A real-time fault diagnosis system for high-speed power system protection based on machine learning algorithms

Elmahdi Khoudry[1], Abdelaziz Belfqih[2], Tayeb Ouaderhman[3],
Jamal Boukherouaa[4], Faissal Elmariami[5]
[1,2,4,5]Team of Electric Networks and Static Converters, Laboratory of Energy and Electrical Systems,
National Higher School of Electricity and Mechanics (ENSEM), Hassan II University, Morocco
[3]MACS Laboratory, FSAC, Hassan II University, Morocco

Article Info	ABSTRACT
Article history: Received Mar 31, 2020 Revised May 27, 2020 Accepted Jun 17, 2020 **Keywords:** Smart fault diagnosis system K-nearest neighbors Gaussian processes Traveling waves Transmission line protection	This paper puts forward a real-time smart fault diagnosis system (SFDS) intended for high-speed protection of power system transmission lines. This system is based on advanced signal processing techniques, traveling wave theory results, and machine learning algorithms. The simulation results show that the SFDS can provide an accurate internal/external fault discrimination, fault inception time estimation, fault type identification, and fault location. This paper presents also the hardware requirements and software implementation of the SFDS.

Corresponding Author:

Elmahdi Khoudry,
Laboratory of Energy and Electrical Systems,
National Higher School of Electricity and Mechanics (ENSEM),
Road El Jadida, Km 7, BP:8118, Oasis-Casablanca, Morocco.
Email: e.khoudry@gmail.com

1. INTRODUCTION

To prevent equipment damage as well as personal injury and to ensure reliable, stable, and affordable electric energy supply, extra-high voltage networks should be fitted with efficient high-speed fault protection relays and schemes. Indeed, inefficient protection relay systems disturb the regular operation of electric power systems and increase the risk of cascading blackouts [1]. Consequently, a smart fault diagnosis system should be developed to monitor and control the operation of a classical relay, or to be itself used as a smart relay. This system can also be used for the ex-post analysis of oscillographic fault records to make a report on what might have happened during the fault condition. Thus, the purpose of a smart fault diagnosis system is to provide precise information on the external/internal fault discrimination, fault inception time, fault type, and fault location.

The kinds of fault diagnosis systems that have been suggested in the literature are mainly based on classical approaches and artificial neural networks. Those that are based on classic circuit [2, 3], symmetrical theories [4-6], digital wavelet transform (DWT) [4-6], and based on artificial neural network (ANN) [8, 9]. The symmetrical theories and digital wavelet transform [4-6] require phasor estimation whose correctness depends highly on the efficiency of non-power frequency filtering [7]. The systems based on ANN [8, 9] give satisfying results when it comes to classification tasks (fault type) but bad results when it comes to regression tasks (fault location). Moreover, ANN is challenging to interpret (it is hard to understand the reasons behind the results they give), structure (there is no specific procedure to determine the architecture of artificial neural networks) and train (computational complexity increases with the number of neurons and layers).

In this paper, we propose a new smart fault diagnosis system (SFDS) for high-speed transmission line protection based on traveling wave theory, advanced signal processing techniques and supervised machine learning algorithms. More detailed precisions about the SFDS are brought in the upcoming sections that are organized as follows. The first section gives an overview of the SFDS. The second section sets the theoretical background of the different techniques that were used for the development of the SFDS. The third section describes the software implementation of the SFDS and addresses the hardware requirements that should go with it. In the next section, simulation results are analyzed and discussed. The last section concludes this paper.

2. OVERVIEW OF THE SMART FAULT DIAGNOSIS SYSTEM (SFDS)

As shown in Figure 1, the offline conception of the SFDS requires, first and foremost, the generation of two databases from the simulated power system, one containing voltage and current measurements of the steady-state condition and the other voltage and current measurements of the fault condition. From these two databases will be extracted the RMS values of the superimposed quantities to be used as inputs of the considered learning machine algorithms during the training process. The K-nearest neighbors (KNN) algorithm is intended for the classification task to identify the fault type, and the gaussian process (GP) is intended for the regression task to determine the fault location.
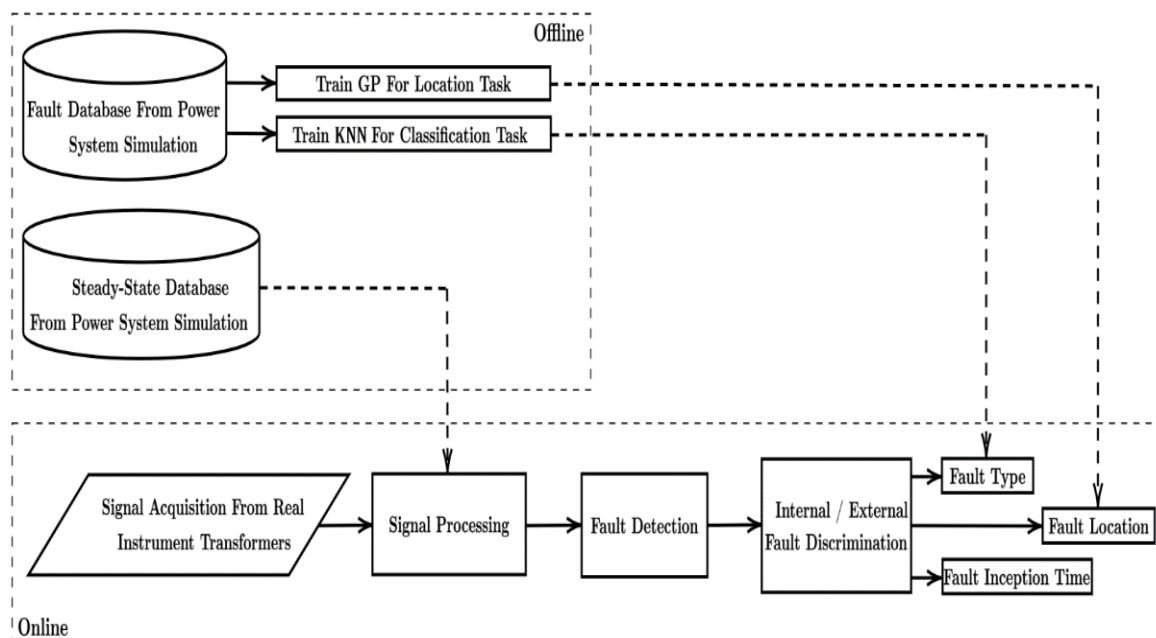

Figure 1. Overview of the smart fault diagnosis system (SFDS)

As depicted in Figure 1, the online processing starts with the acquisition of digital time-domain voltage and current measurements from instrument transformers through an A/D converter. After that, the superimposed quantities are extracted from the obtained measurements to be transformed into Clarke modal components first, then these modal components are filtered by a differentiator-smoother filter before being processed by the peak finding algorithm when a fault event is detected. Then when the peak of the filtered superimposed quantities is detected; the impedance angle is estimated from the non-filtered superimposed quantities to be used as a discriminator for internal faults. In the case where the detected fault is internal, the RMS values of the non-filtered superimposed quantities are calculated and used as inputs of the fault classifier and the fault locator. The fault inception time is estimated from the information obtained about the index of the filtered superimposed quantities peak and the equations derived from the traveling wave theory.

3. THEORETICAL BACKGROUND

The following section gives more insights into the different techniques cited in the previous section during the description of the SFDS.

3.1. Traveling waves and superimposed quantities

As shown in Figure 2, when a fault occurs on a transmission line, it produces transients that propagate at a pace approaching light speed in the form of traveling waves. As soon as the wave arrives at the measurement point, the transient voltage and current are superimposed to the steady-state (or pre-fault) voltage and current, respectively [10]. Since this theory holds only for single-phase transmission lines, its extension to three-phase transmission lines requires the transformation of the measured voltages and currents from the phase domain $(\phi = \{A, B \text{ or } C\})$ into the modal domain $(m = \{\alpha, \beta \text{ or } 0\})$. In the latter domain, each mode is independent of the other and can consequently be treated apart as a single-phase transmission line. For this purpose, we used the Clarke transform that is the most suitable for real-time applications since it uses only real numbers and not complex numbers like the Fortescue transform [11, 12].
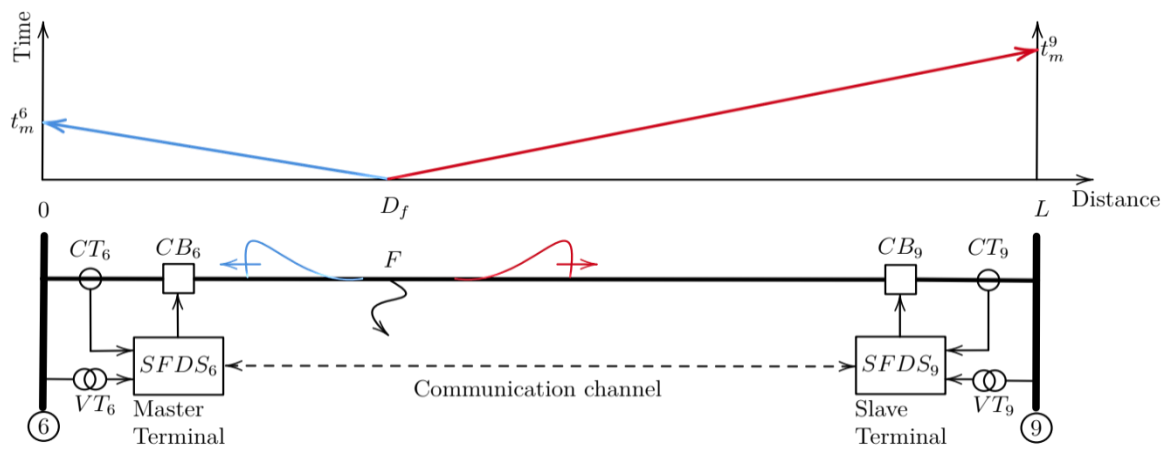

Figure 2. Traveling waves induced by a fault occurring on a line joining buses (6) and (9) of a power system

Consequently, we can derive the following expressions at buses (6) and (9):

$$\text{For } n^6 < \frac{f_s D_f}{\gamma_m} + 1, \Delta x_m^6[n^6] = 0 \tag{1}$$

$$\text{For } n^6 \geq \frac{f_s D_f}{\gamma_m} + 1, x_m^6[n^6] = \bar{x}_m^6[n^6] + \Delta x_m^6[n^6] \tag{2}$$

$$\text{For } n^9 < \frac{f_s(L-D_f)}{\gamma_m} + 1, \Delta x_m^9[n^9] = 0 \tag{3}$$

$$\text{For } n^9 \geq \frac{f_s(L-D_f)}{\gamma_m} + 1, x_m^9[n^9] = \bar{x}_m^9[n^9] + \Delta x_m^9[n^9] \tag{4}$$

where:
− b stands for bus (6) and bus (9) and m for the α-mode, the β-mode, or the 0-mode of the Clarke transform.
− γ_m represents the propagation speed of the m mode traveling wave.
− f_s represents the sampling frequency of $SFDS_b$, and $n^b = 1,2, ...$ is the $SFDS_b$ $(n^b)^{th}$ sample.
− L is the length of the line of concern and D_f is the distance from bus (6) to the fault location F.
− $x_m^b[\cdot]$ denotes the time-domain discrete modal voltages and currents measured at b.
− $\bar{x}_m^b[\cdot]$ denotes the time-domain discrete pre-fault modal voltages and currents measured at b.
− $\Delta x_m^b[\cdot]$ denotes the time-domain discrete superimposed modal voltages and currents measured at b.

The Clarke modal components of the superimposed quantities are obtained from the phase components of the measured signals and the pre-fault signals by the following expression:

$$
\begin{bmatrix} \Delta x_0^b[\cdot] \\ \Delta x_\alpha^b[\cdot] \\ \Delta x_\beta^b[\cdot] \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 2 & -1 & -1 \\ 0 & \sqrt{3} & -\sqrt{3} \end{bmatrix} \begin{bmatrix} x_A^b[\cdot] - \bar{x}_A^b[\cdot] \\ x_B^b[\cdot] - \bar{x}_B^b[\cdot] \\ x_C^b[\cdot] - \bar{x}_C^b[\cdot] \end{bmatrix}
\tag{5}
$$

Supposing that both ends are synchronized, the wave arrival times t_m^6 and t_m^9 indicated in the lattice diagram are measured with respect to the m mode, and with respect to a common time reference. Thus, we can compute the fault inception time t_f using the following relation:

$$
t_f = -\frac{L}{2\gamma_m} + \frac{t_m^6 + t_m^9}{2}
\tag{6}
$$

3.2. Differentiator-smoother filter

For the filtering process, we use a simple differentiator-smoother filter that has a square data window, as depicted in Figure 3(a) [13]. Considered over $20\ \mu s$, the abrupt change in the superimposed quantities, when the traveling wave reaches the measurement point, can be captured by this filter. In other words, this filter can capture the transition of the superimposed quantities from the zero value to the maximum non-zero value. Indeed, as shown in Figure 3(b) this differentiator-smoother filter has a triangular step response that allows the detection of the traveling wave arrival time through the time stamping of the triangle peak. As presented in Figure 3(c), we chose the amplitude of the square data window so that the amplitude of the triangular output signal matches the magnitude of the step change in the input signal.
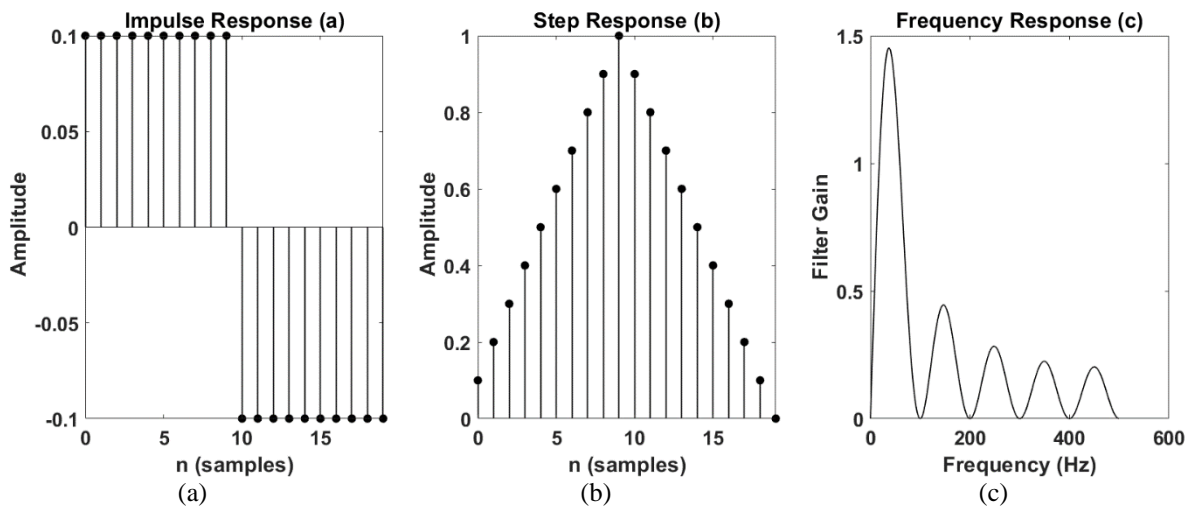

Figure 3. Square differentiator-smoother filter Impulse, Step and Frequency responses

Formally, if the modal superimposed quantity $\Delta x_m^b[\cdot]$ is the input of the square differentiator-smoother filter, $\Delta_f x_m^b[\cdot]$ its output, $h = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1]/10$ the filter impulse response and $L = 20 \cdot 10^{-6} \cdot f_s$ the square window length. We can express the relationship between these elements with respect to causality, linearity, and time-invariance conditions using the convolution product:

$$
\Delta_f x_m^b[n^b] = (\Delta x_m^b * h)[n^b] = \sum_{l=1}^{L} \Delta x_m^b[n^b - l + 1] h[l]
\tag{7}
$$

3.3. Internal/external fault discrimination criterion

According to [14, 15], when a fault occurs in the forward direction of a relay, the modal superimposed voltages $\Delta v_m^b[\cdot]$ and currents $\Delta i_m^b[\cdot]$ have different polarities. That means that if an internal fault is assumed to happen on our line of interest, we should have the following properties at buses (6) and (9).

$$\Delta v_m^6[\cdot]\Delta i_m^6[\cdot] < 0 \ and \ \Delta v_m^9[\cdot]\Delta i_m^9[\cdot] < 0 \tag{8}$$

Thus, we deduce that for a fault to be internal, the impedance angle θ_b at both ends of the line should be close to one. That is:

$$\theta_6 \sim 180° \ and \ \theta_9 \sim 180° \ \rightarrow Internal \ Fault \tag{9}$$

In real-time applications, we propose to estimate the impedance angle using the formula below that is derived from the dot product and the active power definition:

$$\hat{\theta}_b = \cos^{-1}\left(\frac{\sum_{n^b=1}^N(\Delta v_m^b-[n^b])(\Delta i_m^b-[n^b])}{\sqrt{\sum_{n^b=1}^N(\Delta v_m^b-[n^b])^2}\sqrt{\sum_{n^b=1}^N(\Delta i_m^b-[n^b])^2}}\right) \tag{10}$$

where $m^- \stackrel{\text{def}}{=} \{\alpha \ or \ \beta\}$ stands for the α-mode or the β-mode, N is the number of samples considered and $\hat{\theta}_b$ is the estimated impedance angle at bus b. We also propose for the internal/external fault discrimination the following criterion:

$$\left|\hat{\theta}_6\right| > \eta \ and \ \left|\hat{\theta}_9\right| > \eta \ \rightarrow Internal \ Fault \tag{11}$$

The angular threshold η is to be determined empirically.

3.4. K-nearest neighbors (KNN) algorithm and fault classifier

Under the assumption that similar inputs share the same properties, for a D-dimensional query input, the KNN algorithm assigns the most common label among its K most similar training inputs [16, 17]. Let $\mathbf{x}_* \in \mathbb{R}^D$ denote a D-dimensional query point, $D_{tr} = \{(\mathbf{x}^{(n)}, y^{(n)}): \mathbf{x}^{(n)} \in \mathbb{R}^D \ and \ y^{(n)} \in \{0,1\}\}_{n=1}^N$ the training dataset, $dist(\mathbf{a}, \mathbf{b})$ the distance between two D-dimensional points $\mathbf{a} = [a_1 \dots a_D]^T$ and $\mathbf{b} = [b_1 \dots b_D]^T$, T the matrix/vector transposition, and $V_{\mathbf{x}_*}$ the set of the K nearest neighbors of $\mathbf{x}_*$. We can formally define $V_{\mathbf{x}_*}$ as follows:

$$V_{\mathbf{x}_*} \subseteq D_{tr} : \ card(V_{\mathbf{x}_*}) = K \ and \ \forall(\mathbf{x}, y) \in D_{tr} \setminus V_{\mathbf{x}_*}, dist(\mathbf{x}, \mathbf{x}_*) \geq \max_{(\mathbf{x}', y') \in V_{\mathbf{x}_*}} dist(\mathbf{x}, \mathbf{x}_*) \tag{12}$$

That is, every point belonging to D_{tr} and not to $V_{\mathbf{x}_*}$ is at least far away from the query point $\mathbf{x}_*$ by a distance equal to that of the furthest point in $V_{\mathbf{x}_*}$. The classifier $h()$ can be defined as the hypothesis that returns the label of the highest occurrence in $V_{\mathbf{x}_*}$. Formally we have,

$$h(\mathbf{x}_*) = mode\left(\{y': (\mathbf{x}', y') \in V_{\mathbf{x}_*}\}\right) \tag{13}$$

where $mode()$ means to select the most frequent label. Concerning the distance function on which KNN relies, there is the Minkowski metric that is the most commonly used to reflect the closeness between a query point and the inputs of the training set, and by extension, the similarity between their labels [18]:

$$dist_p(\mathbf{a}, \mathbf{b}) = \left(\sum_{d=1}^D (a_d - b_d)^p\right)^{1/p} \tag{14}$$

As shown in Figure 4, the fault classifier is composed of two stages. The first stage contains three functions based on the KNN algorithm and each one of them indicates whether or not its corresponding phase is involved in the fault condition. The second stage indicates, based on a comparison with a threshold value to be determined empirically, whether or not the ground is involved in the fault condition. For instance, if the fault ABG happens, h_A should indicate a value of 1, h_B should indicate a value of 1, h_C should indicate a value of 0 and h_G should indicate a value of 1. In that example, the output of the fault classifier is the column vector $[1 \ 1 \ 0 \ 1]^T$.
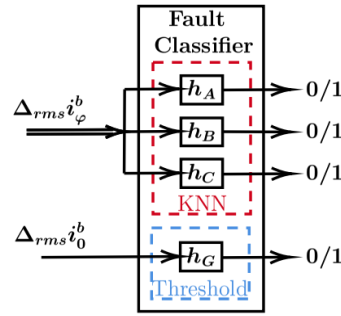
Figure 4. Fault classifier

3.5. Gaussian processes (gps) and fault Locator

Observing a training dataset $D_{tr} = \{(\mathbf{x}^{(n)}, y^{(n)}) : \mathbf{x}^{(n)} \in \mathbb{R}^D \text{ and } y^{(n)} \in \mathbb{R}\}_{n=1}^{N}$ in the context of supervised learning, we assume that $y^{(n)} = f(\mathbf{x}^{(n)}) + \epsilon$. Where $y^{(n)}$ is the noisy function evaluation of the training input $\mathbf{x}^{(n)}$, $f(\mathbf{x}^{(n)})$ the target function evaluation of $\mathbf{x}^{(n)}$, and $\epsilon \sim N(0, \sigma^2)$ is a Gaussian noise. Given a test dataset $D_{test} = \{(\mathbf{x}_*^{(n)}, y_*^{(n)}) : \mathbf{x}_*^{(n)} \in \mathbb{R}^D \text{ and } y_*^{(n)} \in \mathbb{R}\}_{n=1}^{N_*}$, we want to estimate the test outputs $\{y_*^{(n)}\}_{n=1}^{N_*}$. As part of the gaussian process (GP), we assume that the training and test outputs are drawn from the following joint Gaussian distribution [19, 20]:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0}_N \\ \mathbf{0}_{N_*} \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix}\right) = \frac{\exp\left(-\frac{1}{2}\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix}^T \boldsymbol{\Sigma}^{-1} \begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix}\right)}{(2\pi)^{\frac{N+N_*}{2}} \det(\boldsymbol{\Sigma})^{1/2}} \tag{15}$$

where $\mathbf{y}$ is the column vector $[y^{(1)} \dots y^{(N)}]^T$, $\mathbf{y}_*$ is the column vector $[y_*^{(1)} \dots y_*^{(N_*)}]^T$, $\mathbf{0}_N$ is a column vector containing N zeros, and $\mathbf{0}_{N_*}$ is a column vector containing N_* zeros. K, K_* and K_{**} are kernel matrices defined by a kernel function κ as follows: $(K)_{ij} = \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \sigma^2 \delta_{ij}$, where δ_{ij} is the Kronecker delta, $(K_*)_{ij} = \kappa(\mathbf{x}^{(i)}, \mathbf{x}_*^{(j)})$ and $(K_{**})_{ij} = \kappa(\mathbf{x}_*^{(i)}, \mathbf{x}_*^{(j)})$. The supervised learning using GPs is based on the idea that the points with similar inputs $\mathbf{x}^{(n)}$, naturally have close output values $y^{(n)}$. This similarity is expressed by the covariance function that can be defined by different kernel functions. Among the various kernels that exist, we can mention, for instance, the Rational Quadratic kernel (RQ kernel) and the squared exponential kernel (SE kernel) that are defined by equations (16) and (17), respectively:

$$\kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)} | \boldsymbol{\theta}_{RQ}) = \sigma_f^2 \exp\left(1 + \frac{dist_2(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})^2}{2\zeta l^2}\right)^{-\zeta} \tag{16}$$

$$\kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)} | \boldsymbol{\theta}_{SE}) = \sigma_f^2 \exp\left(-\frac{dist_2(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})^2}{2l^2}\right) \tag{17}$$

where $l > 0$ is the characteristic length scale, σ_f is the signal standard deviation, ζ is a positive-valued scale-mixture parameter, $\theta_1 = \ln(l)$, $\theta_2 = \ln(\sigma_f)$, $\theta_3 = \ln(\zeta)$, $\boldsymbol{\theta}_{RQ} = [\theta_1 \ \theta_2 \ \theta_3]^T$ and $\boldsymbol{\theta}_{SE} = [\theta_1 \ \theta_2]^T$.

From (15) and the rules for conditioning Gaussian distributions, it follows that the posterior density is $p(\mathbf{y}_* | \mathbf{x}_*, D_{tr}) = N(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$, where the posterior mean column vector $\boldsymbol{\mu}_* = [(\mu_*)_1 \dots (\mu_*)_{N_*}]^T$ and the posterior covariance matrix $\boldsymbol{\Sigma}_*$ are, respectively, given by the following expressions:

$$\boldsymbol{\mu}_* = K_*^T K^{-1} \mathbf{y} \tag{18}$$

$$\boldsymbol{\Sigma}_* = K_{**} - K_*^T K^{-1} K_* \tag{19}$$

The estimated test outputs are the posterior mean vector component (i.e. $\hat{y}_*^{(1)} = (\mu_*)_1, \dots, \hat{y}_*^{(N_*)} = (\mu_*)_{N_*}$).

As shown in Figure 5, the inputs of the fault locator, $\Delta_{rms}v_\phi^b$ and $\Delta_{rms}i_\phi^b$, are the root mean square values of the time-domain discrete superimposed voltages and currents $\Delta v_\phi^b[\cdot]$ and $\Delta i_\phi^b[\cdot]$ measured at b in the phase domain. The fault locator gives the estimated fault location $\hat{D}_f$ with respect to bus (6) and it is composed of ten GPs, namely $GP_{ABC}, GP_{ACG}, GP_{BCG}, GP_{ABG}, GP_{AC}, GP_{BC}, GP_{AB}, GP_{CG}, GP_{BG}$ and GP_{AG}. Each of these GPs predicts the fault location and is activated when the fault type to which it is related is involved.
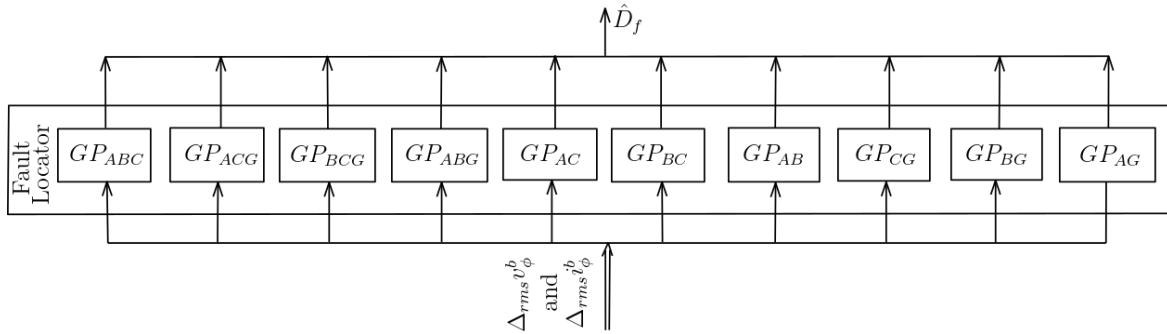

Figure 5. Fault locator

4. HARDWARE REQUIREMENTS AND SOFTWARE IMPLEMENTATION OF THE SFDS

It should be noted first that all SFDSs have a similar structure, but for illustration purposes, we will consider only the $SFDS_6$ shown in Figure 6 and its relation to the $SFDS_9$. That being said, the SFDS should meet the following main hardware requirements:

− CPU where to implement the SFDS, GPS receiver for data sampling synchronization, and high-speed communication framework as described in [21].
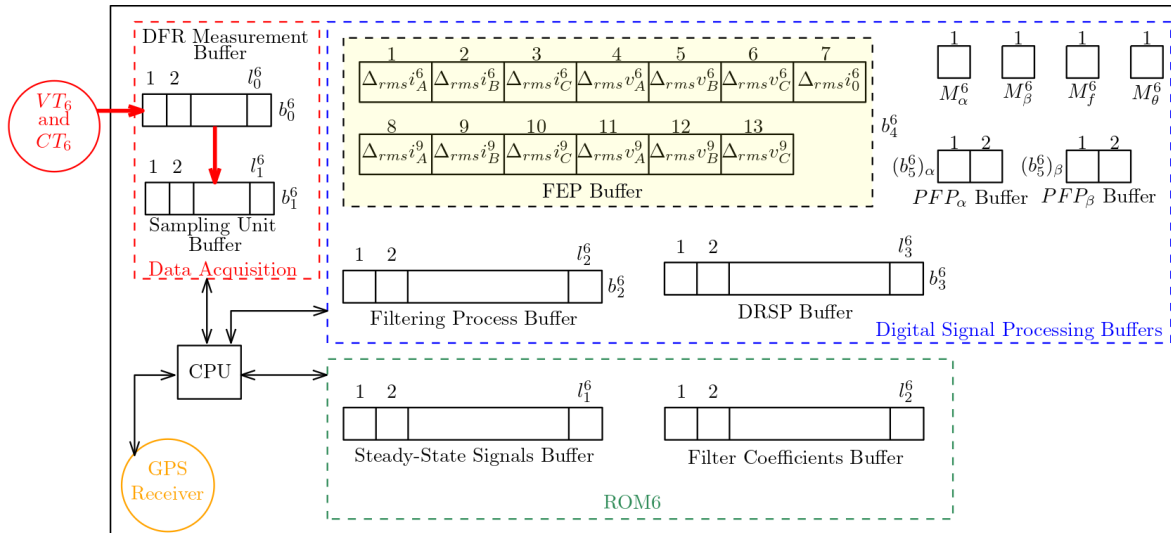

Figure 6. $SFDS_6$ hardware requirements

− Data Acquisition Buffer: We conceptually split it into the sampling unit (SU) buffer and the Data Fault Recorder (DFR) buffer. The SU buffer should contain at least six buffers $b_{1.i_A}^6$, $b_{1.i_B}^6$, $b_{1.i_C}^6$, $b_{1.v_A}^6$, $b_{1.v_B}^6$, and $b_{1.v_C}^6$, where to save the phase-domain voltages and currents $x_\phi^6 = \{i_A^6[\cdot], i_B^6[\cdot], i_C^6[\cdot], v_A^6[\cdot], v_B^6[\cdot],$ and $v_C^6[\cdot]\}_{SU}$ sampled from the DFR buffer. This latter, should also have at least six buffers $b_{0.i_A}^6$, $b_{0.i_B}^6$, $b_{0.i_C}^6$, $b_{0.v_A}^6$, $b_{0.v_B}^6$, and $b_{0.v_C}^6$, where to save the phase-domain voltages and currents $y_\phi^6 = \{i_A^6[\cdot], i_B^6[\cdot], i_C^6[\cdot], v_A^6[\cdot], v_B^6[\cdot], and v_C^6[\cdot]\}_{DFR}$ measured by CT6 and VT6. In this study, the sampling rate of the SU buffer is $f_s = 1\ MHz$, and its length is $l_1^6 = 50000$ (i.e. it can store up to 50000 samples) because we

recorded the 60 Hz steady-state signals over 3 cycles. The length of the DFR buffer should be in the order of $l_0^6 = Q \times l_1^6$, where $Q = f_a/f_s \geq 1$ and f_a is the sampling rate of the DFR buffer. It is advised to choose f_a so that Q is an integer.

− Data Signal Processing Buffer: We conceptually split it into five buffers. The Filtering Process buffer has the same structure as the SU buffer, except that the length is equal the filter window length. Since our differential-smoother filter covers a time period of $20\ \mu s$, and the SU sampling rate is $f_s = 1\ MHz$, the length of the Filtering Process buffer is then equal to $l_2^6 = 20$. The Data Recording and Saving Process buffer has also the same structure as the SU buffer, except that its length is equal to $l_3^6 = 640$. This length has been determined from the time needed for a refracted

− Wave to go back and forth between bus (6) and bus (4) at the speed of light. The Feature Extraction Process (FEP) buffer contains the RMS values of the phase-domain superimposed currents and voltages of bus (6), the RMS value of the zero superimposed current of bus (6) and the RMS values of the phase-domain superimposed currents and voltages of bus (9). The remaining buffers are used for the fault detection process.

− ROM6: During the offline conception of the SFDS, the considered power system should be simulated in normal conditions, and the phase-domain steady-state voltage and current signals should be recorded and saved in the Steady-State Signals buffer that has the same structure as the SU buffer. These signals will be used during the online process to calculate the superimposed quantities. The second buffer contains the coefficients of the differentiator-smoother filter.

      The software implementation of the SFDS is depicted in Figure 7, and commented as follows:

− Step (1): Indicates the beginning of online processing.
− Step (2): Initializes the indices k_{st}^6 and k_2^6 of the SU buffer b_1^6 and FP buffer b_2^6, respectively. It also initializes the flags p_{st}^6, fl_{int}^6 , and fl_β^6 .
− Step (3): Launches the Data Acquisition Process (DAP) that is demonstrated in Figure 8:
     − Step (3.1): Indicates the beginning of the DAP.
     − Step (3.2): Initializes the index k_0^6 of the DFR Measurement buffer (b_0^6).
     − Step (3.3): At every $T_a = 1/f_a$, new samples are measured from CT6 and VT6, and stored in their corresponding buffers in b_0^6 at position k_0^6.
     − Step (3.4): Checks whether or not k_0^6 has reached the end of b_0^6.
     − Step (3.5): Increments k_0^6 by one.
     − Step (3.6): Reinitializes k_0^6.
     − Step (3.7): Gets k_{st}^6 from steps (2), (8) or (16).
     − Step (3.8): Stores the $(Q(k_{st}^6 - 1) + 1)^{th}$ sample of b_0^6 in b_1^6 at position k_{st}^6. $Q = f_a/f_s$. This operation is called decimation and it is used for down-sampling.
     − Step (3.9): Ends the DAP, and goes to step (4)
− Step (5): Calculates the phase-domain superimposed quantities.
− Step (6): Store the phase-domain superimposed quantities in their corresponding buffers in b_2^6 at position k_2^6.
− Step (7): Checks whether or not k_2^6 has reached the end of b_2^6.
− Step (8): Increments k_{st}^6 and k_2^6 by one.
− Step (9): Applies Clarke transform to the phase-domain quantities of b_2^6 using (5).
− Step (10): Stores the convolution product, $\Delta_f i_\alpha^6$, of the α-mode superimposed current and the coefficients of the differentiator-smoother filter in buffer M_α^6.
− Step (11): Checks whether or not $\Delta_f i_\alpha^6$ has exceeded the threshold Th_α^6.
− Step (12): Stores the convolution product, $\Delta_f i_\beta^6$, of the β-mode superimposed current and the coefficients of the differentiator-smoother filter in buffer M_β^6.
− Step (13): Checks whether or not $\Delta_f i_\beta^6$ has exceeded the threshold Th_β^6.
− Step (14): Checks whether or not k_{st}^6 has reached the end of b_1^6.
− Step (15): Reinitializes k_{st}^6 and increments the flag p_{st}^6 by one.
− Step (16): Increments k_{st}^6 by one.
− Step (17): The samples at position 1 in b_2^6 are discarded, and the rest of the samples are shifted one location to the left. Position l_2^6 is now free and the different buffers of b_2^6 can receive their corresponding new samples.
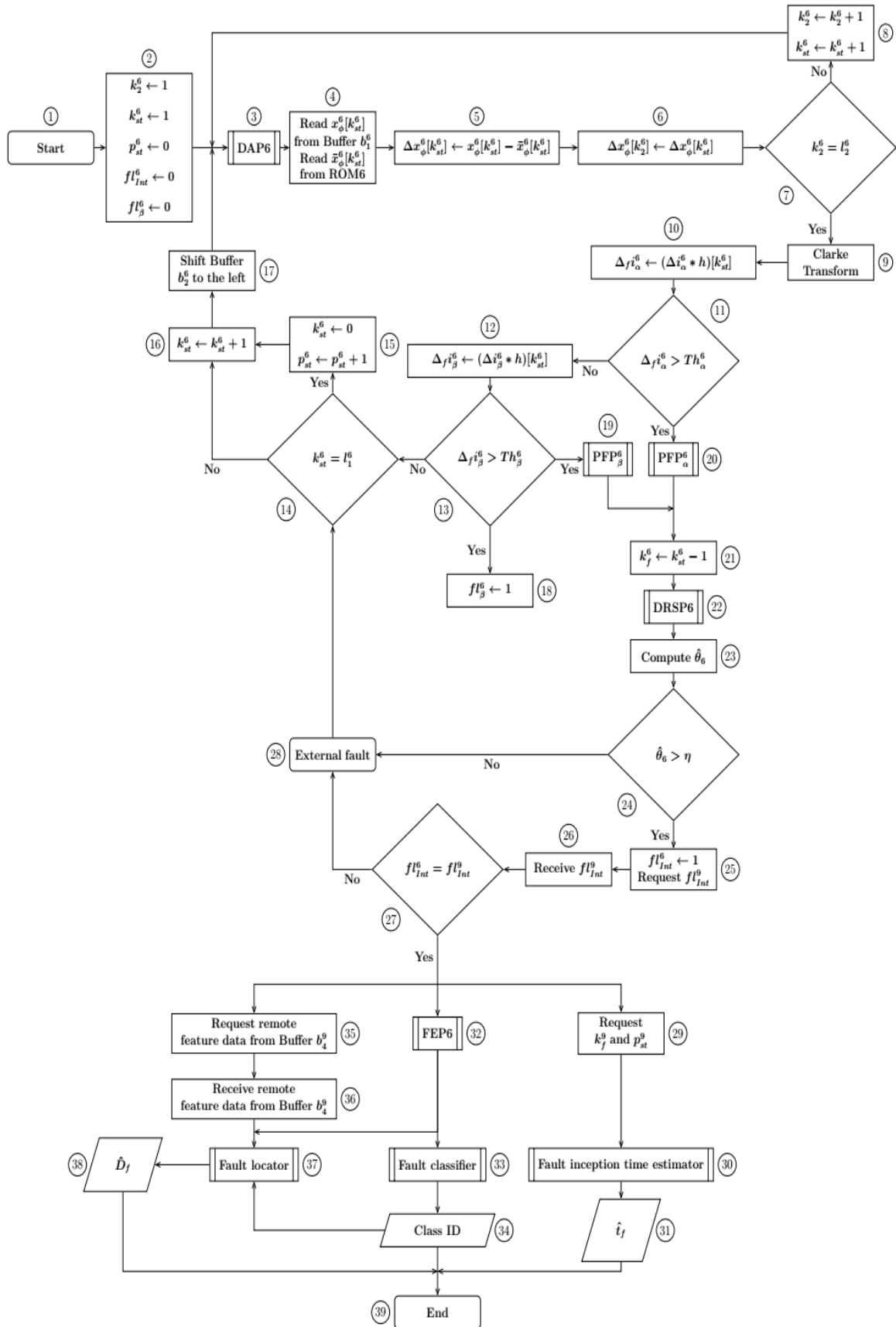− Step (18): Sets the flag fl_β^6 to one.

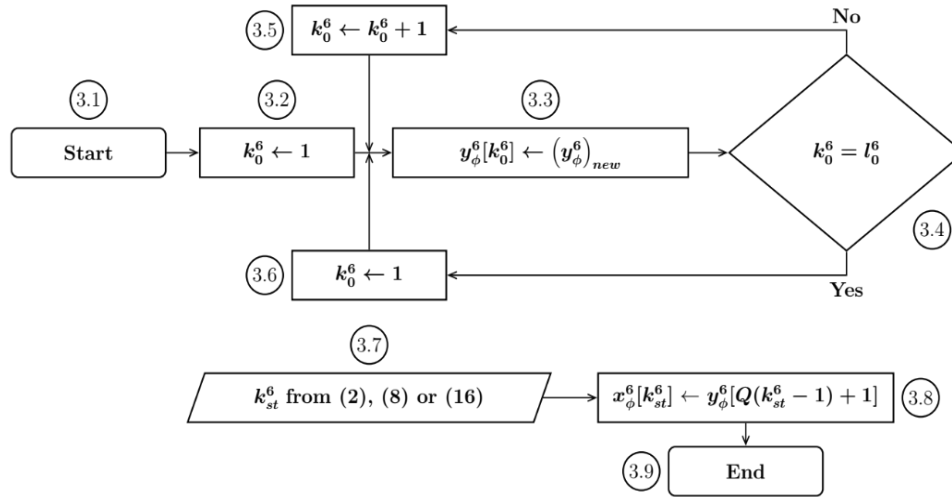Figure 7. Flowchart of the $SFDS_6$ software implementation

Figure 8. Data acquisition process (DAP6) flowchart

- Steps (19) or (20) launches the Peak Finding Process (PFP) that is shown in Figure 9:
  - Step (19/20.1): Indicates the beginning of the PFP.
  - Step (19/20.2): Initializes the index k_5^6 of the PFP_{m^-} buffer $(b_5^6)_{m^-}$. We recall that m^- stands either for α or β.
  - Step (19/20.4): Stores $\Delta_f i_{m^-}^6$ in $(b_5^6)_{m^-}$ at position k_5^6.
  - Step (19/20.5): Checks whether or not k_5^6 has exceeded one.
  - Step (19/20.6): Increments k_5^6 by one.
  - Step (19/20.7): Checks whether or not k_5^6 has exceeded one.
  - Step (19/20.8): Checks whether or not the absolute value of the new $\Delta_f i_{m^-}^6$ is greater than the absolute value of the old one.
  - Step (19/20.9): The sample at position 1 in $(b_5^6)_{m^-}$ is discarded, and the sample at position 2 is shifted to position 1. Position 2 is now free to receive the new sample.
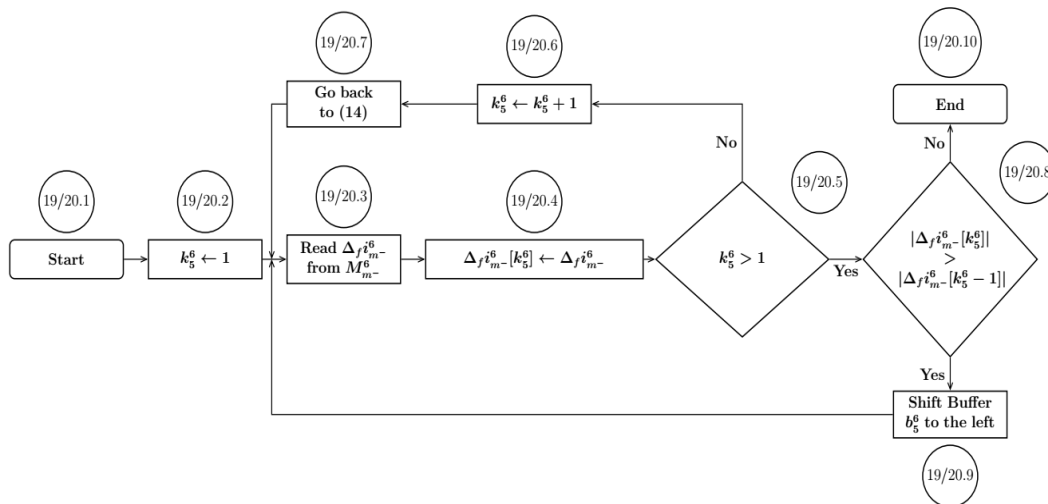 - Step (19/20.10): Ends the PFP, and goes to step (21).

Figure 9. The peak finding process (PFP) flowchart

- Step (21): Stores the index corresponding to the peak value, k_f^6, in buffer M_f.
- Step (22) launches the Data Recording and Saving Process (DRSP) that is shown in Figure 10:
  - Step (22.1): Indicates the beginning of the DRSP.

- Step (22.2): Initializes the index k_3^6 of the DRSP buffer b_3^6.
- Step (22.3): Stores the $(l_2^6)^{th}$ phase-domain samples and the $(l_2^6)^{th}$ zero superimposed current of b_2^6 in their corresponding buffers in b_3^6 at position k_3^6.
- Step (22.4): Checks whether or not k_5^6 has reached the end of b_3^6.
- Step (22.5): Increments k_3^6 by one.
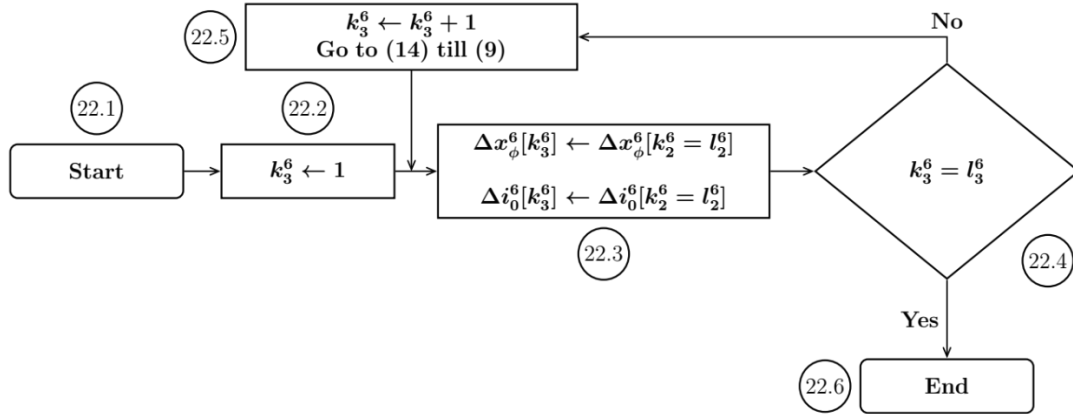- Step (22.6): Ends the DRSP, and goes to step (23).

Figure 10. The data recording and saving process flowchart

- Step (23): Calculates the estimation of the impedance angle, $\hat{\theta}_6$, using equation (10), then stores it in M_θ^6
- Step (24): Checks whether or not $\hat{\theta}_6$ has exceeded the angular threshold η.
- Step (25): Sets the flag fl_{Int}^6 to one, and requests the flag fl_{Int}^9 from $SFDS_9$.
- Step (26): Receives fl_{Int}^9 from $SFDS_9$.
- Step (27): Checks whether or not fl_{Int}^6 and fl_{Int}^9 are equal.
- Step (28): If fl_{Int}^6 and fl_{Int}^9 are not equal, then the fault is external to the line of interest.
- Step (29): Requests k_f^9 and p_{st}^9 from $SFDS_9$.
- Step (30): The fault time estimator uses the following formulas to estimate the fault inception time in step (31), where $c = 2.99792458 \cdot 10^8\ m/sec$ is the speed of light, t_f^6 is the wave arrival time at bus (6), t_f^9 is the wave arrival time at bus (9), and L is the line of interest length:

$$t_f^6 = \frac{1}{f_s}(1 - k_f^6 + p_{st}^6 l_1^6) \tag{20}$$

$$t_f^9 = \frac{1}{f_s}(1 - k_f^9 + p_{st}^9 l_1^6) \tag{21}$$

$$\hat{t}_f = \frac{t_f^6 + t_f^9}{2} - \frac{L}{2c} \tag{22}$$

- Step (32): Launches the Feature Extraction Process (FEP) that is demonstrated in Figure 11:
    - Step (32.1): Indicates the beginning of the FEP.
    - Step (32.2): Checks whether or not the flag fl_β^6 is set to one.
    - Step (32.3): If $fl_\beta^6 = 1$, then the fault class identifier is Class ID=5 (BC fault)
    - Step (32.5): Calculate and store the RMS values of the phase-domain quantities in their corresponding positions in the FEP buffer b_4^6
    - Step (32.6): Ends the FEP, and goes to step (33) where the fault classifier (see Figure 4) will identify the type of the internal fault (step (34)).
- Step (35): Requests the RMS values of the phase-domain superimposed quantities from the FEP buffer b_4^9.
- Step (36): Once received, these RMS values are stored in their corresponding position in b_4^6 (see Figure 6).

– Step (37) and step (38): The fault locator (see Figure 5) estimates from the values saved in b_4^6 the fault location $\widehat{D}_f$.
– Step (39): Ends the online process.
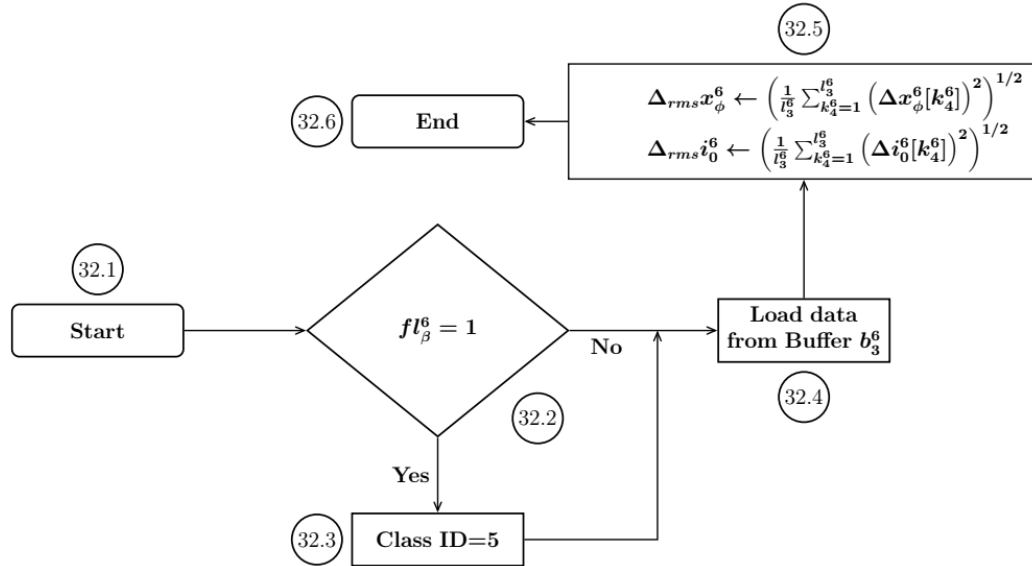It should be noted that the flags, indices and loaded data are handled by the CPU internal memory.

Figure 11. The fault extraction process (FEP) flowchart

5. SIMULATION RESULTS

In order to assess its performance, we implemented the SFDS during the simulation study of the WSCC 3-machines 9-bus test system (see Figure 14) [22], and we modeled and evaluated this latter using the Matlab/Simulink software package. Since we were concerned with the 180 Km transmission line joining bus (6) and bus (9), the training and test data contained the phase-domain voltages and currents measured by CT6, VT6, CT9, and VT9, essentially. The training and test data were randomly and uniformly generated under various fault conditions. Indeed, for all type of faults (AG, ABC), the fault resistance (in ohms) was drawn from $unif(0.001\,;40)$, the fault location (in kilometers) was drawn from $unif(1\,;179)$, and the fault inception angle (in degrees) was drawn from $unif(0\,;360)$, where $unif(a\,;b)$ stands for the uniform distribution across the real interval $[a, b]$.

To make sure that in the case of an internal fault the discriminator triggers a true alarm while considering $\eta = 90°$ and respecting the criterion given by equation (11), we evaluated the discriminator using a dataset containing about 1000 estimations of the impedance angles corresponding to buses (6) and (9). After generating the dataset $D_{Int} = \left\{\hat{\theta}_6^{(n)}, \hat{\theta}_9^{(n)}\right\}_{n=1}^{1000}$ by simulating internal faults under the conditions described in the paragraph above, we assessed the performance of the discriminator using the sensitivity indicator given by:

$$\text{Sensitivity (Discriminator)} = \frac{TP}{N} = 95.1\,\% \tag{23}$$

where TP is the number of samples that triggered a true alarm, and ($N = 1000$) is the total number of samples generated during internal fault conditions.

It appears from Figure 12 that the peak detection time at bus (6), $t_f^6 = 22.157\,ms$, is an excellent approximation of the effective wave arrival time, which is about 22.15 ms. To validate these observations for all internal fault types, we generated a test set containing 100 instances, $\left\{\hat{t}_f^{(n)}, t_f^{(n)}\right\}_{n=1}^{100}$, where $\hat{t}_f^{(n)}$ is the nth estimated Fault Inception Time (FIT) using equations (20), (21), and (22), and $t_f^{(n)}$ is its corresponding exact fault inception time. We obtained a Root Mean Square Error (RMSE) equal to:

$$RMSE_{FIT} = \sqrt{\frac{1}{100}\sum_{n=1}^{100}\left(t_f^{(n)} - \hat{t}_f^{(n)}\right)^2} = 0.0164\ ms \tag{24}$$
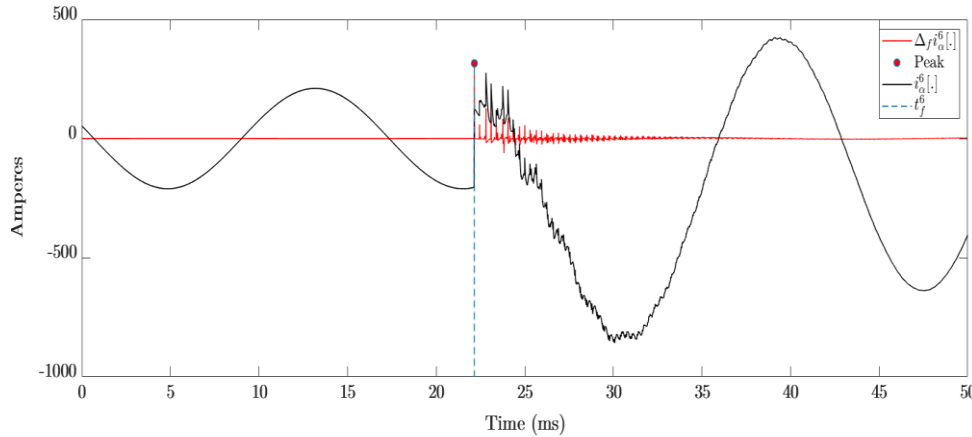

Figure 12. Time-stamping the peak corresponding to the wave arrival time at bus (6) for ABG fault

The development of the first stage of the fault classifier of the $SFDS_6$ went through the model selection process to determine the suitable input features as well as the suitable output features (labels), the number of nearest neighbors to consider, and the adequate distance metric. For this process, we used a training set of about 5000 instances and the best setting that we have found is listed in the Table 1:

Table 1. Settings of the classifiers $h_A()$, $h_b()$ and $h_c()$.

Input Features $\Delta_{rms}i_\phi^6$	$\Delta_{rms}i_A^6$, $\Delta_{rms}i_B^6$ and $\Delta_{rms}i_C^6$ as shown in Figure 4 and described in the predefined process (32).
Label y_ϕ	$y_\phi = 1$ if phase $\phi = \{A, B\ or\ C\}$ is involved, and $y_\phi = 0$ otherwise.
Number of nearest neighbors	1
Distance metric	Euclidean $(dist_2(a,b))$

As explained in section 3.4, the training set will also be used by the classification functions $h_A()$, $h_B()$, and $h_C()$ to determine the closest point among the inputs of the training set to a new test input. So to evaluate the performance of these classifiers in the context of the setting described in Table 1, we generated a test set of 1000 instances, $S_\phi = \left\{ (\Delta_{rms}i_A^6)_*^{(n)}, (\Delta_{rms}i_B^6)_*^{(n)}, (\Delta_{rms}i_C^6)_*^{(n)}, (y_\phi)_*^{(n)} \right\}_{n=1}^{1000}$, for each classification function $h_\phi()$. The performance of a classification hypothesis $h()$ is given by the accuracy, and it is defined by the following formula:

$$Accuracy(h) = \frac{TP + TN}{N} \tag{25}$$

where TP (True Positive) is the number of samples that are positive $\left((y_\phi)_* = 1\right)$ and they were classified as positive $(\hat{y}_\phi = 1)$, TN (True Negative) is the number of samples that are positive $\left((y_\phi)_* = 0\right)$ and they were classified as negative $(\hat{y}_\phi = 0)$, N is the total number of samples. So the performance of $h_A()$, $h_B()$, and $h_C()$ are as follows:

$$Accuracy(h_A) = 97.7\ \%,\ \ Accuracy(h_B) = 97.5\ \%,\ Accuracy(h_C) = 97.7\ \% \tag{26}$$

Concerning the second stage of the fault classifier of the $SFDS_6$, we have found the following relation:

$$\begin{cases} h_G(\Delta_{rms}i_0^6) = 1, & if\ \Delta_{rms}i_6^0 > 0.05\ A \\ h_G(\Delta_{rms}i_0^6) = 0, & otherwise \end{cases} \tag{27}$$

Relation (27) can also be depicted in Figure 13, where the red samples represent ground faults and the blue ones represent faults where the ground is not involved.
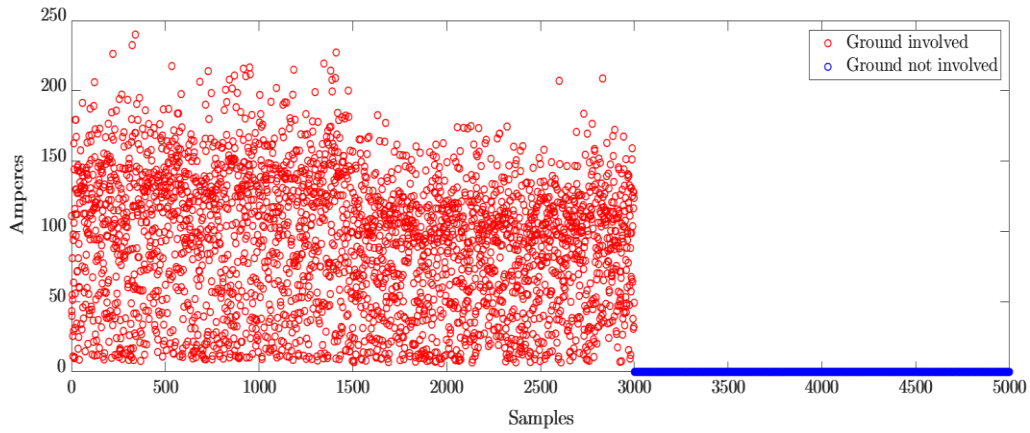

Figure 13. Samples with the ground involved vs samples with the ground not involved

In like manner, the conception of the fault locator went through a 10-folds cross validation selection process that used about 500 examples per GP to determine the most suitable input features and kernel functions for the different GPs ($GP_{AG}, GP_{BG}, \ldots, and\ GP_{ABC}$) shown in Figure 5. At the end of this process, we found that $\Delta_{rms}x_\phi^6 = \{\Delta_{rms}i_A^6[\cdot], \Delta_{rms}i_B^6[\cdot], \Delta_{rms}i_C^6[\cdot], \Delta_{rms}v_A^6[\cdot], \Delta_{rms}v_B^6[\cdot], \Delta_{rms}v_C^6[\cdot]\}_{SU}$ as well as $\Delta_{rms}x_\phi^9 = \{\Delta_{rms}i_A^9[\cdot], \Delta_{rms}i_B^9[\cdot], \Delta_{rms}i_C^9[\cdot], \Delta_{rms}v_A^9[\cdot], \Delta_{rms}v_B^9[\cdot], \Delta_{rms}v_C^9[\cdot]\}_{SU}$ were the most suitable input features (see Figure 5, and processes (32), (35) and (36)). Moreover, this model selection process showed that all GPs had the RQ kernel (see equation (16)) as the most suitable kernel function, except GP_{AB} which had the SE kernel (see equation (17)) as an appropriate kernel function. After the selection process, the training of the GP took place.

Training the GPs consisted of using the training data to tune the kernel function parameters by maximizing the marginal log likelihood using the quasi-newton optimization method [23]:

$$\ln(p(\mathbf{y}|\mathbf{X};\boldsymbol{\theta})) = -\frac{1}{2}\mathbf{y}^{\mathrm{T}}\boldsymbol{K}^{-1}\mathbf{y} - \frac{1}{2}\ln(\det(\boldsymbol{K})) - \frac{N}{2}\ln(2\pi) \tag{28}$$

where $p()$ is the density function of the multivariate Gaussian distribution, $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \ldots y^{(500)} \end{bmatrix}^T$ the column vector of the outputs (distances) of the training set, $\mathbf{X} = [\mathbf{x}^{(1)} \ldots \mathbf{x}^{(500)}]^T$ is the design matrix, the column vector $\mathbf{x} = \begin{bmatrix} \Delta_{rms}x_\phi^6 \ \Delta_{rms}x_\phi^9 \end{bmatrix}^T \in \mathbb{R}^{12}$, and $\boldsymbol{\theta}$ stands for $\boldsymbol{\theta}_{RQ}$ or $\boldsymbol{\theta}_{SE}$. In order to assess the performance of the trained GPs, we generated for each one of them a test set containing 100 instances $S_{ID} = \left\{ \mathbf{x}_{ID}^{(n)}, y_{ID}^{(n)} \right\}_{n=1}^{100}$. The ID identifies the fault type (and by extension the considered GP) as shown in Table 2. The performance of these GPs is given by the Average Relative Error (ARE) in fault location:

$$ARE(GP_{ID}) = \frac{RMSTE_{GP_{ID}}}{L} \tag{29}$$

where $MSTE_{GP_{ID}} = \sqrt{\frac{1}{100}\sum_{n=1}^{100}\left(y_{ID}^{(n)} - \hat{y}_{ID}^{(n)}\right)^2}$, and $\hat{y}_{ID}$ is the GP_{ID} estimated output (distance). The obtained results are listed in the Table 2 below.

Table 2. The ARE of the different GPs

Fault type	AG	BG	CG	AB	BC	AC	ABG	BCG	ACG	ABC
Class ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$ARE(GP_{ID})$	0.4 %	0.31 %	0.34 %	0.3 %	0.25 %	0.32 %	1.34 %	1.13 %	1.23 %	0.48 %

The values of the different indicators established in equations (23), (24), (26), and in Table 2 show clearly that the proposed smart fault diagnosis system is accurate and reliable. In fact, the value of the discriminator sensibility obtained in equation (23) shows that the SFDS is highly reliable since the percentage of internal faults that are correctly detected is 95 %. Moreover, the value of the $RMSE_{FIT}$ obtained in equation (24) indicates that the difference between the fault inception time predicted by the SFDS and the observable fault inception time is, on average, of the order of $0.0164\,ms$. This small average difference proves that the proposed SFDS estimates with high accuracy the fault inception time. Furthermore, the SFDS can correctly classify all types of faults with an accuracy of about 97.7 % according to relation (26). This result concurs with the classification accuracies stated in [24]. It should also be mentioned that the very small fault location average relative errors listed in Table 2 show that the SFDS can localize faults with high accuracy regardless of the type of fault involved. In addition, the errors listed in Table 2 are in good agreement with the fault location errors established in [24]. It should be noted that the proposed SFDS is fast since it achieves the fault classification task within $30-40\,ms$ and the tasks of fault detection, internal/external fault discrimination, and fault inception time estimation within $3-4\,ms$ [25, 26]. Furthermore, the proposed SFDS achieves the fault location task within $330-400\,ms$. When the SFDS is used in real-time, the data transmission delay between the master terminal and the slave terminal should be added to these achievement times and should remain below $10\,ms$ [27].
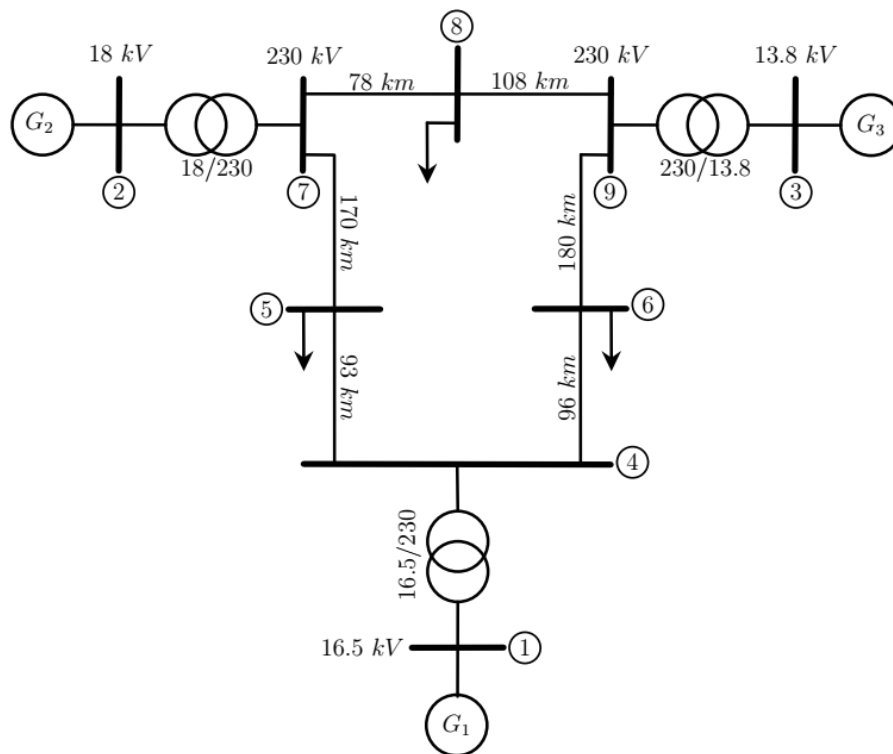

Figure 14. WSCC 3-machines 9-bus test system

6. CONCLUSION

This paper presents a smart fault diagnosis system (SFDS), which is a mixture of various and complementary time-domain techniques. Indeed, the peak detection process uses the differentiator-smoother filter. The internal/external fault discrimination process relies on the impedance angle estimation. The fault inception time estimator uses results from the traveling wave theory. The fault classifier and the fault locator are based on the KNN algorithm and the GP, respectively.

The proposed system can be used for online as well as for offline applications. The extremely low-test errors obtained in the simulation results, as well as the very short time windows used to analyze the measured signals and to extract the various input features confirm that the SFDS can provide an accurate, reliable and fast fault diagnosis.

REFERENCES

[1] P. Pourbeik, P. Kundur, and C. Taylor, "The anatomy of a power grid blackout - root causes and dynamics of recent major blackouts," *IEEE Power and Energy Magazine,* vol. 4, no. 5, pp. 22-29, 2006.

[2] H. Hong and D. Colwell, "Intelligent system identifies and locates transmission faults," *IEEE Computer Applications in Power,* vol. 10, no. 2, pp. 31-35, 1997.

[3] M. Kezunovic and B. Perunicic, "Automated transmission line fault analysis using synchronized sampling at two ends," *IEEE Transactions on Power Systems,* vol. 11, no. 1, pp. 441-447, 1996.

[4] A. E. Lazzaretti, V. H. Ferreira, H. V. Neto, L. F. R. B. Toledo, and C. L. S. Pinto, "A new approach for event detection in smart distribution oscillograph recorders," *IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America),* 2013.

[5] B. Mahamedi and J. G. Zhu, "Fault classification and faulted phase selection based on the symmetrical components of reactive power for single-circuit transmission lines," *IEEE Transactions on Power Delivery,* vol. 28, no. 4, pp. 2326-2332, 2013.

[6] Y.-H. Lin, C.-W. Liu, and C.-S. Chen, "A new PMU-based fault detection/location technique for transmission lines with consideration of arcing fault discriminationpart i: Theory and algorithms," *IEEE Transactions on Power Delivery,* vol. 19, no. 4, pp. 1587-1593, 2004.

[7] D. B. Solovev, "Selection of digital filter for microprocessor protection relays," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 8, no. 3, pp. 1498-1512, 2018.

[8] N. Saravanan and A. Rathinam, "A comparitive study on ANN based fault location and classification technique for double circuit transmission line," *Fourth International Conference on Computational Intelligence and Communication Networks,* 2012.

[9] H. Kaur and M. Kaur, "Fault classification in a transmission line using levenbergmarquardt algorithm based artificial neural network," *Advances in Intelligent Systems and Computing. Springer Singapore,* pp. 119-135, 2019.

[10] A. Johns, "New ultra-high-speed directional comparison technique for the protection of ehv transmission lines," *IEE Proceedings C (Generation, Transmission and Distribution),* vol. 127, no. 4. pp. 228-239, 1980

[11] E. Khoudry, A. Belfqih, J. Boukherouaa, and F. Elmariami, "Traveling wave based fault location for power transmission lines using morphological filters and clarke modal components," *International Journal of Electrical and Computer Engineering,* vol. 10, no. 2, pp. 1122-1134, 2020.

[12] D. E. Hedman, "Propagation on overhead transmission lines i-theory of modal analysis," *IEEE Transactions on Power Apparatus and Systems,* vol. 84, no. 3, pp. 200-205, 1965.

[13] E. O. Schweitzer, B. Kasztenny, and M. V. Mynam, "Performance of time-domain line protection elements on real-world faults," *69th Annual Conference for Protective Relay Engineers (CPRE),* 2016.

[14] E. Khoudry, A. Belfqih, T. Ouaderhman, J. Boukherouaa, and F. Faissal, "Multi-scale morphological gradient algorithm based ultra-high-speed directional transmission line protection for internal and external fault discrimination," *International Journal of Electrical & Computer Engineering,* vol. 9, no. 5, pp. 3891-3904, 2019.

[15] M. Chamia and S. Liberman, "Ultra high speed relay for EHV/UHV transmission lines-development, design and application," *IEEE Transactions on Power Apparatus and Systems,* vol. PAS-97, no. 6, pp. 2104-2116, 1978.

[16] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory,* vol. 13, no. 1, pp. 21-27, 1967.

[17] E. G. Nihad, E.-N. El Mokhtar, Z. Abdelhamid, and A. A. Mohammed, "Hybrid approach of the fuzzy c-means and the k-nearest neighbors methods during the retrieve phase of dynamic case based reasoning for personalized follow-up of learners in real time," *International Journal of Electrical & Computer Engineering,* vol. 9, no. 6, pp. 4939-4950, 2019.

[18] R. R. Isnanto, M. A. Riyadi, and M. F. Awaj, "Herb leaves recognition using gray level co-occurrence matrix and five distance-based similarity measures," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 8, no. 3, pp. 1920-1932, 2018.

[19] C. E. Rasmussen, "Gaussian processes in machine learning," *Summer School on Machine Learning,* pp. 63-71, 2003.

[20] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable gps," *IEEE Transactions on Neural Networks and Learning Systems,* 2020.

[21] E. Khoudry, A. Belfqih, J. Boukherouaa, F. Elmariami, and B. Abdelmajid, "An overview of smart grid communication networks technologies, protocols and challenges," *Proceedings of the 2nd International Conference on Networking, Information Systems & Security,* 2019.

[22] P. M. Anderson, A. A. Fouad, "Power System Control and Stability," *2nd Ed., John Wiley & Sons,* 2002.

[23] Nocedal, J. and S. J. Wright, "Numerical Optimization," *Second Edition. Springer Series in Operations Research, Springer Verlag,* 2006.

[24] Y. Q. Chen, O. Fink, and G. Sansavini, "Combined fault location and classification for power transmission lines fault diagnosis with integrated feature extraction," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 1, pp. 561-569, 2017.

[25] Z. Radojevic, C. Kim, M. Popov, G. Preston, and V. Terzija, "New approach for fault location on transmission lines not requiring line parameters," *Ínternational Conference on Power System Transients Proceedings,* pp. 28-48, 2009.

[26] A. Raza, A. Benrabah, T. Alquthami, and M. Akmal, "A review of fault diagnosing methods in power transmission systems," *Applied Sciences,* vol. 10, no. 4, 2020.

[27] V. Celebic, A. Kabovic, M. Kabovic, J. Gajica, and I. Salom, "Solutions for the alternative route of the teleprotection communication channel," *First South East European Regional CIGRÉ Conference, Portoroz,* 2016.

BIOGRAPHIES OF AUTHORS

Elmahdi Khoudry is a Ph.D. student at the National Higher School of Electricity and Mechanics (University Hassan II of Casablanca- Morocco). His research activities concern power systems protection and automation as well as Smart Grids technologies.

Abdelaziz Belfqih Professor at the National High School of Electricity and Mechanics (University Hassan II of Casablanca - Morocco). PhD, Engineer and holder of the University Habilitation searches (HDR). Head of the research team" Electrical Networks and Static Converters." Teacher researcher currently working on electricity network and smart grids.

Tayeb Ouaderhman Assistant professor at Polydisciplinary faculty of Khouribga. Sultan My Slimane University. Operation research PhD. Currently working on data science and decision aid.

Jamal Boukherouaa Professor at the National High School of Electricity and Mechanics (University Hassan II of Casablanca - Morocco). PhD, Engineer and holder of the University Habilitation searches (HDR). Head of the research team" Electrical Networks and Static Converters." Teacher researcher currently working on high frequency static converters.

Faissal Elmariami Professor at the Superior National High School of Electricity and Mechanics (University Hassan II of Casablanca - Morocco). PhD, Engineer and holder of the University Habilitation searches (HDR). Head of the research team" Electrical Networks and Static Converters." Teacher researcher currently working on stability of the electricity network and smart grids.