Risk-constrained minimisation of combined event detection and decision time for online transient stability assessment

Risk-constrained Minimisation of Combined Event Detection and Decision Time for Online Transient Stability Assessment

Jhonny Gonzalez, Panagiotis N. Papadopoulos, *Member, IEEE*, Jovica V. Milanović, *Fellow, IEEE*,
Goran Peskir, John Moriarty

***Abstract*— This paper addresses the problem of 'quickest possible' online transient stability assessment, by minimizing the decision time of combined event detection that might lead to a system split and unstable generator group prediction, from real-time wide area power system measurements. More importantly it does so by respecting predefined probabilistic error constraints for the prediction. The statistical theory of optimal detection is applied, firstly to choose the detection threshold and secondly to select a flexible assessment time, after using probabilistic neural networks to provide a temporal representation of the data. On simulated wide area measurements from the interconnected New England test system and New York power system this approach is between two and three times faster on average than strategies based on fixed assessment times, despite having comparable error rates.**

***Index Terms*—dynamic security assessment, optimal detection, phasor measurement units, probabilistic neural network, transient stability assessment, risk.**

I. Introduction

DUE to various social, environmental, economic and technical reasons, power systems may be driven to operate closer to their stability limit. New techniques and algorithms need to be implemented which will facilitate close to real-time identification of impending instability and will enable corrective control measures, such as controlled islanding, generator tripping, load shedding, and so on. Considering also the increasing uncertainty in such systems, reliance on traditional preventive control may not suffice and the focus should shift to corrective control. For effective corrective control it is essential that an action is taken as fast as possible and also that adequate information on the transient stability status of the system are available to reduce the possibility of system instability or cascading failures that could lead to blackouts [1], [2].

The combined time when a distrubance is detected and transient stability status evaluated are key to successfully address the above-mentioned problems. If the transient stability status is evaluated/predicted with a long delay, the system may be already in an unstable situation with excessive operational risk. On the other hand, if the transient stability status prediction is quick but the control is not adequate (possibly because of limited understanding or observation of the contingency) or alternatively not deployed quickly enough, system instability may not be avoided resulting again in compromised system security. This follows because in some cases and contingencies the appropriate corrective actions could be identified and taken sooner, while in the other it is more appropriate to observe the response of the system for a longer period, rather than to risk deploying a potentially inappropriate corrective measure (this may also vary depending on the complexity of the dynamic response). In addition, it is also important to provide additional flexibility to system operators to make automated decisions as soon as possible according to the desired level of risk. For example, an unstable contingency wrongly classified in real time as a stable contingency might have more significant consequences, and therefore the corresponding risk constraint could be tightened. Consequently, both the appropriate *minimisation* of the decision time is essential as well as the ability to implement predefined risk constraints rather than using fixed times.

Both the fault detection and the transient stability assessment problems have received considerable attention in the literature. This has been mainly due to the possibility of collecting real-time and near real-time data from phasor measurement units (PMUs) [3]-[5]. Methods utilizing data from dynamic state estimators have been proposed for local out-of-step protection of individual generators [7]. In addition, many alternative classification techniques have been used for global instability prediction, including decision trees [2], [3], [8]-[11], support vector machines [12]-[14] and artificial neural networks [15]-[18]. Binary classification (i.e. prediction of stable or unstable system) is most widely studied in the literature. Multiclass instability classification has also been introduced [2], [17], [19],

This work was supported by the EPSRC grant EP/I031650/1. Moriarty was supported in part by EPSRC grant EP/K00557X/2, Milanović and Papadopoulos were supported in part by the collaborative EPSRC-India project ACCEPT (grant number: EP/K036173/1). Papadopoulos is also supported in part by UKRI grant MR/S034420/1.

The first and fourth authors are with the Department of Mathematics and the third author with the Department of Electrical and Electronic Engineering at The University of Manchester, UK. The second author is with the Department of Electronic and Electrical Engineering at the University of Strathclyde. The fifth author is with the School of Mathematical Sciences, Queen Mary University of London, UK (e-mail: j.moriarty@qmul.ac.uk).

All results can be fully reproduced using the methods and data described in this paper and provided references.

[20] providing additional information considering the type of instability (e.g. groups of generators losing synchronism). In [6], recurrent neural networks are used to reduce the prediction time for binary classification. However, in all these methods the input to the machine learning proxy are post-disturbance measurements after the fault is cleared, which assumes that the time the fault happens and is cleared is known. In reality, this time when the instability prediction method should be triggered is generally unknown and can also influence the decision time if triggered with a delay, a challenge which this paper addresses. In addition, [6] and [21] are aiming to decrease decision time by evaluating the output of a recurrent neural network or an ensemble classifier at each time step and deciding to proceed with the prediction when a pre-set threshold of the classifier is exceeded. This highlights the importance of the balance between faster decision time and accuracy. However, both methods do not offer the ability to define and ensure probabilistic error constraints are met for the prediction, an aspect this paper is addressing. In addition, they address binary (stable/unstable) and not multiclass (prediction of unstable generator group) classification.

Very recently [22] and [23] highlight another important aspect related to providing some confidence in the prediction when applying machine learning based methods, further to just using performance metrics (e.g. accuracy). In particular, [22] has suggested a framework for Dynamic Security Assessment to provide a confidence interval along with the prediction and [23] a method for neural network verification (i.e., for specific input regions, ensure that the classification of the neural network does not change). In this context, the method proposed in this paper ensures predefined probabilistic error constraints are not violated while at the same time minimizing the combined event detection and prediction time.

Consequently, the key contributions of the proposed method are as follows: i) It addresses optimality in the *combined detection and prediction time* for transient stability assessment while at the same time and more importantly ii) it ensures that *probabilistic error constraints* for the prediction are not violating a pre-defined acceptable threshold. The intertwined problems of *detecting a disturbance that might lead to a system split* combined with the consequent activation of transient stability prediction from wide area measurements have not been considered simultaneously in the past. The importance of combining the two steps is to enable a realistic approach where measurement data are continuously used as an input a method for transient stability prediction without assuming explicit knowledge of the time when an event requiring the activation of the prediction algorithm has happened This is an issue that has not been addressed in existing literature (including [20] where only the prediction time is considered).

This paper is therefore addressing the significant challenge of answering when is the best time to take a decision and apply a corrective measure (e.g. controlled islanding). More importantly, the method provides the ability to a system operator to set a defined acceptable error threshold that eventually determines the optimal decision time for each specific contingency in real-time.

II. Methodology

The proposed method uses Probabilistic Neural Networks (PNNs) combined with theory of optimal detection to minimise the time required for the combined problem of disturbance and transient instability prediction. PNNs are propose to be trained offline and then used for the online (close to real time) prediction of upcoming instability as quickly as possible. This will be achieved using online measurement data and the proposed method would be utilized in order to activate corrective control measures or special protection schemes, once instability is predicted.

PNNs are a type of artificial neural network suitable for classification problems. The MATLAB function ‘newpnn’ [25] was employed, which creates a two-layer network. The first layer computes the distance from the input to the classes of training data, while the second layer converts these distances to a probability distribution for the class of the input, represented as a vector of probabilities summing to 1.

The input and training data were generated by dynamic simulation (as described in Section IV.B below). In particular, the measured responses used in the input and training data are the voltage observations V for event detection and rotor angle observations δ for the specific unstable generator group prediction. The use of voltage observations for the event detection was chosen, rather than using the rotor angles δ, because changes in electrical quantities (voltage magnitude) are observed immediately compared to the changes in rotor angles due to slower electromechanical phenomena. In a similar manner, current could also potentially be used as a variable.

The PNNs are pre-trained to be used in an online manner as measurement data is streamed to them from PMUs as part of a Wide Area Monitoring System (WAMS) or from a dynamic state estimator [7], [24]. The event detection PNN receives voltage measurements V as input and creates a process of probabilities $\rho(t)$, denoted in the paper as Process 1. Process 1 is needed to identify that a disturbance has occurred. It is worth mentioning that Process 1 is trained to identify specifically large disturbances that might potentially lead to instability and it would therefore not be expected to be activated by small voltage fluctuations. Following event identification, the transient stability analysis PNN is triggered. This receives rotor angles δ as inputs and creates a process of probability distributions (vectors) $\pi(t)$ which are denoted Process 2. Process 2 is needed for the identification of the specific instability pattern. As new measurements continue to arrive, these processes continue to be updated and consequently they have a time varying output. It should be noted that Process 1 does not aim at substituting the event detection and activation of conventional protection devices. It rather aims at triggering Process 2, that is responsible for online stability prediction. The overall method therefore, aims at activating additional (to the conventional protection) corrective measures or special protection schemes. To highlight this distinction, the reference is made to an event or a contingency detection instead of fault detection throughout this paper. Therefore, Process 1 is a distinct and necessary mechanism that in a realistic setting will continuously monitor online measurement data and automatically detect events that would require the activation of the online stability prediction algorithm. Process 1 is specifically trained to identify only such events that might lead to a system split and not merely to activation of traditional protection devices, since this would have implications on the needed training of the second stage of online transient stability assessment too.

More specifically, to account for the dynamic nature of the considered problems, separate PNNs are created for each observation time t, as follows. Regarding the disturbance detection PNN, for each simulation in the training data, the class is 1 if the actual time of the disturbance θ is smaller than the current time step t and is 0 otherwise. Thus when the event detection PNN is provided with measurement data (i.e. voltages) at a single time t, it returns a probability $\rho(t)$ that an event has occurred by time t. In the transient stability prediction PNN, for each simulation in the training data, the class is equal to the index i (with $1 \leq i \leq I$) of the instability class (as defined in Section II.A), or equal to 0 if the contingency is stable. Thus when the transient stability analysis PNN is provided with measurement data (i.e. rotor angles) at time t cycles, following event detection, it returns a probability distribution in the form of a vector $\pi(t)$. Its ith entry $\pi^i(t)$ represents the probability that the contingency will ultimately result in specific unstable grouping pattern i. Note that unlike the event detection PNN, which takes the complete data (from time 0 onwards) as input, the transient stability analysis PNN takes only the post-fault detection data. This fact makes the event detection part of the proposed methodology essential.

Representative trajectories of processes 1 and 2 are illustrated in Figure 1. When the output of Process 1 reaches a certain threshold, the occurrence of a contingency is declared and the first component of the output of Process 2 (denoted π^0) is then monitored. When in turn, the output of π^0 crosses a time-varying boundary (explained in detail in subsequent Sections) and visualised in Fig. 1, the specific instability pattern defined as the predicted unstable generator groupings, is identified. It is given by the largest of the remaining elements of Process 2 (namely π^i, $1 \leq i \leq I$, one for each instability class). This approach may be interpreted that the PNNs first provide a temporal *representation* of the data (from WAMS), then statistical optimisation theory guides the *selection* of the decision rules.

The proposed methodology described in this Section has been applied to the transient stability prediction problem. However, the method can be extended to any stability prediction algorithm (for different types of stability) in a similar manner, to optimally define the time required for a decision.

A. Problem formulation and definition of instability classes

It is assumed that at some random time θ (which is generally unknown) a transient disturbance occurs and a fixed length of time T immediately following the disturbance is then considered, by which time possible loss of synchronisation is assumed to have occurred. Hierarchical clustering is applied at time T to define unstable generator grouping patterns as in [24]. The agglomerative (bottom up) method is used with a cut-off value of 360 degrees, since exceeding this limit between any two generators points to loss of synchronism. This results in generator groupings where generators belonging to one group have less than 360 degrees difference in rotor angles. Consequently, in the case where only one group exists, the case is stable. By applying consecutively this procedure up to time T, the order of unstable groups is also derived (more details on this process can be found in [24]). An illustration of how unstable groups are defined, is presented in Fig. 2.

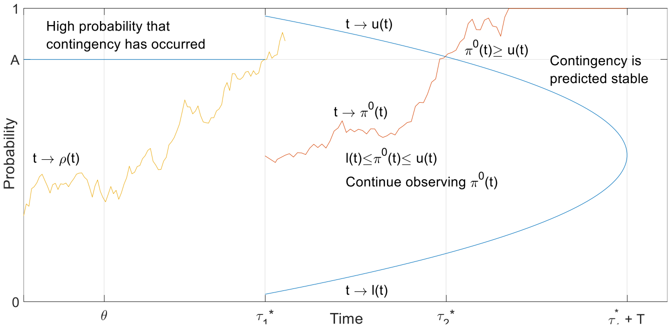

Fig. 1. Illustration of optimal decision rules for the RQP and RSP.

Following a probabilistic approach and simulating a number of contingencies (more details provided in Section IV) a list of possible unstable generator groupings is obtained and each one of them is assumed to fall into one of I different *instability classes*, represented as C={0,1,...,I}, where 0 is the stable contingency class and the positive integers are labels for the I unstable contingency classes. It should be noted that the definition of instability classes, originally introduced in [24], is provided here for completeness of discussion only and it is not a new contribution of this paper.

B. Decision rules

In this setup decision rules are selected for three variables τ_1, τ_2, d, where: 1) $\tau_1 > 0$ is the time at which a transient disturbance is detected; 2) $\tau_2 > \tau_1$ is the time at which the transient stability assessment is made; 3) $d \in C$ is the contingency class predicted at time τ_2.

Consider a power system with g generators labelled by $i = 1,\dots,g$ and denote by $V^i(t)$ the voltage at generator terminal i, and by $\delta^i(t)$ the rotor angle at generator i, at time $t \geq 0$. Additionally let us write $V(t) = (V^1(t),...,V^g(t))$ and $\delta(t) = (\delta^1(t),\dots,\delta^g(t))$, i.e. the vectors of all voltages and rotor angles respectively at time t.

To be suitable for real-time applications the decision rule must be non-anticipative. That is, τ_1 must be chosen based only upon observation of the generator voltages ($V^i(t)$) and rotor angles ($\delta^i(t)$) up to time τ_1 inclusive, and similarly for τ_2. Further, in this paper the problems of event detection and transient stability assessment are temporally separated. That is, τ_2 and d are chosen using only observations between time τ_1 and time τ_2 inclusive. This is one of the main contributions of the paper which, alongside the explicit consideration of risk constraints, differentiates the proposed method from the approach followed in most such methods (e.g. [2]-[4], [6], [8]-[11], [24]) that do not consider the event detection. The mathematical implications of, and justification for, this decomposition are discussed in

Section III.

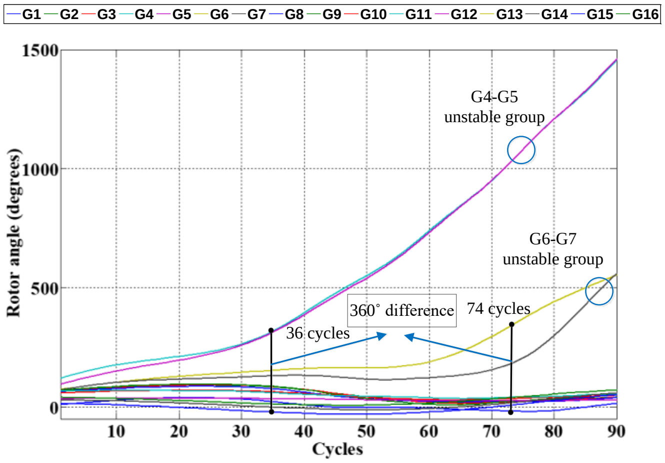

Fig. 2. Example of instability class.

C. *Objectives*

The aim in this work is to assess power system transient stability status as quickly as possible following a contingency which is not directly observable, subject to probabilistic error constraints. Decision rules for τ_1, τ_2 and d will therefore be selected which aim to minimise the average total delay

$$J=\mathbb{E}[(\tau_2-\theta)^+] \quad (1)$$

(see below for the definition of $\mathbb{E}$). Recall that θ is the actual time of the disturbance (as also shown in Fig. 1). The application of the function $(a)^+ := \max(0; a)$ in (1) ensures that any classifications occurring before the event itself do not make a negative contribution to the average in (1), thus artificially reducing the average delay, but are instead counted as zero and hence make no contribution to (1). Since corrective actions taken on the basis of an incorrect transient stability assessment may cause unnecessary operational disruption and cost, four kinds of error are considered:

- If $\tau_1 < \theta$ (a false alarm), error A_0 occurs;
- If the contingency is predicted stable ($d = 0$) when it is actually unstable of any class, error A_1 occurs;
- If the contingency is predicted unstable ($d > 0$) when it is actually stable, error A_2 occurs;
- If the contingency is predicted unstable of type $d_1 > 0$ but the contingency is actually unstable but of a different class $d_2 \neq d_1$, error A_3 occurs.

The probability p_i of error A_i for a given decision rule and dataset is interpreted simply as the proportion of samples in the dataset for which error A_i occurs under that decision rule. For example, when Process 1 identifies that a disturbance has happened (process $\rho(t)$ crossing threshold A as visualised in Fig. 1) at a given time that eventually ends up being smaller than the actual time that the disturbance has happened (θ), this counts as a case of false alarm and counts towards error A_0. Expectations, denoted by $\mathbb{E}$ (used in following part), are then simple averages over the samples in the dataset. From the definition of false alarms A_0, the maximum potential false dismissal rate (i.e. a contingency not identified) is bounded theoretically to be 1-p_0. It should be noted however, that in our test dataset the maximum detection delay time is 17 cycles, which means that all contingencies are detected and therefore the false dismissal rate is 0. This is expected as the delay in such a case of a false dismissal would be equal to the maximum delay and therefore heavily penalised during the optimisation process.

D. *Optimisation problems*

In order to address the minimisation of (1) two sub-problems are identified: first the event detection problem, then the transient stability unstable pattern identification problem.

1) Risk-constrained quickest detection problem (RQP).

Select a decision rule for τ_1 to minimise the average detection delay $J_1=\mathbb{E}[(\tau_1-\theta)^+]$, subject to the probabilistic false alarm constraint

$$\mathbb{P}[A_0] \leq p_0 \quad (2)$$

Suppose that an optimal decision rule τ_1^* has been found for the RQP problem. The second problem is stated as follows:

2) Risk-constrained sequential testing problem (RSP).

Given τ_1^* select decision rules for τ_2 and d to minimise the average prediction delay $J_2=\mathbb{E}[(\tau_2-\tau_1^*)]$, subject to the probabilistic error constraints

$$\mathbb{P}[A_1] \leq p_1 \quad (3)$$
$$\mathbb{P}[A_2] \leq p_2 \quad (4)$$
$$\mathbb{P}[A_3] \leq p_3 \quad (5)$$

Note that if no event is declared by the end of the observation period (t=90 cycles) then the event detection time τ_1 is set to be equal to the maximum, i.e., 90 cycles. Similarly, if no transient stability prediction is made then the prediction time τ_2 is set equal to 90 cycles. This incurs the maximum possible penalisation for failure of detection during optimisation.

The symbols τ_2^* and d^* are used to denote optimal decision variables for the RSP. The parameters p_0, p_1, p_2 and p_3 are probabilistic risk constraints for the corresponding types of error (A_0 through A_3). They are set according to the level of error considered acceptable while trying to balance the decision time and are a design parameter of the proposed method that would be decided by the system operator using the proposed method. Section V.A provides practical suggestions on setting the error rate bounds. A lower error threshold is usually expected to lead to slower but more accurate decisions. These parameters also provide flexibility to be strict in specific types of error only. For example, a system operator might want to set a very low value for a false alarm (threshold p_0) as it may be acceptable to happen only on very rare occasions. More information and numerical examples regarding this are provided in Section V.B and V.C.

E. *Detailed Description of Decision Rule Structure and PNN Processes*

The approach to event detection is to use Process 1 $\rho(t)$ (describing the probability that an event occurred), described in Section II. Detection occurs when this probability is sufficiently high: that is, when an upper threshold $A \in [0,1]$ is crossed by Process 1 (see also Fig. 1 for illustration).

In turn, the RSP problem is addressed through the use of Process 2 $\pi(t)$, providing a probability distribution for the transient stability status (stable/unstable and exact identification of unstable generator groups) based on the measurements at time t. The first component $\pi^0(t)$ of the vector

$\pi(t)$ provides a temporal representation of the probability that the detected contingency is stable. In the same way, for each i with $1\leq i\leq I$, the ith component $\pi^i(t)$ of the vector $\pi(t)$ provides a temporal representation of the probability that the detected contingency is unstable of type i. An important contrast with the RQP above is that the optimal 'free boundary' is time dependent in the RSP, (shown in Fig. 1). This means that the probability threshold for a very quick decision, which is necessarily based on fewer post-contingency observations, can be higher. Recall that neither the threshold for the RQP nor the boundary for the RSP are updated online as measurements are coming in, but are generated offline from the training dataset (as described in Section III below). However, adding test cases in the training dataset, for example for additional operating conditions or different topologies, could lead to an updated threshold and boundary.

Upper and lower free boundary functions, denoted u and l respectively, are optimised through dynamic programming and the use of Lagrange multipliers. If the upper boundary is crossed first by the process π^0 then stability is immediately predicted (that is, τ_2 is set equal to the time of the boundary crossing and d is set equal to 0). Conversely, if the lower boundary is crossed first by π^0 at time τ_2 then instability ($d \geq 1$) is predicted. In this case the probability processes π^i, $1\leq i\leq I$, one for each instability class, are inspected. Their maximum at time τ_2 indicates the most likely unstable class, so provides the value of d: that is, $d=j$ where $\pi^j(\tau_2) = \max_{1\leq i\leq I} \pi^i(\tau_2)$. It should be noted that the boundaries merge at some point to ensure a decision is made within a maximum time T, after a disturbance is detected (as shown in Fig. 1). Effectively this is imposing a maximum decision time.

An example of the decision rules for both the RQP and RSP is provided in Fig. 1. Here Process 1 passes above the level A at time τ_1^*, at which time an event is detected. (Note that since the actual time of the disturbance $\theta<\tau_1^*$ this is not a false alarm.) From time τ_1^* onwards the post-detection probability process π^0 is observed, and a prediction of the unstable class is made when π^0 crosses the boundary at time τ_2^*. Since in this example it is the upper boundary u which is crossed first, stability (d=0) is predicted. Alternatively had the lower boundary l been crossed first, the most likely unstable class at time τ_2^* would have been predicted.

The optimisation problem formulations introduced in Section II.D and the above decision rule structures are derived from the theory of *optimal detection* [26], which provides guarantees of optimality in problems of rapid detection under probabilistic uncertainty. The theory of *quickest detection* addresses change point detection problems, while *sequential testing* concerns the choice between two or more hypotheses as quickly as possible on average.

III. Solving the Risk-Constrained Quickest Detection and Sequential Testing Problems

In this Section, the detailed process to solve the described RQP and RSP is presented. We use the method of Lagrange multipliers to determine the optimal threshold A for the RQP and boundary $\upsilon=(l,u)$ for the RSP, as illustrated in Fig. 1. A threshold or boundary will be called *feasible* if the decision rule constructed from this threshold or boundary satisfies the risk constraints of the problem when applied to the training data. In the RQP, for each threshold A and nonnegative multiplier λ_0 the Lagrangian is

$$L_1(A;\lambda_0) = \mathbb{E}[(\tau_1-\theta)^+ + \lambda_0(\mathbf{1}_{A_0} - p_0)] \tag{6}$$

where the function $\mathbf{1}_{A_0}$ takes the value 1 if the error A_0 (as described in Section II.C) occurs and the value 0 otherwise. Similarly the same indicator function is used in (9) and (10) for the remaining A_i.

By the saddle point theorem (see for example Proposition 5.1.6 of [27]), (A^*,λ_0^*) is an optimal threshold-Lagrange multiplier pair if and only if A^* is feasible, $\lambda_0^*\geq 0$, and (A^*,λ_0^*) is a saddle point of the Lagrangian (6), that is:

$$L_1(A^*;\lambda_0) \leq L_1(A^*;\lambda_0^*) \leq L_1(A;\lambda_0^*) \tag{7}$$

for any other feasible threshold A and multiplier $\lambda_0\geq 0$. All solutions to the RQP may therefore be found by searching numerically for such pairs (A^*,λ_0^*). Similarly, the Lagrangian for the RSP is

$$L_2(\upsilon;\lambda) = \mathbb{E}[\tau_2 - \tau_1^* + \sum_{i=1}^{3} \lambda_i(\mathbf{1}_{A_i} - p_i)] \tag{8}$$

where $\upsilon = (l, u)$ jointly denotes the lower and upper boundaries described above, $\lambda = (\lambda_1, \lambda_2, \lambda_3)$, and τ_1^* is an optimal solution to the RQP. Again by the saddle point theorem, solutions to the RSP correspond to saddle points (υ,λ^*) with υ feasible and $\lambda^*\geq 0$. However since the space of all possible boundaries $\upsilon = (l,u)$ is high dimensional, for the RSP we employ a dynamic programming procedure. In particular, for each nonnegative multiplier $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ we determine an optimal boundary $\upsilon_\lambda = (l_\lambda, u_\lambda)$ as follows.

As noted above, the boundaries must merge by the terminal time T to ensure that a decision is made. We thus impose $l_\lambda(T) = u_\lambda(T) = x$ and take the value of $x \in [0, 1]$ which minimises the Lagrangian L_2 when observations are started at time T.

Then for each time t with $\tau_1^* \leq t < T$, employing backwards dynamic programming, we suppose knowledge of the boundaries $(l_\lambda(s), u_\lambda(s))$ for all times $t < s < T$. When observations are started at time t, the Lagrangian L_2 is then a function of $(x, y) = (l_\lambda(t), u_\lambda(t)) \in [0, 1] \times [0, 1]$, which we minimise over (x, y) with the condition that $x\leq y$. This procedure is continued backwards in time until $t = \tau_1^*$ to obtain the boundary functions $\upsilon_\lambda = (l_\lambda, u_\lambda)$. To search for saddle points we then maximise over the multiplier λ.

IV. Test Network and Presentation of Case Study

The case study uses the IEEE 68-bus reduced order equivalent model of the interconnected New England test system and New York power system (NETS-NYPS) as detailed in [28] and [29]. The test network, consists of 16 synchronous generators (G1-G16) in five interconnected areas. Standard sixth order models are used for all synchronous generators. G1-

G16 are equipped with either slow IEEE DC1A DC exciters or fast acting static exciters of type IEEE ST1A, and G9 is equipped with a power system stabiliser. All generators are also equipped with generic governors, representing gas, steam and hydroelectric turbines. Dynamic RMS simulations are performed using DIgSILENT/PowerFactory software.

A. *Modelling of uncertainty*

Each of the loads in the test network is varied by scaling according to a daily load curve taken from UK National Grid data [24]. The hour of day is then sampled randomly following a uniform distribution to determine the per unit (p.u.) values for all loads. An independent random scaling factor is also applied to each hour of the day and to each load, drawn from a normal distribution with mean value 1 p.u. and standard deviation 3.33% as in [2]. An optimal power flow (OPF) problem is then solved to determine the output and disconnection of synchronous generators. Three phase self-clearing faults are considered in this study as contingencies that can happen with equal probability at any point of any line. A normal distribution with mean value of 13 cycles and standard deviation of 6.67% is used to model the fault duration [2]. The actual time at which each disturbance occurs (θ) is drawn independently at random from the uniform distribution between 10 and 30 cycles. The methodology could, however, include any additional uncertain parameters, including any additional types of contingencies that can be simulated using any simulation framework, as long as they are sampled and included in the training dataset that will calculate the optimal threshold and boundary.

B. *Description of datasets*

The above mentioned probability distribution functions (for system loading, fault location and fault duration) have been randomly sampled to generate 5971 simulations, a representative enough sample to keep the error of the sample mean lower than 5% as described in [24]. Monte Carlo simulations are used to generate the dataset in this paper, however efficient or importance sampling techniques have also been proposed in the literature and can be utilized to make this process more effective [31]-[33].

For each of these simulations the responses of each generator (voltage and rotor angle) to a single contingency are recorded with a 1/60s=1cycle sampling rate. Consequently, this is also the assumed sampling time of a PMU in this paper. The hierarchical clustering method presented in Section II.A is used to determine the groups of generators exhibiting instability in the obtained dataset. The post-fault time at which the clustering methodology is applied and, consequently, for which the predictions of unstable generator groupings are made, is chosen as T=90 cycles (1.5s). In general T is a parameter to be defined by the system operator and could vary depending on the specific network and time frame for the application of corrective measures. The choice T=90 cycles in this case study is a reasonable time frame for generators to exhibit transient instability, since the aim is to apply corrective control actions as quickly as possible (before time T).

After classification these samples were then divided into three disjoint datasets: 2000 samples for the PNN training dataset, 2000 samples for the boundary training dataset and the remaining 1971 samples for the test dataset. The PNN training, boundary training and test datasets contained 222, 220 and 197 unstable contingencies respectively (1778, 1780 and 1774 stable contingencies respectively). There were in total 15 contingency classes including the stable class, and all were represented in each dataset.

V. NUMERICAL RESULTS

A. *Error rate bounds p_i considerations*

Before specification of the probabilistic bounds p_i, a general discussion is first provided on the choice of these parameters. In principle the probabilistic bounds p_i may be chosen freely, according to the user's desired accuracy requirements for event detection and transient stability assessment from wide area measurements which link to the level of risk a system operator is willing to take. However this comes with the caveat that in practice the user will be faced with a restricted choice of these bounds, for two reasons. Firstly if the error bounds are set too low then they may be impossible to satisfy because of the uncertainty inherent in the problem. Secondly the discrete nature of the training sample means that the set of bounds which yield feasible solutions is typically not continuous.

In this case study the speed of flexible prediction times is compared to that of fixed prediction times under equivalent error constraints. The choices of p_1, p_2 and p_3 can be therefore guided by the error rates achieved by fixed prediction times on the training data. More precisely, the choice of p_1, p_2 and p_3 can be based on the lowest levels achieved for these error rates using fixed times, which can then be progressively relaxed (that is, increased) if necessary to obtain feasible solutions, as in [20]. In the present paper there is additional complexity since the RQP and RSP are intertwined. Indeed the optimal decision rule τ_1^* of the RQP depends on the level of the constraint p_0 and is then passed as input to the RSP (described in Section II.D), which also depends on p_1, p_2 and p_3.

To address this, a list of values for p_0 was first identified which yielded feasible solutions to the RQP. These values were p_0=0.25%, 0.3%, 7%, 12.5% and 36.75%. The error rates p_1, p_2 and p_3 selected as above (according to the error rates achieved by fixed prediction times) were then progressively relaxed until a combination was found which yielded feasible solutions for all values of p_0 in the list. This combination was p_1=6/220=2.73%, p_2=6/1780=0.34% and p_3=24/220=10.9%. The values p_1, p_2 and p_3 specified above are therefore driven by both the comparison with fixed prediction times (are comparable to the values obtained with fixed times) and the requirements of the sensitivity analysis for p_0. It should be noted that relatively high values of p_3 are also observed for the fixed prediction times investigated in Section VI.D below. Indeed this could be expected a priori since some instability classes had very few representatives in the training data.

The above procedure followed in this paper is a suggestion in defining error rates and ensuring the existence of feasible solutions. However, in practice the system operator may choose the error bounds p_i freely (subject as above to the existence of feasible solutions). Lower values for these bounds are more conservative and so tend to give rise to slower but more accurate predictions, and conversely for higher bounds.

B. Optimal boundaries

A numerical calculation of the optimal boundary for the RSP for the case study presented here, is shown in Fig. 3, to highlight the information we can get by observing the boundary itself. Since the first component of Process 2 is the calculated probability that the system is stable, crossing the relatively high probability upper threshold would trigger the decision that the system is stable (recall that the required accuracy for this decision is 100 - p_1 = 97.27%). In contrast, a very low value for the first component of Process 2 crossing the lower boundary would trigger the decision that the system is unstable. As mentioned earlier the probability threshold for identification as stable is generally higher for a quick decision (based on fewer post-contingency observations) and consequently the first component of Process 2 needs to be very high to cross the boundary. On the other hand, the probability needs to be 0 (in this case; recall that the corresponding error constraint is p_2 = 0.34%) to trigger a decision that the system is unstable for the first approximately 14 cycles after the event detection and the lower threshold increases as more post-contingency observations are gathered.

The resulting in-sample error rates (that is, the error rates on the boundary training dataset) are presented in Table I. For each p_0 the solution τ_1^* to the RQP was passed as input to the USP with the parameters p_1 = 2.73%; p_2 = 0.34% and p_3 = 10.9% to obtain the solutions τ_2^* and d^*. From this it can be seen that the conditions (2) and (5) are satisfied approximately in each case. The numerical error can be attributed to the discrete nature of the Monte Carlo sample, as discussed above. The corresponding decision rules τ_1^* and (τ_2^*, d) may therefore be considered as feasible for the RQP and RSP problems respectively.

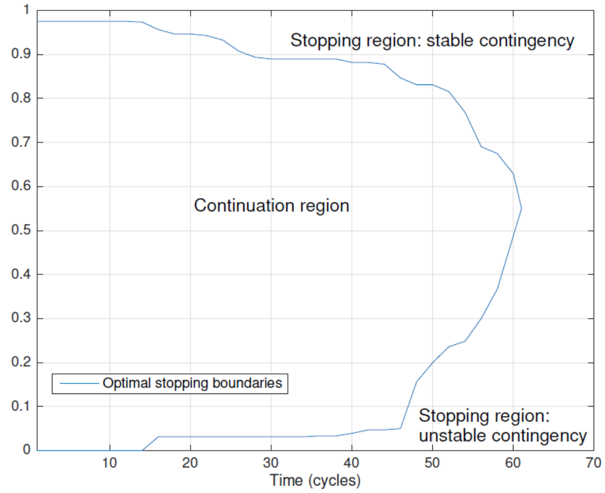

Fig. 3. Numerical optimal decision rule for the RSP in the case study of Section VI (time is measured relative to τ_1^*).

C. Test dataset performance and sensitivity analysis

The out-of-sample performance of the optimal decision rules τ_1^*, τ_2^* and d^* (that is, their performance on the test dataset) is shown in Table II. Performance measures J_1^*, J_2^* and J^* refer to the average delays to declare a contingency, declare that a specific unstable generator grouping will happen and to the overall delay for declaring both aforementioned events, respectively. They are computed for the test dataset, using the decision rules τ_1^*, τ_2^* and d^* from Table I. The RQP optimal threshold A^* and out-of-sample error probabilities $\mathbb{P}[A_1], \mathbb{P}[A_2], \mathbb{P}[A_3]$ are also shown. Relaxing the false alarm constraint by increasing p_0 gradually decreases the average time J_1^* taken for event detection (note that this relationship must hold in the training dataset, but is not guaranteed in the test dataset due to noise). In contrast there is no systematic reduction in the average time J_2^* for event classification. Overall, and particularly since J_1^* is an order of magnitude smaller than J_2^*, there is also no systematic reduction in the total average delay J^*.

D. Comparison to pre-committed strategies

The coupled RQP and RSP problems have been used in this paper to develop decision rules for transient stability prediction which both accommodate rapid event detection and are appropriately flexible, in order to be as quick as possible subject to probabilistic error constraints. The analysis of this approach is completed with a comparison to fixed or pre-committed prediction times, whereby one fixes beforehand the post-disturbance time at which stability predictions are made. In order to focus on this comparison it is assumed in this section that the actual time of the disturbance θ is observable, and further we set θ=0 by temporarily removing pre-fault observations from each sample.

TABLE I
ERROR PROBABILITIES IN BOUNDARY TRAINING DATASET AND LAGRANGE MULTIPLIERS

	p_0(%)	0.25%	7%	12.5%	36.75%
RQP	λ_0^*	9.75	2.98	2.23	2.09
	$\mathbb{P}[A_0]$	0.2%	6.95%	12.4%	36.75%
RSP	λ_1^*	121	138.17	161.03	163.59
	λ_2^*	11	45.5	240.93	372.04
	λ_3^*	78.5	78.5	156	2.01
	$\mathbb{P}[A_1]$	0.91%	0.91%	2.73%	2.27%
	$\mathbb{P}[A_2]$	0.06%	0.06%	0.00%	0.06%
	$\mathbb{P}[A_3]$	10.45%	10.45%	10.91%	10.91%

TABLE II
PERFORMANCE IN TEST DATASET

	p_0(%)	0.25%	7%	12.5%	36.75%
RQP	A^*	0.72	0.47	0.37	0.2
	$\mathbb{P}[A_0]$	0.3%	5.63%	12.99%	37.09%
RSP	J_1^* (cycles)	2.17	1.9	1.73	1.15
	J_2^* (cycles)	23.71	23.45	25.9	24.65
	J^* (cycles)	25.85	25.28	27.38	24.11
	$\mathbb{P}[A_1]$	0.51%	3.05%	2.03%	3.05%
	$\mathbb{P}[A_2]$	0%	0%	0%	0%
	$\mathbb{P}[A_3]$	10.15%	10.15%	9.64%	8.63%

Fixed prediction delays of τ_2=1, 10, 20,..., 60 cycles are considered. For each fixed delay a PNN model is applied directly to solve the contingency classification problem. Table III provides the rates of errors A_1, A_2 and A_3 corresponding to each fixed (pre-committed) post-fault classification time. The lowest error rates, which are also comparable to the probabilistic bounds used in the case study (cf. Table I), are achieved by pre-committed times of at least 50 cycles. The RSP (setting $\tau_1^* = \theta$) requires only one-third of this time (16.35 cycles or 0.27 seconds) on average, a relative improvement comparable to that observed in [20]. Among the 197 unstable contingencies in the test dataset the average delay was 30.13 cycles (0.50 seconds), while among the 1774 stable contingencies the average delay was 14.82 cycles (0.25s).

On the other hand, the values reported for the total delay J^* in Table II show that the combined RQP and RSP performed both event detection and classification within around 25 cycles on average (0.42s), which represents half of the approximately 50 cycles taken for classification alone by the pre-committed strategies with comparable accuracy (even with the benefit of observable events). Further, 90% of predictions were made within 31 cycles (0.52s) and 95% within 40 cycles (0.67s) while the maximum prediction delay was 62 cycles (1.03s). It should also be noted, that the proposed method decides on the optimal time for each specific contingency which is an inherent advantage over previously used pre-committed strategies with fixed decision time.

E. Discussion

Although in this study the training dataset is obtained from Monte Carlo simulations, the methodology developed here is equally applicable to data obtained from actual WAMS measurements. In this case, however, the data may need some pre-processing to eliminate or reduce noise. In the presence of additional noise, the decision rules selected by the above methodology will naturally become more conservative and thus prediction times will tend to be increased. Further it is assumed above that each sample in the training and test data is labelled with its `true' contingency class. It is straightforward in the above setup to incorporate uncertainty in these labels by simply allowing them to be sampled from a probability distribution. In this case a larger dataset will generally be required in order to achieve a representative sample. The pre-processing of recorded data, choice of distribution for labels, and questions of sample size are, however, beyond the scope of this paper.

TABLE III
COMPARISON OF PERFORMANCE WITH FIXED PREDICTION TIME

Time (cycles)	Overall error	$\mathbb{P}[A_1]$	$\mathbb{P}[A_2]$	$\mathbb{P}[A_3]$
1	9.99%	100%	0%	0%
10	9.99%	100%	0%	0%
20	4.06%	32.49%	0.9%	0%
30	2.44%	19.8%	0.51%	3.55%
40	0.96%	9.64%	0%	5.58%
50	0.2%	2.03%	0%	7.61%
60	0.15%	1.02%	0%	6.6%
RSP (avg. 16.35 cycles)	0.25%	2.53%	0%	7.61%

The temporal problem decomposition, which was introduced in Sections II and III, also merits discussion. While the RQP takes as input the observations from time 0 until the contingency detection time τ_1, for computational reasons the RSP has access only to observations from τ_1 onwards. However, since τ_1 is only approximately 2 cycles on average, the resulting information loss is negligible.

VI. CONCLUSIONS

In this paper we propose a method to address the question of risk-constrained optimality in the timing of the decision of online identification of transient instability (exact unstable generator grouping), by incorporating also the event detection prior to classification. The approach has been validated on the IEEE 68 bus test system, where it is proven to be between two and three times faster on average than strategies based on fixed decision times with comparable error rates. Additionally, the proposed method provides the ability to a system operator to set constraints based on the number of errors considered acceptable.

Finally conclusions are drawn from the sensitivity analysis with respect to the error probability rates p_0 (false alarm), p_1 (stable cases classified as unstable), p_2 (unstable cases classified as stable), p_3 (unstable cases that are misclassified as different unstable generator group). The rate of false detections in the RQP (related to disturbance detection) does not feature explicitly in the constraints of the RSP (related to specific unstable generator grouping identification). This means that for a given choice of the probabilistic constraints p_1, p_2 and p_3 in the RSP, there is some freedom in choosing the value of p_0 in the RQP. However, no systematic relationship between p_0 and the speed or accuracy of the classification has been observed. Further, a priori it is rational to constrain the false alarm rate p_0 to be as low as possible (lowest value for which a feasible solution exists).

REFERENCES

[1] J.-A. Jiang, J.-Z. Yang, Y.-H. Lin, C.-W. Liu, and J.-C. Ma, “An adaptive PMU based fault detection/location technique for transmission lines. I. Theory and algorithms,” IEEE Transactions on Power Delivery, vol. 15, pp. 486–493, Apr. 2000.

[2] T. Guo and J. Milanovic, “Online Identification of Power System Dynamic Signature Using PMU Measurements and Data Mining,” IEEE Transactions on Power Systems, vol. PP, no. 99, pp. 1–9, 2015.

[3] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, “Decision trees for real-time transient stability prediction,” IEEE Transactions on Power Systems, vol. 9, pp. 1417–1426, Aug. 1994.

[4] K. Sun, S. Likhate, V. Vittal, V. S. Kolluri, and S. Mandal, “An Online Dynamic Security Assessment Scheme Using Phasor Measurements and Decision Trees,” IEEE Transactions on Power Systems, vol. 22, pp. 1935–1943, Nov. 2007.

[5] J.-A. Jiang, Y.-H. Lin, J.-Z. Yang, T.-M. Too, and C.-W. Liu, “An adaptive PMU based fault detection/location technique for transmission lines. II. PMU implementation and performance evaluation,” IEEE Transactions on Power Delivery, vol. 15, pp. 1136–1146, Oct. 2000.

[6] J. J. Q. Yu, D. J. Hill, A. Y. S. Lam, J. Gu and V. O. K. Li, "Intelligent Time-Adaptive Transient Stability Assessment System," IEEE Transactions on Power Systems, vol. 33, no. 1, pp. 1049-1058, Jan. 2018.

[7] E. Farantatos, R. Huang, G. J. Cokkinides, A. P. Meliopoulos, "A Predictive Generator Out-of-Step Protection and Transient Stability Monitoring Scheme Enabled by a Distributed Dynamic State Estimator," IEEE Trans. Power Delivery, vol. 31, no. 4, pp. 1826-1835, Aug. 2016.

[8] L. Wehenkel, T. V. Cutsem, and M. Ribbens-Pavella, “An artificial intelligence framework for online transient stability assessment of power systems,” IEEE Trans. Power Systems, vol. 4, pp. 789–800, May 1989.

[9] N. Senroy, G. T. Heydt, and V. Vittal, “Decision Tree Assisted Controlled Islanding,” IEEE Transactions on Power Systems, vol. 21, pp. 1790– 1797, Nov. 2006.

[10] M. He, V. Vittal, and J. Zhang, “Online dynamic security assessment with missing pmu measurements: A data mining approach,” IEEE Transactions on Power Systems, vol. 28, pp. 1969–1977, May 2013.

[11] M. He, J. Zhang, and V. Vittal, “Robust Online Dynamic Security Assessment Using Adaptive Ensemble Decision-Tree Learning,” IEEE Transactions on Power Systems, vol. 28, pp. 4089–4098, Nov. 2013.

[12] L. S. Moulin, A. P. A. d. Silva, M. A. El-Sharkawi, and R. J. Marks, “Support vector machines for transient stability analysis of large-scale power systems,” IEEE Transactions on Power Systems, vol. 19, pp. 818–825, May 2004.

[13] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, “Support Vector Machine-Based Algorithm for Post-Fault Transient Stability Status Prediction Using Synchronized Measurements,” IEEE Transactions on Power Systems, vol. 26, pp. 1474–1483, Aug. 2011.

[14] A. D. Rajapakse, F. Gomez, K. Nanayakkara, P. A. Crossley, and V. V. Terzija, “Rotor Angle Instability Prediction Using Post-Disturbance Voltage Trajectories,” IEEE Transactions on Power Systems, vol. 25, pp. 947–956, May 2010.

[15] C.-W. Liu, M.-c. Su, S.-S. Tsay, and Y.-J. Wang, “Application of a novel fuzzy neural network to real-time transient stability swings prediction based on synchronized phasor measurements,” IEEE Transactions on Power Systems, vol. 14, pp. 685–692, May 1999.

[16] A. G. Bahbah and A. A. Girgis, “New method for generators’ angles and angular velocities prediction for transient stability assessment of multimachine power systems using recurrent artificial neural network,” IEEE Transactions on Power Systems, vol. 19, pp. 1015–1022, May 2004.

[17] N. Amjady and S. F. Majedi, “Transient Stability Prediction by a Hybrid Intelligent System,” IEEE Transactions on Power Systems, vol. 22, pp. 1275–1283, Aug. 2007.

[18] N. I. A. Wahab, A. Mohamed, and A. Hussain, “Fast transient stability assessment of large power system using probabilistic neural network with feature reduction techniques,” Expert Systems with Applications, vol. 38, no. 9, pp. 11112–11119, 2011.

[19] S. Kretsinger, S. Rovnyak, D. Brown, and J. Thorp, “Parallel decision trees for predicting groups of unstable generators from synchronized phasor measurements,” in Precise Measurements in Power Systems Conference, Arlington, Virginia, 1993.

[20] J. Gonzalez, Y. Kitapbayev, T. Guo, J. Milanovic, G. Peskir, and J. Moriarty, “Application of sequential testing problem to online detection of transient stability status for power systems,” in 55th Conference on Decision and Control, (Las Vegas, US), 2016.

[21] R. Zhang, Y. Xu, Z. Y. Dong, and K. P. Wong, “Post-disturbance transient stability assessment of power systems by a self-adaptive intelligent system,” IETGener., Transm. Distrib., vol. 9, no. 3, pp. 296–305, Feb. 2015.

[22] T. Zhang, M. Sun, J. L. Cremer, N. Zhang, G. Strbac, & C. Kang, “A Confidence-Aware Machine Learning Framework for Dynamic Security Assessment,” *IEEE Transactions on Power Systems*, Early Access.

[23] A. Venzke, A., S. & Chatzivasileiadis, “Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications,” *IEEE Transactions on Smart Grid*, *12*(1), 383–397, January 2021.

[24] P. N. Papadopoulos, T. Guo and J. V. Milanović, "Probabilistic Framework for Online Identification of Dynamic Behavior of Power Systems With Renewable Generation," IEEE Trans. Power Syst., vol. 33, no. 1, pp. 45-54, Jan. 2018.

[25] P. D. Wasserman, Advanced methods in neural computing. John Wiley & Sons, Inc., 1993.

[26] P. Johnson, J. Moriarty, and G. Peskir, “Detecting changes in real-time data: a user’s guide to optimal detection,” Phil. Trans. R. Soc. A 375: 20160298, Jul. 2017.

[27] D. P. Bertsekas, Nonlinear programming. Athena Scientific, 1999.

[28] G. Rogers, Power System Oscillations. Springer Science & Business Media, Dec. 2012.

[29] B. Pal and B. Chaudhuri, Robust control in power systems. Springer Science & Business Media, 2006.

[30] R. Lewis and V. Torczon, “A Globally Convergent Augmented Lagrangian Pattern Search Algorithm for Optimization with General Constraints and Simple Bounds,” SIAM J. Optim., vol. 12, pp. 1075–1089, Jan. 2002.

[31] F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, "Efficient Database Generation for Data-Driven Security Assessment of Power Systems," *IEEE Transactions on Power Systems,* vol. 35, no. 1, pp. 30-41, 2020.

[32] P. N. Papadopoulos and J. V. Milanovic, "Efficient Identification of Transient Instability States of Uncertain Power Systems," in *10th Bulk Power Systems Dynamics and Control Symposium – IREP’2017*, Espinho, Portugal, 2017, pp. 1-7.

[33] M. Sun, I. Konstantelos, G. Strbac, “A Deep Learning-Based Feature Extraction Framework for System Security Assessment,” *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 5007–5020, 2018.

Jhonny Gonzalez holds a MSc in Mathematical Finance (2011) and a PhD in Mathematics (2015) both from the University of Manchester, UK, where he also completed his postdoctoral research. He is currently a commodities derivatives quantitative analyst with Citigroup in London.

Panagiotis N. Papadopoulos (S’05-M’14) received the Dipl. Eng. and Ph.D. degrees from the Department of Electrical and Computer Engineering at the Aristotle University of Thessaloniki, in 2007 and 2014, respectively. Currently, he is a Senior Lecturer and a UKRI Future Leaders Fellow in the Department of Electronic and Electrical Engineering at the University of Strathclyde.

Jovica V. Milanović (M'95, SM'98, F’10) received the Dipl.Ing. and M.Sc. degrees from the University of Belgrade, Belgrade, Yugoslavia, the Ph.D. degree from the University of Newcastle, Australia, and the Higher Doctorate (D.Sc. degree) from The University of Manchester, U.K., all in electrical engineering. Currently, he is a Professor of Electrical Power Engineering, and Deputy Head of Department in the Department of Electrical and Electronic Engineering at the University of Manchester, U.K.

Goran Peskir received Ph.D. and Dr.Scient. degrees in Mathematics from the University of Aarhus, Denmark. He currently holds the Chair of Probability in the Department of Mathematics at the University of Manchester, U.K.

John Moriarty received the B.A. in Mathematical Sciences, M.Sc. in Applied Statistics and D.Phil. in Mathematics from the University of Oxford. Currently, he is a Professor of Mathematics in the School of Mathematical Sciences at Queen Mary University of London, U.K. and a Research Fellow at the Alan Turing Institute, U.K.