefore, $n$ SVMs are required for identification of $n$ types of faults.
In this paper, Fault 10 is detected by using a SVM (SVM\textsubscript{10}), Faults 2, 4, and 6 are detected using two SVMs (i.e., SVM\textsubscript{24} and SVM\textsubscript{26}). The outputs of the two SVMs are binary coded and listed in Table II. As shown in Table II, two SVMs are used to identify four different states of the system.

<table>
<thead>
<tr>
<th>SVM\textsubscript{24}</th>
<th>SVM\textsubscript{26}</th>
<th>Fault Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>4</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>No fault</td>
</tr>
</tbody>
</table>

### B. Residual-Based Fault Detection

To identify the pitch actuator dynamics faults, i.e., Faults 7 and 8, the value of pitch angle is firstily estimated. Then the value estimated by the following model is compared to the observed value.

\[
\hat{\beta}(t) = f(z_t) \tag{7}
\]

where \(\hat{\beta}(t)\) is the estimated value of pitch angle \(\beta(t)\); \(z_t = [\beta(t-1), \beta(t-2), \ldots, \beta(t-d), \beta(t), \beta(t-1), \ldots, \beta(t-d+1)]\) is the time series that consists of the historically observed values of \(\beta\) and \(\beta_t\), where \(\beta_t\) is the reference pitch angle; \(f(\cdot)\) is the estimation function, which is realized by a three-layer artificial neural network (ANN). The number of hidden neurons is 5 and the logistic function is chosen as the activation function for the ANN.

The ANN model is trained to estimate the value of the pitch angle as expressed by (7). Then the residual, i.e., error, can be calculated as follows:

\[
r_q(t) = \beta(t) - \hat{\beta}(t) \tag{8}
\]

which is used as the feature for fault detection. If \(|r_q(t)|\) is larger than a threshold value, \(\sigma_{t_r}(t)\), then it indicates that a fault occurs. The threshold is calculated as follows.

\[
\sigma_{t_r}(t) = k \cdot \sigma_{t_{r1}}(t) = k \cdot \left( \frac{1}{d} \sum_{i=1}^{d} (r_q(i) - \mu_{t_r}(i)) \right)^2 \tag{9}
\]

where \(\mu_{t_r}(i)\) and \(\sigma_{t_r}(i)\) are the mean value and the standard deviation of \(r_q(t); k > 1\) is a parameter to specify the distance between the threshold value and the standard deviation. It should be noted that \(r_q(t) = X_q(t)\) if there is no estimated value of \(X_q(t)\).

However, for those abrupt change faults or sensor scaling faults, the residual value only exceeds the threshold value at the instant when the fault occurs and disappears. Fig. 1 shows the values of \(r_1(t)\) and \(r_2(t)\) when the 4th fault occurs.

As shown in Fig. 1, the fault occurs and disappears at \(t_1\) and \(t_2\), respectively. The values of \(r_1(t)\) and \(r_2(t)\) exceed their corresponding threshold values in a short period, e.g., \(T_s\) or \(2T_s\), where \(T_s\) is the sampling period of \(r_1(t)\) and \(r_2(t)\). Taking \(r_1(t)\) and \(r_2(t)\) into account, a new fault detection law can be written as follows:

\[
Y^J(t) = Y^J(t-1) \cdot \text{sgn}(r_q(t) - \mu_{t_r}(i) + \sigma_{t_r}(t)) \tag{10}
\]

where \(Y^J(t-1)\) is the latest value of the \(j\)th fault; \(\text{sgn}(\cdot)\) is the sign function. Obviously, (10) is suitable for fault detection with both \(r_1(t)\) and \(r_2(t)\) signals. Therefore, (10) can be used as a general law for fault detection.

Based on the fault analysis and the proposed two methods, the input feature(s) and the corresponding method used for detection of each fault are listed in Table III.

### Table III: Features and Methods for FDI

<table>
<thead>
<tr>
<th>No.</th>
<th>Inputs</th>
<th>Dimension (d)</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 4</td>
<td>(d(\xi)/dt, \beta)</td>
<td>5</td>
<td>SVM\textsubscript{24}</td>
</tr>
<tr>
<td>3</td>
<td>(\dot{\alpha}_g)</td>
<td>1</td>
<td>RFD</td>
</tr>
<tr>
<td>5</td>
<td>(P_g)</td>
<td>1</td>
<td>RFD</td>
</tr>
<tr>
<td>2, 6</td>
<td>(d(\xi)/dt, \phi)</td>
<td>1</td>
<td>SVM\textsubscript{26}</td>
</tr>
<tr>
<td>7, 8</td>
<td>(\beta_1, \beta_2, \beta_3)</td>
<td>1</td>
<td>RFD</td>
</tr>
<tr>
<td>9</td>
<td>(\beta_0, \beta_2, \beta_3)</td>
<td>1</td>
<td>RFD</td>
</tr>
<tr>
<td>10</td>
<td>(\ddot{\xi}_g)</td>
<td>20</td>
<td>SVM\textsubscript{10}</td>
</tr>
</tbody>
</table>

### IV. Simulation Results

Simulation studies are carried out in MATLAB environment to obtain the FDI results. The simulation step is set as \(T_s\), which equals the sampling time of the sensor measurements.

#### A. SVM-Based FDI Results

As listed in Table III, Faults 2, 4, 6, and 10 are detected by SVMs. For Faults 2, 4, and 6, the inputs of SVM\textsubscript{24} and
Fig. 2. The SVM-based FDI results: (a) Fault 2; (b) Fault 4; (c) Fault 6; (d) Fault 10.

Fig. 3. RFD-based FDI results: (a) Fault 7; (b) Fault 8.
detected within $2T_s$. Such a quick detection is largely attributed to the ANN-based estimation model, which is sensitive to the change of the state value.

For Faults 3, 5, and 9, residuals are calculated from the original signals because there are no estimated values provided by the reference models. As listed in Table III, the 2nd order derivative of the generator speed ($\dot{\omega}_g$) and generator power ($\dot{P}_g$) are used for detecting Faults 3 and 5, respectively. For Fault 9, both the 1st and 2nd order derivatives of generator power are utilized for fault detection. The values of the parameter $k$ are set to be 8, 20, and 5 for Faults 3, 5, and 9, respectively. Fig. 4 shows the detection results for these faults. As shown in Fig. 4 (a)-(b), Faults 3 and 5 are detected within $T_s$ and $2T_s$, respectively. Fault 9 is detected within $3T_s$, as shown in Fig. 4(c).

Table IV summarizes the detection time for each fault. It shows that the detection times of all nine faults are less than the corresponding required time limits.

C. Monte Carlo Simulation

Monte Carlo simulation has been applied to test the robustness of the proposed FDI methods. In the Monte Carlo simulation, the system is simulated 100 times where different measurement noise is generated for different simulation runs. Fig. 5 shows a typical result in a Monte Carlo simulation for each fault. It shows that the fault indices (the solid lines without any marks) obtained by using the proposed methods follow closely their nominal fault vectors (the lines marked with stars). These results indicate that faults 2–10 have always been successfully detected.

V. CONCLUSION

This paper has presented a novel scheme combining SVM and RFD methods for wind turbine FDI. The binary codes of fault types have been used as the outputs of the SVMs. This has reduced the number of SVMs needed for fault detection. Moreover, a RFD method, in which the threshold value is a time-variant value instead of a constant, has been developed and validated the effectiveness of the proposed FDI methods.

REFERENCES

Fig. 5. Typical results in a Monte Carlo simulation (The curves marked with stars are the nominal fault vectors. The solid lines without any marks represent the state of the system identified by the proposed fault detection methods): (a) Faults 2-4; (b) Faults 5-7; (c) Faults 8-10.


