Extending the Survival Signature Paradigm to Complex Systems with Non-repairable Dependent Failures

Journal Title
XX(X):1–21

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Hindolo George-Williams[1,3], Geng Feng[2], Frank P.A. Coolen[4], Michael Beer[5,1,6], and Edoardo Patelli[1]

Abstract

Dependent failures impose severe consequences on a complex system's reliability and overall performance, and a realistic assessment, therefore, requires an adequate consideration of these failures. System survival signature opens up a new and efficient way to compute a system's reliability, given its ability to segregate the structural from the probabilistic attributes of the system. Consequently, it outperforms the well-known system reliability evaluation techniques, when solicited for problems like maintenance optimization, requiring repetitive system evaluations. The survival signature, however, is premised on the statistical independence between component failure times and more generally, on the theory of weak exchangeability, for dependent component failures. The assumption of independence is flawed for most realistic engineering systems whilst the latter entails the painstaking and sometimes impossible task of deriving the joint survival function of the system components. This paper, therefore, proposes a novel, generally applicable, and efficient Monte Carlo Simulation approach that allows the survival signature to be intuitively used for the reliability evaluation of systems susceptible to induced failures. Multiple component failure modes, as well, are considered, and sensitivities are analysed to identify the most critical Common-Cause Group to the survivability of the system. Examples demonstrate the superiority of the approach.

Keywords

Dependencies, Survival Signature, Monte Carlo Simulation, System Reliability, Multiple Failure Mode

[1]Institute for Risk and Uncertainty, University of Liverpool, Liverpool, United Kingdom
[2]School of Engineering, University of Central Lancashire, Preston, United Kingdom
[3]Institute of Nuclear Engineering & Science, National Tsing Hua University, Hsinchu, Taiwan
[4]Department of Mathematical Sciences, Durham University, Durham, United Kingdom
[5]Institute for Risk and Reliability, Leibniz University Hannover, Hannover, Germany
[6]School of Civil Engineering & Shanghai Institute of Disaster Prevention and Relief, Tongji University, Shanghai, China

Corresponding author:
Edoardo Patelli, Institute for Risk and Uncertainty, University of Liverpool, Liverpool, United Kingdom
Email: edoardo.patelli@liverpool.ac.uk
Note: The first two authors contributed equally to this work.

Prepared using sagej.cls *[Version: 2017/01/17 v1.20]*

Introduction

Dependent failures are failure events affecting multiple components simultaneously. Their origin is traceable to entities external to the system or to the system components themselves. The proper consideration and modelling, therefore, of these failures is essential in complex systems reliability analysis, as they may impose adverse effects on a system's overall functionality.

Interdependencies in engineering systems are manifested at two levels; between components (inter-component), which can be functional or induced and between systems/subsystems (inter-system). Functional dependencies are due to the topological and/or functional relationships between components. For instance, a motor-operated valve would not work if the electric motor controlling its actuator stopped due to a breaker failure. In this case, the valve is said to be functionally dependent on the breaker through the motor. Induced dependencies, on the other hand, are due to a state change in one component (the initiator) triggering a state change in another (the induced), such that even when the initiator is reinstated, the induced does not reinstate, unless manually made to do so. In the valve-motor-breaker example, for instance, the valve would resume its normal operation, once the faulty breaker is replaced, highlighting the dichotomy between functional and induced dependencies. Functional dependencies are intrinsically accounted for by the innate attributes of the system reliability technique while induced dependencies require explicit modelling. Inter-system dependencies, on the other hand, are due to functional or induced couplings between multiple systems. Unlike standalone systems, functional dependencies in these systems may require explicit modelling. This is the case especially for components relying on material generated and transmitted by the components of another system, under which condition the reliability modelling technique used may prove inadequate. The modelling of this type of dependency is outside the scope of this work.

Induced dependencies are further divided into Common-Cause Failures (CCF) and cascading failures, as illustrated in Figure 1. Common-Cause Failures (CCF) are the simultaneous failure of multiple similar components due to the same root cause[1–3]. Their origin is traceable to a coupling that normally is external to the system. Notable instances are shared manufacturing lines/materials, shared maintenance teams, shared environments, and human error. A group of components susceptible to the same CCF event is called a Common-Cause Group (CCG). An important point to note about CCF is that, on occurrence of the failure event, there is a probability associated with multiple component failure and that the affected components only fail in the same mode. A CCF may affect an entire system or only a fraction of its components. Consequently, the number of components involved in a CCF event ranges from 1 to the total number of components in its CCG. CCF have been shown (in Ref.[4], for instance) to decrease the reliability and performance of multi-component systems. They, therefore, must be given due consideration.

CCF modelling and quantification has always attracted keen interest from both researchers and practitioners of system reliability and safety engineering. A total of five parametric models have been put forward to express the CCF probability of a CCG. The original model, the Basic Parameter Model (BPM), expresses the probability of a basic failure event involving a specific number of components. The other models, the β-factor model, the Multiple Greek Letter Model (MGL), the α-factor model, and the Binomial Failure Rate model, are only a mere reparameterization of the BPM. Of these, the MGL and the α-factor models are the most widely used in the reliability and risk assessment of systems. See Refs.[1,3] for details on these models and

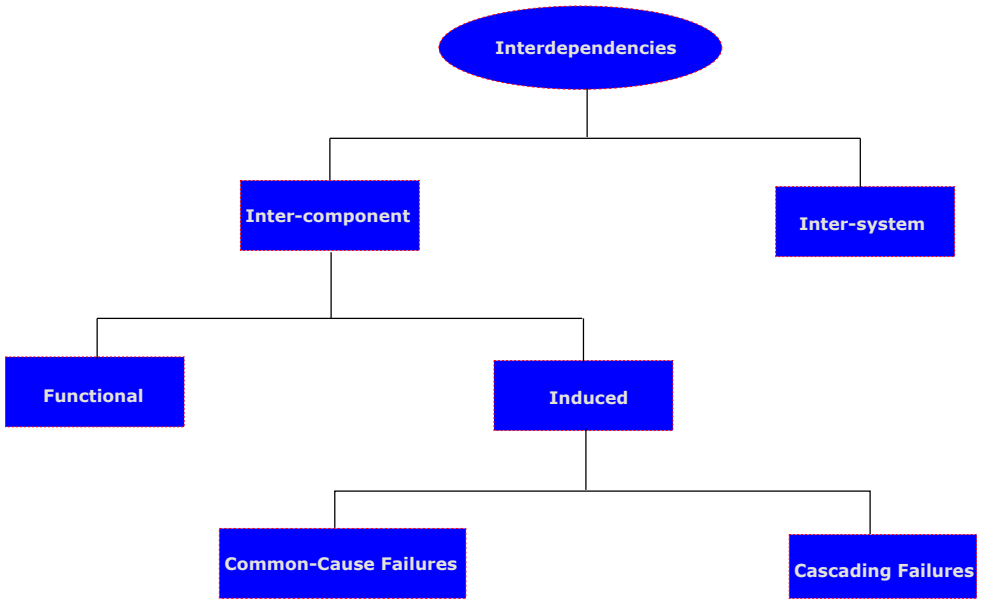

Figure 1. Forms of interdependencies in engineering systems

their relationships. Rasmuson and Kelly also reviewed, in their work[5], the basic concepts of modelling CCF in systems. Rausand and Arnljot[6] proposed the square-root method, a simple bounding technique that estimates the effects of CCF on a system but which, however, has no mathematical support for application. A robust Bayesian approach for quantifying the α-factor parameters of a CCG in the presence of epistemic uncertainties has also been put forward by Troffaes et al.[7] Their approach, however, is limited to component level reliability and, therefore, requires a second approach to obtain the system level reliability indices. For this, Fan's stochastic hybrid systems model[8], O'Connor's general cause-based methodology[9], or Ramirez-Marquez's reliability optimization approach[10] would do. Of course, only if the reliability analyst is willing to turn a blind eye to their respective drawbacks. These models are built on reliability evaluation techniques that do not segregate the topological from the probabilistic attributes of the system. As such, they are computationally expensive for problems involving multiple reliability analysis of the same system.

Cascading failures are those with the capacity to trigger the instantaneous failure of one or more components of a system. They can originate from a component or from a phenomenon outside the system boundary. The likelihood of the initiating event originating from within the system, distinguishes them from CCF. Another point of dichotomy is that the affected components do not necessarily have to be similar or fail in the same mode. In addition, at the occurrence of the initiating event, the probability of all the coupled components failing is unity, save for the case when they are in a state rendering them immune. A few prominent examples of initiating events external to the system are extreme environmental events, natural disasters, external shocks, erroneous/malicious human-system interactions, and terrorist threats (see e.g.[11]). Various models have been developed to study the effects of cascading failures on complex systems (see

Ref. [12] for a detailed review). Again, like the existing CCF modelling techniques, these models, including the load-flow simulation proposed by George-Williams et al. [13], consider the probabilistic and topological attributes of the system as a unit. They, therefore, are computationally expensive for certain system reliability problems.

The system survival signature segregates the structure function of a system from the probabilistic properties of its components. Since a system's structure function changes only with changes in its topology, survival signature-based approaches are a computationally efficient alternative for maintenance optimization, uncertainty, and sensitivity analysis problems, where only the probabilistic attributes of the system change. For these problems, the system's survival signature is computed just once and reused in multiple reliability analyses. Other techniques, however, would require the evaluation, directly or otherwise, of both the probabilistic and topological attributes of the system, on every analysis.

Since its introduction, the survival signature has been invoked in various ways, and it is gradually gaining popularity in system reliability analysis. Aslett et al. [14], for instance, incorporated it into their Bayesian framework for system reliability analysis. Reed [15] used BDD to develop an efficient and exact algorithm to compute system survival signatures. Feng et al. [16] went a step further by proposing an analytical approach for analysing systems with imprecision in component failure time distributions. Patelli et al. [17], on the other hand, proposed a generic simulation approach for computing the reliability of complex systems, using the survival signature. However, these works assume full independence between the component failure times. In fact, only Coolen, Eryilmaz, and Coolen-Maturi have made realistic attempts at extending the notion of survival signature to systems with interdependencies. In 2014, Coolen and Coolen-Maturi proposed a predictive model [18] that deduces the number of components that fail, as well as the subsequent reliability of the system, following a CCF event. Their model, however, stops short at computing the overall effect of CCF on the system and is applicable only to systems with a single component type. Most importantly, it does not consider cascading failures. Eryilmaz, Coolen, and Coolen-Maturi later adapted the survival signature to compute the importance measures [19] and mean residual life [20] of coherent systems with dependent components. Their adaptations were based on the theory of weak exchangeability [21] of component failure times. With this theory, components of the same type can be dependent and have exchangeable failure times while components of different types may or may not be dependent. The only downside to its use, however, is the need for knowledge of the joint survival function of the components, prior to system analysis. While this is not impossible, it is in no way straight-forward for complex systems with nested cascade failures.

In this work, therefore, we extend the survival signature-based approach to complex systems exhibiting susceptibility to various forms of interdependencies. We will consider induced dependencies in standalone systems, with the assumption that the relevant CCF parameters and cascading dependencies are known. Functional dependencies will be implicitly accounted for by the proposed reliability framework while inter-system dependencies are outside the scope of this work. We will also perform a series of computational experiments to validate the framework and compare its computational efficiency with the load-flow simulation technique.

Overview of Proposed Approach

We harness the existing survival signature-based modelling formalism to propose a more realistic approach to system reliability analysis. In the proposed approach,

the survival signature of the system is obtained prior to system analysis, using its topological attributes only. An efficient event-driven Monte Carlo simulation is then invoked to recreate the failure of the components and propagate the ensuing dependencies, where necessary. The number of operating components is determined for each component type, with the corresponding system survival signature directly read off from a predefined register and saved as a function of time. These stored values are later used to compute the time-dependent reliability of the system, using basic probabilistic principles.

Since the survival signature of a system is fixed so long as its topology does not change, the proposed approach makes an efficient alternative. Its efficiency particularly stands out in maintenance optimization problems, sensitivity & uncertainty analyses, and other problems requiring multiple system reliability evaluations. Also, because it is simulation-based, it can accommodate any component failure time distribution type, including user-defined distributions. In summary, the proposed approach inherits the desirable attributes of both the survival signature-based and Monte Carlo simulation-based approaches. It is, to the best of our knowledge, the first documented extension of the survival signature-based approach to the complete system level reliability evaluation of complex systems susceptible to both common-cause and cascading failures.

Theoretical Basics

The operating status of an M-component system at time, t, can be deduced from its state vector, $\underline{\boldsymbol{x}} = (x_1, x_2, ..., x_M)$, where x_i is the state of the i^{th} component at that time. For binary-state systems, $x_i = 1$, if the i^{th} component is working and 0, if failed. Consequently, $\underline{\boldsymbol{x}} \in \{0,1\}^M$ and $x_s \in \{0,1\}$, x_s being the state of the system. By considering all the possible state vectors of the system, a function that maps the states of the components to the states of the system can be obtained. This mapping, otherwise known as the structure function, $\varphi(\underline{\boldsymbol{x}})$, of the system, is an algebraic expression taking the value 1 when the system works, and 0, when it is failed. It is an algebraic representation of the system topology and dissociates the connectivity of the components from their probabilistic attributes. Given its structure function alone, the reliability of a system can be computed from the reliabilities of its components directly. The structure function also finds use in system indexing and comparison[22]. However, being an algebraic expression, the possibility of multiple equivalent expressions for the same system exists. This is the case especially for topologically complex systems, which was why Samaniego[22] proposed an alternative representation of the system structure. This new representation, which he called the system signature, is an M-dimensional probability vector whose i^{th} element denotes the probability of the i^{th} component failure leading to system failure. It is hinged on the assumption that all the components of the system are identical, with independently distributed failure times. This assumption, however, is unrealistic in two ways; first, most practical systems are composed of a variety of components. Second, as discussed in the previous section, interdependencies are an inevitability in most systems, rendering their component failure times correlated.

In response, Coolen et al.[23] proposed a new formalism, the survival signature, to generalise Samaniego's system signature. With the survival signature, the assumption of identical components is no longer mandatory, only, they too must fail independently. The survival signature, $\boldsymbol{S}_\tau(l_1, l_2, ..., l_K)$, of a system with K different types of components, is the probability that the system will work when l_1 components of type 1, l_2 components of type 2, l_3 components of type 3, and so on, are working.

Mathematical Formulation

Consider a system with K component types, with M_k components of type $k \in \{1, 2, ..., K\}$, such that $\sum_{k=1}^{K} M_k = M$. Let the random failure times of components of the same type be identical and independently distributed. Consequently, components of the same type can be grouped and defined by the set $\boldsymbol{\rho}^{\{k\}}$. Each $\boldsymbol{\rho}^{\{k\}} \forall k \in \{1, 2, ..., K\}$ is considered an independent subsystem, which gives rise to a total of K subsystems, at the system level. The system state vector can then be written as, $\underline{\boldsymbol{x}} = \{\underline{\boldsymbol{x}}_1, \underline{\boldsymbol{x}}_2, ..., \underline{\boldsymbol{x}}_K\}$, where $\underline{\boldsymbol{x}}_k$ is the state vector for subsystem k (type k components).

Now, let's modify $\underline{\boldsymbol{x}}$ to denote the actual number of available components of each component type, at a given instance. The modified system state vector, $\underline{\boldsymbol{x}}'$, is a K-element vector, such that $\underline{\boldsymbol{x}}' = \{x'_1, x'_2, ..., x'_K\}$, where x'_k, the number of available type k components, is equivalent to $\sum \underline{\boldsymbol{x}}_k$. Since components of the same type are similar, there are $\binom{M_k}{x'_k}$ state vectors, $\underline{\boldsymbol{x}}_k$, where exactly x'_k of the M_k components are working. Therefore, there are $\prod_{k=1}^{K} \binom{M_k}{x'_k}$ system state vectors, $\underline{\boldsymbol{x}}$, corresponding to $\underline{\boldsymbol{x}}'$. If this set of vectors is denoted by $\underline{\boldsymbol{X}}$, following from the definition of the survival signature and the fact that all the state vectors in $\underline{\boldsymbol{X}}$ are equally likely to occur,

$$\boldsymbol{S}_\tau\left(\underline{\boldsymbol{x}}'\right) = \left[\prod_{k=1}^{K} \binom{M_k}{x'_k}\right]^{-1} \times \sum_{\underline{\boldsymbol{x}} \in \underline{\boldsymbol{X}}} \varphi\left(\underline{\boldsymbol{x}}\right) \qquad (1)$$

where $\boldsymbol{S}_\tau\left(\underline{\boldsymbol{x}}'\right)$ is the system survival signature, given $\underline{\boldsymbol{x}}'$.

Let $F_k(t)$ be the cumulative failure time distribution (CDF) for type k components, then the probability of exactly x'_k components being in operation and $M_K - x'_k$, failed, is deduced from the binomial theory as, $\binom{M_k}{x'_k} \left[F_k(t)\right]^{M_K - x'_k} \left[1 - F_k(t)\right]^{x'_k}$. Hence, the occurrence probability, $P\left(\underline{\boldsymbol{x}}'\right)$, of the state vector, $\underline{\boldsymbol{x}}'$, is expressed as,

$$P\left(\underline{\boldsymbol{x}}'\right) = \prod_{k=1}^{K} \binom{M_k}{x'_k} \left[F_k(t)\right]^{M_K - x'_k} \left[1 - F_k(t)\right]^{x'_k} \qquad (2)$$

Therefore, the expected survival function of the system, given $\underline{\boldsymbol{x}}'$, is the product, $\boldsymbol{S}_\tau\left(\underline{\boldsymbol{x}}'\right) \times P\left(\underline{\boldsymbol{x}}'\right)$. The survival function or the reliability, $R(t)$, of the system, is the sum of the expected survival functions yielded by all its modified state vectors. For a system with K component types, there are $\prod_{k=1}^{K} (M_k + 1)$ such state vectors and,

$$R(t) = \sum_{\underline{\boldsymbol{x}}' \in \underline{\boldsymbol{X}}'} \left[\boldsymbol{S}_\tau\left(\underline{\boldsymbol{x}}'\right) \times P\left(\underline{\boldsymbol{x}}'\right)\right] \qquad (3)$$

where $\underline{\boldsymbol{X}}'$ is the global set containing all the modified state vectors, $\underline{\boldsymbol{x}}'$, of the system.

Modelling & Simulating the System

Consider the system described in the previous section, and suppose the random failure of a component of type k may trigger the failure of one or more components. In addition, let one or more components have multiple total failure modes, which in turn may have different effects on the system. By total failure we mean, the component is completely failed and its output/structure function is 0. The component, in other words, is still deemed binary-state. Suppose also that the system is not only susceptible to cascading failures emanating from within its boundaries but to cascading failures triggered by external factors, as well. Clearly, the existing survival signature-based reliability evaluation approaches are inadequate for such a system. This section provides a detailed description of the modelling approach for such systems.

Components with Multiple Failure Modes

The survival signature, by default, is suited to binary-state components and systems. Therefore, when the system being modelled also contains components with multiple failure modes, the analyst would need to make these compatible with the signature-based approach.

Consider a component with two failure modes, and which occurrence times follow the CDF, $F_1(t)$ and $F_2(t)$, respectively. If these failure modes have the same effect on the system, then, they can be merged. Two total failure modes have the same effect on a system if they do not trigger dependent failures or if they have an equal likelihood of affecting the same set of components. The effective CDF, $F_i(t)$, of the component is computed from the probability, $P(min(T_1, T_2) \leq t)$, where T_1 and T_2 are the random occurrence times of failure modes 1 and 2, respectively. This relation follows from the reasoning that the component is failed on the occurrence of any of the failure modes. The resulting probability could also be viewed as the complement of the probability that none of the failure modes occurs, yielding, $F_i(t) = 1 - [1 - F_1(t)][1 - F_2(t)]$. Generally, the effective CDF, $F_i(t)$, of an n failure mode component is given by $F_i(t) = 1 - \prod_{l=1}^{n}[1 - F_l(t)]$, so long as the failure modes are total and impose the same effect on the system.

There are times when a set of component failure modes do not satisfy the condition for merging. Currently, such a scenario cannot be solved analytically, and we will, therefore, not bother ourselves with computing the effective CDF of the component. Instead, we will propose a set of procedures to segregate the component into several binary-state elements, which then can be easily implemented by a Monte Carlo simulation algorithm. It should be noted that this segregation is only required to enhance the intuitive representation of the inter-component dependencies and ensure a simplified simulation algorithm. The system, otherwise, could still be analysed, only the sampling algorithm and dependency matrix proposed in Ref.[24] and Ref.[13], respectively will be required, complicating an otherwise simple solution.

An n failure mode component, i, is segregated by redefining its state-space to contain the working state (conservatively assumed to be state 1), and one of the failure modes (assumed to be state 2). The remaining $n-1$ failure modes (assumed to be state 3 to state n) are each then assigned to a virtual binary-state node. Component i retains the failure time distribution to state 2 while the virtual nodes inherit the failure time distributions to their respective failure modes. The virtual nodes, as their name implies, are not really a part of the system, and should, therefore, be considered external factors/nodes. External nodes are not considered when deriving the survival signature set, $\boldsymbol{S}_\tau$, of the system, which is why their numbering starts from $M+1$, M being the number of system components. Since in practice, the virtual nodes, together with the parent node, i, represent the same component, the failure of any of these nodes denotes the failure of the component. Hence, each virtual node is a dual of the parent node, i.

When a node fails, its duals can no longer affect the system, since the failure modes of a component are mutually exclusive and in this work, non-repairable. Consequently, the system simulation algorithm should be equipped with a special routine to ensure affected dual nodes are removed, following a failure event. For this, we propose an efficient recursive algorithm. It takes the current values of $\underline{\boldsymbol{x}}'$, $\boldsymbol{\rho}^{\{k\}}$ for all $k \in \{1, 2, ..., K\}$, the set, $\mathbb{F}$, of failed components, and the set, $\mathbb{D}_i$, of duals of component i for all $i \in \{1, 2, ..., M + M'\}$, returning $\underline{\boldsymbol{x}}'$, $\boldsymbol{\rho}^{\{k\}}$, and $\mathbb{F}$, where M' is the number of external nodes. Following the failure of a component, the algorithm first removes all its duals that are not in operation, from $\mathbb{D}_i$. The component type, k, of the first active dual is

Algorithm 1 Procedure for removing dual nodes

Require: $\underline{\boldsymbol{x}}'$, $\mathbb{F}$, $\boldsymbol{\rho}$, $\mathbb{D}$, i

1: **function** REMOVEDUALNODES($\underline{\boldsymbol{x}}'$, $\mathbb{F}$, $\boldsymbol{\rho}$, $\mathbb{D}$, i)
2: $\mathbb{D}_i \leftarrow \mathbb{D}_i - \mathbb{F}$ ▷ Remove failed nodes
3: **if** $\mathbb{D}_i \leftarrow \emptyset$ **then**
4: go to line 13 ▷ Exit algorithm if i has no duals
5: **end if**
6: **for** $j \in \mathbb{D}_i$ **do** ▷ Loop over dual nodes
7: $k \leftarrow$component type of node j
8: $\boldsymbol{\rho}^{\{k\}} \leftarrow \boldsymbol{\rho}^{\{k\}} - j$ ▷ Remove j from set
9: $(\underline{\boldsymbol{x}}', k) \leftarrow \mid \boldsymbol{\rho}^{\{k\}} \mid$ ▷ Update state vector
10: $\mathbb{F} = \mathbb{F} \cup j$ ▷ Update failed nodes/components
11: $(\underline{\boldsymbol{x}}', \mathbb{F}, \boldsymbol{\rho}) \leftarrow$REMOVEDUALNODES($\underline{\boldsymbol{x}}'$, ..., j)
12: **end for**
13: **return** ($\underline{\boldsymbol{x}}'$, $\mathbb{F}$, $\boldsymbol{\rho}$)
14: **end function**

determined, following which it is removed from the set of components, $\boldsymbol{\rho}^{\{k\}}$, in that group, and the k^{th} element of the modified system state vector, $\underline{\boldsymbol{x}}'$, replaced with the cardinality of $\boldsymbol{\rho}^{\{k\}}$, which in other words is written as $(\underline{\boldsymbol{x}}', k) = \mid \boldsymbol{\rho}^{\{k\}} \mid$. Due to the possibility of a dual node possessing its own duals, the algorithm is recursively applied to the node, as highlighted on line 11 of Algorithm 1. The sequence is repeated for the remaining active duals, adding each to set $\mathbb{F}$ before moving on to the next. Algorithm 1 summarises the procedure for removing the duals of component i, following its failure. In the algorithm and the remainder of this paper, $\mathbb{D}$ denotes the global set of $\mathbb{D}_i \forall i \in \{1, 2, ..., M + M'\}$, such that $\mathbb{D} = \{\mathbb{D}_1, \mathbb{D}_2, ..., \mathbb{D}_{M+M'}\}$. Similarly, $\boldsymbol{\rho}$ denotes the global set of $\boldsymbol{\rho}^{\{k\}} \forall k \in \{1, 2, ..., K\}$, such that $\boldsymbol{\rho} = \{\boldsymbol{\rho}^{\{1\}}, \boldsymbol{\rho}^{\{2\}}, ..., \boldsymbol{\rho}^{\{K\}}\}$.

Cascading Failure Modelling and Propagation

We represent the cascading dependency between components by the cascading matrix, $\mathbb{C}$. The cascading matrix, which can be a sparse matrix, is an $(M + M')$ order square matrix which elements denote whether or not the failure of a component can trigger the almost instantaneous failure of another component. The element in row i and column j of the matrix is assigned the value 1 if the failure of component i can induce failures (in the cascading failure sense) in component j, and 0, otherwise. Therefore, the set, $\mathbb{I}_i$, of components which failure is induced by component i is given by the column indices of the non-zero elements of row i of $\mathbb{C}$.

Algorithm 2 Procedure for cascading failures

Require: $\underline{\boldsymbol{x}}'$, $\mathbb{F}$, $\boldsymbol{\rho}$, $\mathbb{D}$, $\mathbb{C}$, i

1: **function** CASCADEFAILURE($\underline{\boldsymbol{x}}'$, $\mathbb{F}$, $\boldsymbol{\rho}$, $\mathbb{D}$, $\mathbb{C}$, i)
2: $\mathbb{I}_i \leftarrow$induced components obtained from $\mathbb{C}$
3: $\mathbb{I}_i \leftarrow \mathbb{I}_i - \mathbb{F}$ ▷ Remove failed nodes
4: **if** $\mathbb{I}_i \leftarrow \emptyset$ **then**
5: go to line 15 ▷ Exit if i cannot induce failure
6: **end if**
7: **for** $j \in \mathbb{I}_i$ **do** ▷ Loop over induced components
8: $k \leftarrow$component type of node j
9: $\boldsymbol{\rho}^{\{k\}} \leftarrow \boldsymbol{\rho}^{\{k\}} - j$ ▷ Remove j from set
10: $(\underline{\boldsymbol{x}}', k) \leftarrow \mid \boldsymbol{\rho}^{\{k\}} \mid$ ▷ Update state vector
11: $\mathbb{F} = \mathbb{F} \cup j$ ▷ Update failed nodes/components
12: $(\underline{\boldsymbol{x}}', \mathbb{F}, \boldsymbol{\rho}) \leftarrow$REMOVEDUALNODES($\underline{\boldsymbol{x}}'$, ..., j)
13: $(\underline{\boldsymbol{x}}', \mathbb{F}, \boldsymbol{\rho}) \leftarrow$CASCADEFAILURE($\underline{\boldsymbol{x}}'$, ..., j)
14: **end for**
15: **return** ($\underline{\boldsymbol{x}}'$, $\mathbb{F}$, $\boldsymbol{\rho}$)
16: **end function**

To account for these cascading dependencies in the operation of the system, we propose a second recursive algorithm to propagate their effects across the system during simulation. The algorithm takes in the same input set required by Algorithm 1 in addition to the cascade matrix and returns $\underline{\boldsymbol{x}}'$, $\boldsymbol{\rho}^{\{k\}}$, and $\mathbb{F}$. Following the failure of a component, the algorithm first deduces the possible set of components that can be affected. From this set, currently inactive components are removed, and the rest of the procedure is similar to what obtains in Algorithm 1. Since an induced component can also induce failures in

other components, the algorithm recursively calls itself, this time, to propagate any failures the induced may induce. The procedure is summarised by Algorithm 2.

CCF Modelling and Propagation

For non-repairable binary-state systems, a CCG is characterised by a set of probabilities. This set defines the likelihood of a given number of components being involved in any random failure event affecting the group.

Let the CCF probability for component type k be defined by $\boldsymbol{\theta}^{\{k\}}$, such that $\boldsymbol{\theta}^{\{k\}} = \{\theta_r^{\{k\}}\}^{M_k} = \{\theta_1^{\{k\}}, \theta_2^{\{k\}}, ..., \theta_{M_k}^{\{k\}}\}$, r being the total number of components affected by the failure event, and $\sum_{r=1}^{M_k} \theta_r^{\{k\}} = 1$. In effect, $\theta_r^{\{k\}}$ denotes the probability of an additional $r-1$ components failing, following the failure of a type k component, in conformity with the α-factor model. A key requirement, therefore, is that CCF probabilities are expressed according to this model. Probabilities expressed according to the Multiple Greek Letter model would need to be converted as outlined in Ref.[1]

$$\mathbb{H} = \{H_{kr}\}^{K \times max\{\boldsymbol{\Lambda}\}} \mid H_{kr} = \begin{cases} \theta_r^{\{k\}} & \text{If } r \leq M_k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In the most general sense, the notation established in the preceding paragraph could as well be used for component types immune to CCF. For this special case, $\theta_1^{\{k\}} = 1$ and $\theta_r^{\{k\}} = 0$ for all $r > 1$, which by definition, means, the probability of no additional component failing, following the failure of a type k component is 1. Leaning on this fact, we introduce the CCF matrix, $\mathbb{H}$, to define the CCF characteristics of a system with a mix of component types susceptible and immune to CCF. $\mathbb{H}$ is a $K \times max\{\boldsymbol{\Lambda}\}$ matrix, where $\boldsymbol{\Lambda} = \{M_1, M_2, ..., M_{K-1}, M_K\}$ is the set of number of components, $M_k \mid k = 1, 2, ..., K$, in each group. Each row of $\mathbb{H}$, therefore, defines the CCF characteristics of the component type corresponding to the index of that row, as outlined as in Equation 4. The attributes of $\mathbb{H}$ impose two constraints;

1. A component can only belong to one CCG. This implies, CCF events in one CCG are independent of the CCF events in other CCGs. They can, however, still induce cascading failures in these CCGs.
2. For a given component type k, all its M_k components must belong to the same CCG. What this means is, no component should be immune to a CCF to which some components of the same type are susceptible. Strictly speaking, in real life, it is unlikely to have a CCF affecting only a fraction of the components of a given type. However, on its unlikely occurrence, we suggest the components be segregated into two component types, based on susceptibility to CCF.

These constraints should, therefore, be kept in mind when defining component types. As a rule-of-thumb, every component type should be viewed as a CCG, and defined as such, whether or not it is susceptible to CCF. This, indeed, is logical, since components of the same type have similar characteristics, and would, therefore, be influenced by the same common-cause event.

With the CCF modelling procedure outlined, the remainder of this section details CCF propagation during system simulation. Recalling simulation entails recreating the actual operating principles of a system, we propose a very simple procedure for propagating CCF, following the failure of a component. When a member of a CCG fails, there could be $0, 1, 2$ up to $M_k - 1$ additional component failures. The total number of component failures is determined by the CCF matrix, as discussed earlier. Therefore, following a component failure, the number of additional components to fail is first deduced. This is achieved by generating a uniform random number, U, between 0 and 1 and comparing it to the cumulative

Algorithm 3 Procedure for propagating CCF

Require: $\underline{\boldsymbol{x}}'$, $\boldsymbol{\rho}$, $\mathbb{F}$, $\mathbb{D}$, $\mathbb{H}'$, $\mathbb{C}$, k

1: **function** PROPAGATECCF($\underline{\boldsymbol{x}}'$, $\boldsymbol{\rho}$, $\mathbb{F}$, $\mathbb{D}$, $\mathbb{H}'$, $\mathbb{C}$, k)
2: **if** $H'_{k1} \leftarrow 1$ **or** $\boldsymbol{\rho}^{\{k\}} \leftarrow \emptyset$ **then**
3: go to line 16 ▷ Exit if CCF is not possible
4: **end if**
5: get r ▷ Get number of components to fail
6: **if** $r > 1$ **then** ▷ Multiple components affected
7: $\boldsymbol{Z} \leftarrow$ set of $(r-1)$ components from $\boldsymbol{\rho}^{\{k\}}$
8: $\boldsymbol{\rho}^{\{k\}} \leftarrow \boldsymbol{\rho}^{\{k\}} - \boldsymbol{Z}$ ▷ Remove components
9: $(\underline{\boldsymbol{x}}', k) \leftarrow\mid \boldsymbol{\rho}^{\{k\}} \mid$ ▷ Update state vector
10: $\mathbb{F} = \mathbb{F} \cup \boldsymbol{Z}$ ▷ Update failed nodes
11: **for** $j \in \boldsymbol{Z}$ **do** ▷ Loop over components
12: $(\underline{\boldsymbol{x}}', \mathbb{F}, \boldsymbol{\rho}) \leftarrow$REMOVEDUALNODES$(..., j)$
13: $(\underline{\boldsymbol{x}}', \mathbb{F}, \boldsymbol{\rho}) \leftarrow$CASCADEFAILURE$(\underline{\boldsymbol{x}}', ..., j)$
14: **end for**
15: **end if**
16: **return** $(\underline{\boldsymbol{x}}', \mathbb{F}, \boldsymbol{\rho})$
17: **end function**

sum, $\mathbb{H}'$, of $\mathbb{H}$. $\mathbb{H}'$ is the cumulative sum of the elements of $\mathbb{H}$ along each row, from left to right. Thus,

$$\mathbb{H}' = \{H'_{kr}\}^{K \times max\{\boldsymbol{\Lambda}\}} \mid H'_{kr} = \sum_{r=1}^{r} H_{kr} \tag{5}$$

The total number, r, of components involved in the CCF of a type k component is equal to the index of the smallest element of the k^{th} row of matrix $\mathbb{H}'$ greater than or equal to U. This is expressed as,

$$r = min\{n, n+1, n+2, ..., max\{\boldsymbol{\Lambda}\}\} \mid H'_{kn} \geq U \tag{6}$$

If $r > 1$, $r-1$ components, excluding the one initiating the CCF, are randomly chosen from the set, $\boldsymbol{\rho}^{\{k\}}$, of components of type k. The selected components are those affected by the CCF event. The condition $r = 1$ denotes the scenario when no additional components are affected by the failure of the first component. On the other hand, if the cardinality of $\boldsymbol{\rho}^{\{k\}}$ is less than or equal to $r-1$, all the active type k components would fail. As in Algorithms 1 and 2, each failed component is removed from $\boldsymbol{\rho}^{\{k\}}$, with $\mathbb{F}$ and $\underline{\boldsymbol{x}}'$ updated accordingly. Algorithms 1 and 2 are also applied to the component, to ensure dual nodes and cascading failures are accounted for, respectively. Since all the components affected by the CCF event belong to the same component type (see the assumptions in this section), they can be removed from $\boldsymbol{\rho}^{\{k\}}$ and added to $\mathbb{F}$, each in just one step, without needing a *for* or *while* loop. Algorithm 3 summarises CCF propagation following the failure of a component. Its second line checks whether or not the component failure can induce CCF in other components. The condition $H'_{k1} = 1$ means type k components are not susceptible to CCF while $\boldsymbol{\rho}^{\{k\}} = \emptyset$ suggests none of these components is active, which is why the algorithm is terminated.

The Simulation Algorithm

Prior to simulation, each system component is assigned a positive integer, $i \mid i \in \{1, 2, ..., M\}$, representing its index in the system. The numbering starts with the components that actually make up the topology of the system, ending with external nodes. These components are then segregated into groups, according to their similarities. For the purpose of this work, the words, component and node, will be used interchangeably to refer to any element that effects the system.

Let $f_k(t)$ denote the common failure time distribution for all components of type k and $\boldsymbol{f}$, the global set containing $f_k(t)$ for all $k \in \{1, 2, ..., K\}$. To prepare the system for simulation, set the initial state vector, $\underline{\boldsymbol{x}}'$, to $\{M_1, M_2, ..., M_{K-1}, M_K\}$ and the set, $\mathbb{F}$, of failed components to $\emptyset$, since all the components are initially working. For the same reason, the survival signature, $\boldsymbol{S}_\tau\ (\underline{\boldsymbol{x}}')$, at the initial time step, $j_0 = 1$, is assigned a value

of 1. The set, $\boldsymbol{\rho}$, the CCF matrix, $\mathbb{H}$, the cascade matrix, $\mathbb{C}$, and the global set of duals, $\mathbb{D}$, are also defined. Finally, the mission time, T_m, is divided into equal time steps of magnitude δ, and the survival function, $R(t)$, preallocated as a vector of zeros, with each element corresponding to a time step. The survival function, in other words, is defined as, $R(t) = \{0\}^{n_t}$, where n_t is the number of time steps.

At time $t = 0$, the failure time of each of the $M + M'$ components of the system is sampled from its appropriate distribution and stored in $\boldsymbol{\tau}$. From $\boldsymbol{\tau}$, the next transition time, t, and the component, i, to fail are deduced. t is equivalent to the minimum element of $\boldsymbol{\tau}$ and i, its index in the set. The simulation is then shifted to this time, at which point the number of time steps, j, t represents is computer as $\lceil t/\delta \rceil$. Elements j_0 to j of the survival function, $R(t)$, are each incremented by $\boldsymbol{S}_\tau(\underline{\boldsymbol{x}}')$. The type, k, of the component is determined, the component is removed from $\boldsymbol{\rho}^{\{k\}}$, added to the set of failed components, and $\underline{\boldsymbol{x}}'$ modified to reflect the changes. Where necessary, Algorithms 1, 2, and 3 are invoked to remove dual nodes, cascade failures, and propagate CCF across the system, respectively. The next transition times of the failed components are set to infinity and j_0 set to $j + 1$. Again, the next transition time, t, and component, i, are determined and the cycle restarts. This procedure continues until $t > T_m$ or $\boldsymbol{S}_\tau(\underline{\boldsymbol{x}}') = 0$, which ever occurs first. The second condition is satisfied only when the non-repairable system is certainly failed, explaining why the simulation is terminated on its occurrence.

The sequence of events described in the preceding paragraph accounts for only one simulation sample. Since component failures are random in nature, this sequence should be repeated for an appreciable number of times. The effective survival function of the system is obtained by dividing the final value of $R(t)$ by the number of simulation samples, N. It is, however, worthwhile noting that the accuracy of the outcome is influenced by the values of N and δ. A large N and a small δ (relative to the mission time), guarantee an accurate $R(t)$. Algorithm 4 summarises the simulation procedure, which is different from the Algorithms proposed by Patelli and Feng[17], as the proposed procedure considers dependencies, as well as multiple failure modes. The block of code between lines 26 and 28 updates the survival function after the

Algorithm 4 System simulation procedure

Require: $\mathbb{H}'$, N, $\boldsymbol{\rho}$, $\boldsymbol{f}$, $\mathbb{C}$, $\mathbb{D}$

1: **function** SIMULATE($\mathbb{H}'$, N, $\boldsymbol{\rho}$, $\boldsymbol{f}$, $\mathbb{C}$, $\mathbb{D}$)
2: $\underline{\boldsymbol{x}}' \leftarrow \{M_1, M_2, ..., M_{K-1}, M_K\}$ ▷ Initialise $\underline{\boldsymbol{x}}$
3: $R(t) \leftarrow \{0\}^{n_t}$ ▷ Initialise survival function
4: $\boldsymbol{\tau} \leftarrow \{0\}^{M+M'}$ ▷ Initialise $\boldsymbol{\tau}$
5: $j_0 \leftarrow 1$ ▷ Define initial time step
6: $\mathbb{F} \leftarrow \emptyset$ ▷ Define initial set of failed components
7: $\boldsymbol{S}_\tau(\underline{\boldsymbol{x}}') \leftarrow 1$ ▷ Set survival signature to 1
8: **for** $k \leftarrow 1$ **to** K **do** ▷ Loop over component type
9: $\left(\boldsymbol{\tau}, \boldsymbol{\rho}^{\{k\}}\right) \leftarrow f_k(t)\text{Ⓢ}m_k$ ▷ Sample failures
10: **end for**
11: $[t, i] \leftarrow min\{\boldsymbol{\tau}\}$ ▷ Get next failure time and i
12: **while** $t \leq T_m$ **and** $\boldsymbol{S}_\tau(\underline{\boldsymbol{x}}') > 0$ **do**
13: $j \leftarrow \lceil t/\delta \rceil$ ▷ Get the number of time steps
14: $(R(t), j_0 \rightarrow j) \leftarrow (R(t), j_0 \rightarrow j) + \boldsymbol{S}_\tau(\underline{\boldsymbol{x}}')$
15: $k \leftarrow$component type of node i
16: $\boldsymbol{\rho}^{\{k\}} \leftarrow \boldsymbol{\rho}^{\{k\}} - i$ ▷ Remove i from set
17: $(\underline{\boldsymbol{x}}', k) \leftarrow \mid \boldsymbol{\rho}^{\{k\}} \mid$ ▷ Update state vector
18: $\mathbb{F} = \mathbb{F} \cup i$ ▷ Update set of failed nodes
19: $(\underline{\boldsymbol{x}}', \mathbb{F}, \boldsymbol{\rho}) \leftarrow$REMOVEDUALNODES($\underline{\boldsymbol{x}}', ..., i$)
20: $(\underline{\boldsymbol{x}}', \mathbb{F}, \boldsymbol{\rho}) \leftarrow$CASCADEFAILURE($\underline{\boldsymbol{x}}', ..., i$)
21: $(\underline{\boldsymbol{x}}', \mathbb{F}, \boldsymbol{\rho}) \leftarrow$PROPAGATECCF($\underline{\boldsymbol{x}}', ..., k$)
22: $(\boldsymbol{\tau}, \mathbb{F}) \leftarrow \infty$ ▷ Update transition times
23: $j_0 \leftarrow j + 1$ ▷ Set next initial time step
24: $[t, i] \leftarrow min\{\boldsymbol{\tau}\}$ ▷ Get next failure time and i
25: **end while**
26: **if** $j_0 \leq n$ **and** $\boldsymbol{S}_\tau(\underline{\boldsymbol{x}}') > 0$ **then**
27: $(R(t), j_0 \rightarrow n) \leftarrow (R(t), j_0 \rightarrow n) + \boldsymbol{S}_\tau(\underline{\boldsymbol{x}}')$
28: **end if**
29: Repeat lines 5 to 28 N times
30: $R(t) = \frac{R(t)}{N}$
31: **return** $(R(t))$
32: **end function**

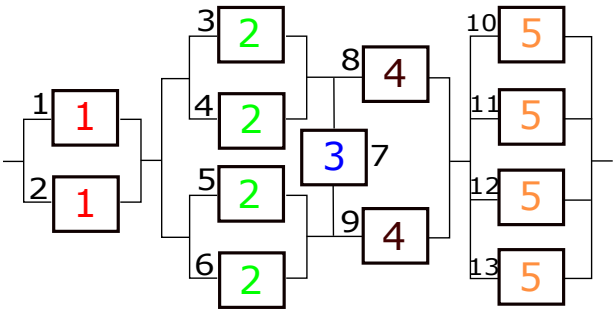

Figure 2. An arbitrary multi-component complex system

last component failure. The notation, $f_k(t)Ⓢ M_k$, on line 9 denotes M_k random failure times sampled from $f_k(t)$.

Sensitivity Analysis

Sensitivity analysis is the study of how the variation in the output of a mathematical model is apportioned to variations in its inputs.[25] In a complex system with multiple CCGs, each CCG may have a unique effect on the system's survivability. Therefore, the relative influence of the various CCGs on the reliability of the system is a vital decision-support information.[26]

Consider a complex system with K component types and its CCG matrix, $\mathbb{H}$. The computation of the relative sensitivity of the CCGs entails analysing the system with only one CCG active, for all the CCGs. To compute the effect of CCG i, H_{k1} is set to 1 and $H_{k2}, H_{k3}, ..., H_{kM_k}$ to 0, for all $k \neq i$. This is equivalent to replacing all rows other than i, of $\mathbb{H}$, with the y-element vector, $[1, 0, ..., 0]$, where $y = max\{\mathbf{\Lambda}\}$. The most critical CCG is the one producing the least deviation from the reliability of the system with all CCGs active. Since simulation provides the time-dependent system reliability, the proposed approach can reveal the evolution, over time, of the relative criticality of the CCGs. The sensitivity of this relative criticality to the Mean-Time-To-Failure (MTTF) of the component groups can also be investigated.

Case Studies

To illustrate how the proposed modelling and simulation approach is applied in practice, two case studies will be considered in this section. Though only numerical examples, the case studies have been carefully designed to reflect the every-day problems encountered by the system engineer in industry. We, therefore, believe they set the tone for the applicability of the proposed approach to realistic problems. To validate the approach, we compare the solutions to the existing analytical survival signature-based approach,[16,23] as well as a simulation approach based on a modification of the load-flow approach.[13,24] The modified load-flow simulator is the same as Algorithm 4, save for the replacement of $\boldsymbol{S}_\tau\ (\underline{\boldsymbol{x}}')$ with a structure function that is assigned the value 1 for non-zero flows across the system and 0, otherwise.

Case Study 1: A Complex Bridge System

Shown in Figure 2 is an arbitrary 13-component complex system, which components are arranged into five groups. The number within each box denotes which group the component belongs to while the number outside defines the index of the component in the system. Components of the same group are assumed to have the same failure time distribution, as defined in Table 1. The CCF parameters define the probabilities of a given number of components being affected by a CCF event and correspond to the α-factor CCF model. In the table, an exponential distribution is defined by its mean (in hours) while a Weibull distribution is defined by a set in which the first element is its scale parameter (in hours) and the second element, its shape parameter.

Analyses and Results: The system was first analysed with and without CCF, using the proposed simulation model and the data presented in Table 1. For this system, $\mathbf{\Lambda} = \{2, 4, 1, 2, 4\}$ and the CCF matrices, with

Table 1. Failure time distribution data and CCF parameters of component groups.

Component Type	Distribution Type	Distribution Parameters	CCF Parameters
1	Weibull	(1.8,2.2)	{0.95, 0.05}
2	Exponential	1.2	{0.8, 0.1, 0.05, 0.05}
3	Weibull	(2.3,1.6)	{1}
4	Weibull	(3.2,2.6)	{0.9, 0.1}
5	Exponential	2.1	{0.75, 0.1, 0.1, 0.05}

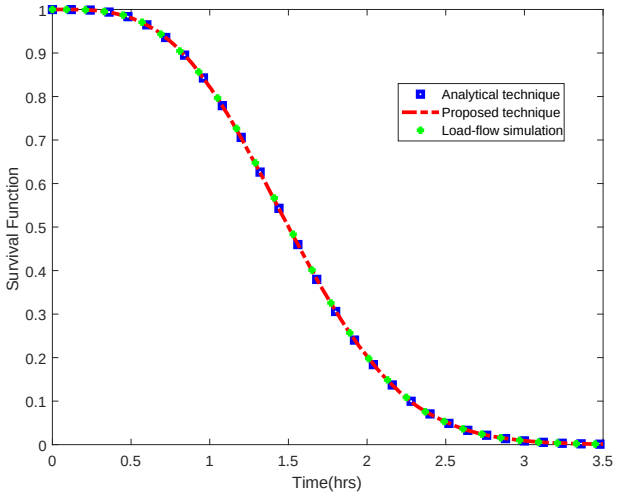

Figure 3. System reliability with dependencies ignored.

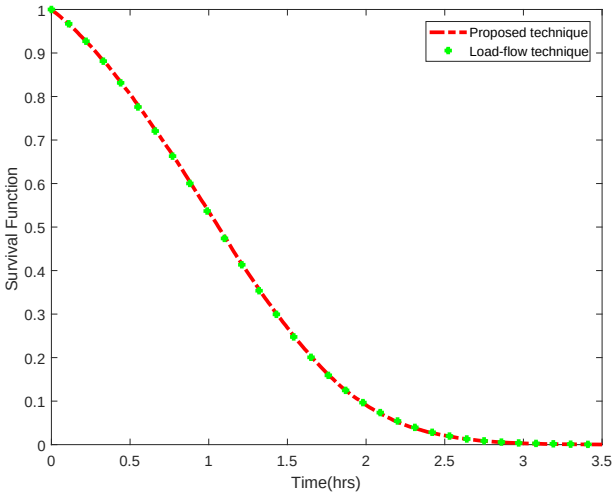

Figure 4. System reliability with dependencies considered.

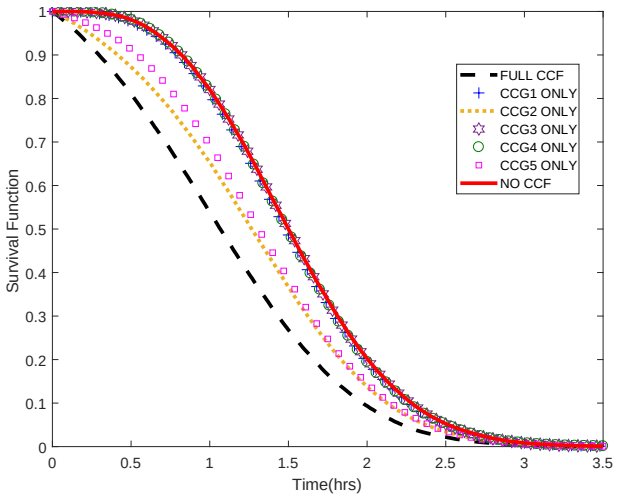

Figure 5. System reliability sensitivity to CCGs.

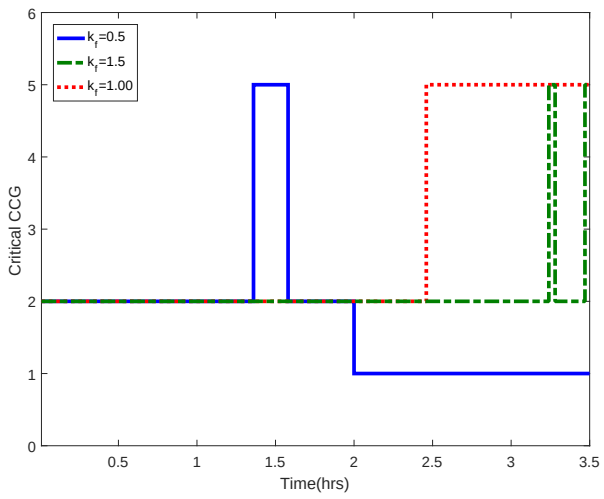

Figure 6. Sensitivity of critical CCG to component MTTF.

Table 2. Comparison between the computation time (in seconds) of the proposed and the existing techniques.

	Dependencies Ignored	**With Dependencies**
Survival Signature computation	92.00	92.00
Analytical technique	0.29	n/a
Proposed technique	32.50	32.80
Load-Flow simulation	2300.00	1654.00

and without CCF, are as expressed in Equations 7 and 8, respectively.

$$\mathbb{H} = \begin{pmatrix} 0.95 & 0.05 & 0 & 0 \\ 0.8 & 0.1 & 0.05 & 0.05 \\ 1 & 0 & 0 & 0 \\ 0.9 & 0.1 & 0 & 0 \\ 0.75 & 0.1 & 0.1 & 0.05 \end{pmatrix} \tag{7}$$

$$\mathbb{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{8}$$

The system was then re-analysed, in separate instances, using load-flow simulation and an analytical survival signature algorithm. The analytical algorithm was used only for the case without dependencies, due to its inapplicability to dependent systems. Figures 3 and 4 show the system reliability plots for a mission time of 3.5 hours and 5×10^4 samples.

The relative sensitivity of the system survival function to the Common-Cause Groups (CCG) was also investigated. The system was analysed considering CCF in all the CCGs (designated full CCG), with no CCF, and CCF in only one group at a time, for all the CCGs. The results obtained are plotted in Figure 5, from which the critical CCG is deduced, as described in Section "Sensitivity Analysis". Figure 6 shows the variation of this CCG with component Mean-Time-To-Failure (MTTF), as a function of time. The factor, k_f, denotes the number by which the nominal MTTF presented in Table 1 is multiplied. For an exponential distribution, the new mean becomes $\lambda_0 k_f$, where λ_0 is its nominal mean. The MTTF of a component with a failure time following a Weibull distribution is varied by keeping the shape parameter constant, while varying the scale parameter. If α_0 is the nominal scale parameter, the new scale parameter becomes $\alpha_0 k_f$.

Discussions: The accuracy and generality of the proposed simulation approach are validated by the plots in Figures 3 and 4, given the agreement between the results yielded by the various techniques. As highlighted in Figure 5, the reliability of the system reduces drastically when the effects of CCF are factored into the analysis. It exemplifies the need to consider this realistic aspect of a system's operation in its reliability evaluation. The figure also reveals CCG-2 as the most critical and CCG-3, the least critical. The latter, however, is not surprising, as CCG-3 is made up of only one component, inferring its immunity to CCF. Figure 6 shows that the criticality of a CCG may or may not remain fixed during the mission, depending on the MTTF of its components. For instance, for $k_f = 1$, corresponding to the nominal values presented in Table 1, CCG-2 is initially the most critical until at time, $t = 2.5$ hours, when it is overtaken by CCG-5. A different trend, however, is realised with $k_f = 0.5$, for instance. The essence of the results presented in Figure 6 could be appreciated on two fronts;

1. Since the most critical CCG is a function of the MTTF of its components, there is sufficient incentive for the system operator to re-identify the most critical CCG after every component replacement.
2. In the face of limited resources, the operator can efficiently allocate CCF mitigating resources during the mission. For instance, with $k_f = 0.5$, resources should be allocated to CCG-2 for $0 \leq t \leq 1.36$, CCG-5 for $1.36 \leq t \leq 1.57$, and CCG-1 for $t \geq 2$.

Though the proposed approach yields the same outcome as the load-flow simulation and analytical techniques, it requires less computational effort than the former but more than the latter, as summarised in Table 2. The table provides the recorded wall clock times (in seconds) for each approach, when the system was analysed on a 2GHz Intel(R) Core(TM) i5-4590T computer. The system with dependencies is less reliable, and, therefore, has a shorter life span. Since a simulation sample is terminated when the system is completely failed or the mission is completed, more samples are computed within a given wall clock time. It is, therefore, easy to realise that the difference in computation time between the system with dependencies and that with dependencies ignored, depends on the mission time. Row 1 of the table provides the time it took to derive the survival signature of the system, for all its possible state vectors. This time is fixed, with or without dependencies, since the survival signature depends only on the system topology.

In survival signature-based techniques, the structure function of a system is computed once for each of its state vectors. In load-flow simulation, however, the simulation computes the flow through the system for every component failure. Therefore, there could be up to N load-flow calculations per state vector, in an N sample simulation. This explains why the proposed approach

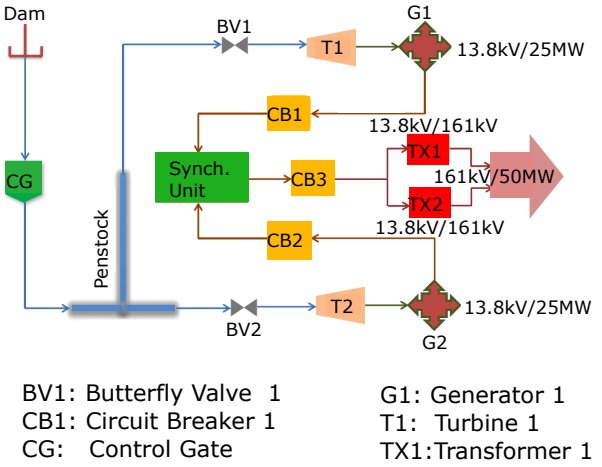

Figure 7. Schematic of a 50MW hydroelectric power plant.

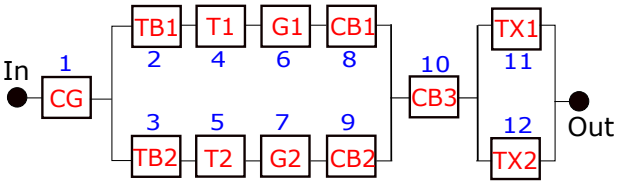

Figure 8. Condensed block diagram of the plant in Figure 7.

is more efficient, even when both are simulation-based. Its computational superiority is most appreciated when employed to solve a problem requiring repeated system evaluations. For instance, if the system under consideration were analysed n times with dependencies ignored, the proposed approach would take $92 + 32.5n$ seconds and the load-flow simulation, $2300n$ seconds.

Case Study 2: A Hydroelectric Power Plant

In this case-study, a two-unit hydroelectric power plant adapted from Ref.[27], and which schematic is shown in Figure 7, is analysed. It is a slightly modified model of the Bumbuna hydroelectric power plant; a 50MW plant in Sierra Leone. Its two units are similar, and each, rated 25MW consists a butterfly valve, turbine, generator,

Table 3. Failure time distribution data and CCF parameters of plant component groups.

Component Type	Components	Distribution Type	Distribution Parameters	CCF Parameters
1	1	Weibull	$(3, 1.8)$	$\{1, 0\}$
2	2,3	Weibull	$(1.8, 2.3)$	$\{0.9, 0.1\}$
3	4,5	Weibull	$(4, 3)$	$\{0.8, 0.2\}$
4	6,7	Weibull	$(2.1, 2.6)$	$\{0.85, 0.15\}$
5	8,9	Exponential	4	$\{0.8, 0.2\}$
6	10	Exponential	3.85	$\{1, 0\}$
7	11,12	Gamma	$(3, 1)$	$\{0.82, 0.18\}$
8	13	Hazard Function	$2t$	$\{1, 0\}$
9	14	Hazard Function	$t^2 + t/100$	$\{1, 0\}$

Table 4. Comparison between the computation time (in seconds) of the proposed approach and the existing techniques.

	Dependencies Ignored	With Dependencies
Survival Signature	34.20	34.20
Analytical technique	0.07	n/a
Proposed technique	29.00	32.80
Load-Flow simulation	757.70	486.40

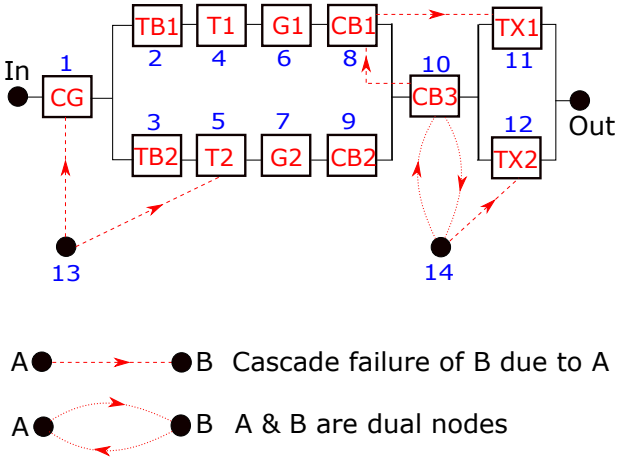

Figure 9. Plant block diagram showing interdependencies.

and circuit breaker. The power generated by the units is synchronized in the synchronizing unit and fed to the step-up transformers for onward transmission. The reliability block diagram of the plant is presented in Figure 8, where the Penstock and Synchronising unit have been neglected due to their very high reliability. CB3, which has two failure modes, is of a make different from that of CB1 and CB2. Its failure in mode 1 forces the failure of CB1 while its failure in mode 2 forces the failure of TX2. The dam is contaminated with impurities that induce failure in CG and T2, at a rate of $2t$ per year, where t is the time spent in operation. T1, however, is conservatively assumed immune to this impurity. The goal of this case study is to compute the reliability of the plant for a mission time of two years.

Analyses and Results: The impurity affecting CG & T2 and the second failure mode of CB3 can each be represented by an external node, as proposed in Section "Components with Multiple Failure Modes". The impurity is assigned a component ID of 13 and failure mode 2 of CB3, a component ID of 14. The latter is also the dual of node 10, since they both represent different failure modes of the same component, CB3. In Figure 9 is the final block diagram of the plant showing cascading dependencies. The components, including the external nodes have been organised into 9 component groups/types

depending on their similarity in functionality, make, and failure characteristics. Table 3 presents these component groups, their composition, failure time distribution (in years), and CCF parameters. In the table, "Hazard Function" as a distribution type suggests only the hazard rate, $h_k(t)$, of failures is known for that component type. Given $h_k(t)$, however, the probability density function, $f_k(t)$, and the cumulative density function, $F_k(t)$, for type k can be obtained thus;

$$\begin{aligned} f_k(t) &= h_k(t)\, e^{-\int_0^t h_k(u)du} \\ F_k(t) &= 1 - e^{-\int_0^t h_k(u)du} \end{aligned} \tag{9}$$

$$\mathbb{H} = \begin{pmatrix} 1 & 0 \\ 0.9 & 0.1 \\ 0.8 & 02 \\ 0.85 & 0.15 \\ 0.8 & 0.2 \\ 1 & 0 \\ 0.82 & 0.18 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \tag{10}$$

For the plant, $\mathbf{\Lambda} = \{1,2,2,2,2,2,2,1,1\}$, $\mathbb{D}_{10} = \{14\}$, $\mathbb{D}_{14} = \{10\}$, and $\mathbb{D}_i = \emptyset \forall i \notin \{10,14\}$. The CCF matrix is given by Equation 10 and the cascade matrix, $\mathbb{C}$, which is defined as a sparse matrix to conserve memory, by,

$$\begin{matrix} (8,11) & 1 \\ (10,8) & 1 \\ (13,1) & 1 \\ (13,5) & 1 \\ (14,12) & 1 \end{matrix}$$

As in the first case study, the plant was first analysed using 5×10^4 simulation samples and the same reliability modelling and evaluation techniques used in that case

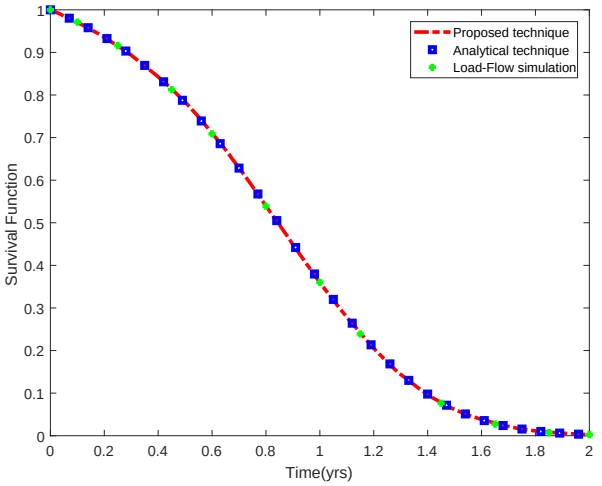

Figure 10. Plant reliability with dependencies ignored.

study, with dependencies neglected. It was then re-analysed using the proposed technique and load-flow simulation, but this time considering CCF and cascading failures. The computation time in each case, in seconds of wall clock time, was recorded as presented in Table 4 while Figures 10 and 11 show the plots of the results obtained. The plant was also analysed separately, with CCF neglected (denoted "Cascade only") and then with cascading failures neglected (denoted "CCF only"). The results obtained were plotted on the same axes, as shown in Figure 12, to lay bare, the effects of dependencies, as well as the relative influence of CCF and cascading failures on the reliability of the plant.

The plant was re-analysed with 5×10^3 samples using the load-flow and the proposed techniques in order to investigate the uncertainties in the simulation approaches and how the variance of the estimators vary with the number of samples. The proposed approach took a wall-clock time of 6.95s with dependencies ignored and 5.22s, with dependencies. Similar to the Case Study 1, the system with dependencies is less reliable, and, therefore less transitions are required to analyse the system.

Table 5. Comparison between the accuracy of the proposed approach and the existing techniques.A, B, and C represent the analytical approach, the proposed method, and the load-flow technique, respectively.

		System Reliability								
	Number of Samples	0.5 yrs			1.0 yrs			1.5 yrs		
		A	B	C	A	B	C	A	B	C
Without Dependencies	5×10^3	0.8152	0.8247	0.8132	0.5083	0.5048	0.5100	0.1938	0.1882	0.1941
	5×10^4	0.8152	0.8156	0.8159	0.5083	0.5092	0.5078	0.1938	0.1934	0.1940
With Dependencies	5×10^3	n/a	0.5767	0.5778	n/a	0.1193	0.1134	n/a	0.0070	0.0036
	5×10^4	n/a	0.5717	0.5714	n/a	0.1145	0.1133	n/a	0.0050	0.0050
		Coefficient of Variation (%)								
Without Dependencies	5×10^3	n/a	0.49	0.67	n/a	0.70	2.22	n/a	1.10	2.05
	5×10^4	n/a	0.33	0.33	n/a	0.52	0.76	n/a	0.72	1.47
With Dependencies	5×10^3	n/a	0.45	0.85	n/a	0.73	1.60	n/a	2.44	4.65
	5×10^4	n/a	0.30	0.36	n/a	0.38	1.00	n/a	0.89	2.23

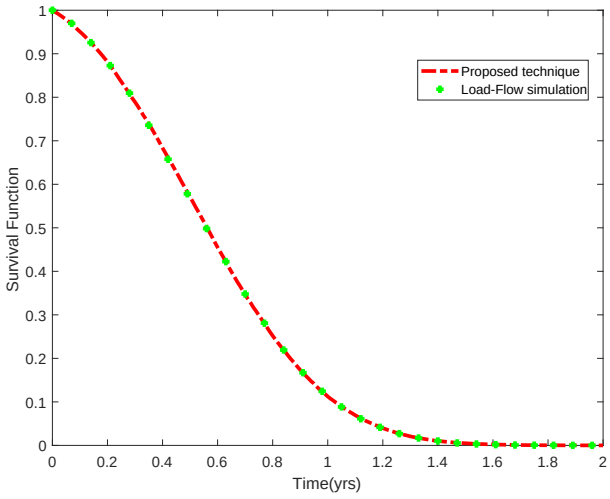

Figure 11. Plant reliability with dependencies considered.

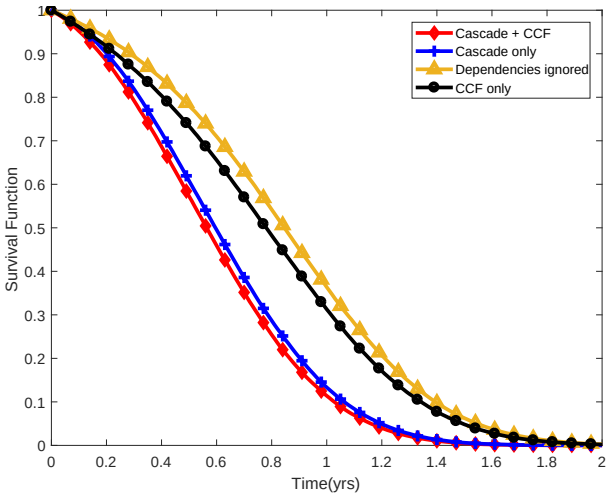

Figure 12. The effects of dependencies on plant reliability.

With the same number of initial samples, the load-flow approach took 25.15s and 13.66s, without and with dependencies, respectively. The reliability of the plant at 0.5, 1.0, and 1.5 years into the mission were deduced from the results and compared with the values obtained with 5×10^4 samples, as shown in Table 5. The coefficient of variation has been used as a measure of uncertainty in the system reliability at the selected times chosen to cover as much as possible the useful life of the system. The effect of the number of simulation samples on the variance of the reliability etimator is shown in Figure 13. The plots are based on the proposed approach applied to the system with dependencies ignored. A similar trend is portrayed by the system with dependencies considered - which observation is also true for the load-flow approach.

Discussions: Figures 10, 11, and Table 5 further validate the accuracy and generality of the proposed approach. As in case study 1, Table 4 shows the proposed approach to be somewhere between the analytical and load-flow techniques, in terms of computational efficiency. This assertion, too, is further ascertained by the outcome yielded with 5×10^3 simulation samples. As expected,

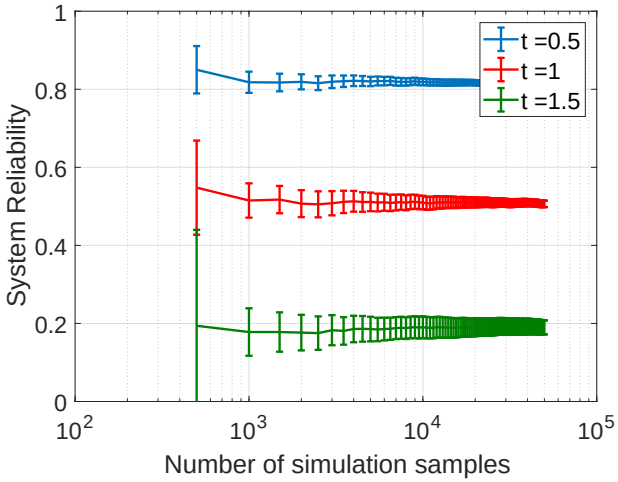

Figure 13. Variation of uncertainty in system reliability with number of simulation samples, plotted on a log scale.

the reliability of the plant is maximum, with dependencies neglected and least, otherwise. The effects of cascading failures, however, are prominent, according to Figure 12. This can be attributed to the effects of the contamination of the dam on the control gate (CG), since the failure of the latter induces the failure of the plant.

It can be deduced from Table 5 that both the proposed and load-flow approaches improve in accuracy with increase in the number of simulation samples used. Their inaccuracy, however, is prominent in the tail region of the system reliability plot. This, in our opinion, can be best explained by the fact that toward the end of the useful life of the system, its survival is a rare event. Its survival function in this region, therefore, requires a large number of samples for an accurate estimate. In essence, the optimal number of samples, as shown in Figure 13, would depend, amongst other factors, on the time (s) into the mission for which the analyst is computing the system reliability. Ensuring system analysis with an optimal number of samples requires first defining a threshold uncertainty for the result. The system is then analysed with an initial number of samples, and the uncertainty in the result computed. If this value is below the threshold, the analysis is terminated, otherwise the number of samples is increased, and the cycle is restarted.

Table 5 also reveals that for the same number of simulation samples and time into the mission, the coefficient of variation of the proposed approach is smaller than that of the load-flow approach. This implies that the system reliability value yielded by the former is more reliable than that yielded by the latter and that for a given threshold uncertainty, the proposed approach requires a fewer simulation samples to converge.

Conclusion

Dependent failures can impose adverse effects on the reliability and performance of a multi-component system.

In this work, we have proposed an approach that extends the applicability of the system survival signature-based approach to system reliability evaluation to systems susceptible to dependent failures. Being a hybrid technique, it inherits the desirable attributes of both the system survival signature and Monte Carlo Simulation. Consequently, it overcomes the issues of topological complexity, diversity in component failure time distributions, and complexities in inter-component interactions. Since the survival signature of a system is computed prior to its reliability evaluation and given this signature is static for a fixed system topology, the proposed approach is ideal for reliability/maintenance optimization and sensitivity/uncertainty analyses. The approach has been shown to be computationally efficient, albeit less efficient than the analytical approach, which, however, is inapplicable to systems with dependent failures. This leaves the proposed approach the most efficient alternative for realistic engineering systems.

We have shown how the approach is used to obtain key system reliability indices. Knowledge of the most critical Common-Cause Group, for instance, and how it varies with time and component Mean-Time-To-Failure, could influence the limited resource allocation in the mitigation of Common-Cause Failures. The proposed approach, therefore, can be used as a decision-support tool in the operation and management of complex systems.

Acknowledgement

This work was partially supported by the EPSRC grant Smart on-line monitoring for nuclear power plants (SMART) (EP/M018415/1) and the EPSRC and ESRC Centre for Doctoral Training on Quantification and Management of Risk & Uncertainty in Complex Systems & Environments (EP/L015927/1).

Declaration of Conflicting Interests

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Mosleh A, Rasmuson DM and Marshall FM. Guidelines on modeling Commom-Cause Failures in probabilistic risk assessment. Technical Report NUREG/CR-5485, U.S. Nuclear Regulatory Commission, 1998.
2. Mosleh A. Common cause failures: an analysis methodology and examples. *Reliability Engineering & System Safety* 1991; 34(3): 249–292.
3. Mosleh A, Fleming K, Parry G et al. Procedures for treating common cause failures in safety and reliability studies: Volume 1, procedural framework and examples: Final report. Technical report, Pickard, Lowe and Garrick, Inc., Newport Beach, CA (USA), 1988.
4. Dhillon B and Anude O. Common-cause failures in engineering systems: A review. *International Journal of Reliability, Quality and Safety Engineering* 1994; 1(01): 103–129.
5. Rasmuson DM and Kelly DL. Common-cause failure analysis in event assessment. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 2008; 222(4): 521–532.
6. Rausand M and Arnljot H. *System reliability theory: models, statistical methods, and applications*, volume 396. John Wiley & Sons, 2004.
7. Troffaes MC, Walter G and Kelly D. A robust bayesian approach to modeling epistemic uncertainty in common-cause failure models. *Reliability Engineering & System Safety* 2014; 125: 13–21.
8. Fan M, Zeng Z, Zio E et al. A stochastic hybrid systems model of common-cause failures of degrading components. *Reliability Engineering & System Safety* 2018; 172: 159 – 170. DOI:https://doi.org/10.1016/j.ress.2017.12.003. URL http://www.sciencedirect.com/science/article/pii/S0951832017304143.
9. O'Connor A and Mosleh A. A general cause based methodology for analysis of common cause and dependent failures in system risk and reliability assessments. *Reliability Engineering & System Safety* 2016; 145: 341 – 350. DOI:https://doi.org/10.1016/j.ress.2015.06.007. URL http://www.sciencedirect.com/science/article/pii/S0951832015001829.
10. Ramirez-Marquez JE and Coit DW. Optimization of system reliability in the presence of common cause failures. *Reliability Engineering & System Safety* 2007; 92(10): 1421 – 1434. DOI:https://doi.org/10.1016/j.ress.2006.09.004. URL http://www.sciencedirect.com/science/article/pii/S0951832006001992.
11. Rocchetta R, Zio E and Patelli E. A power-flow emulator approach for resilience assessment of repairable power grids subject to weather-induced failures and data

deficiency. *Applied Energy* 2018; 210: 339–350. DOI: https://doi.org/10.1016/j.apenergy.2017.10.126.

12. Ouyang M. Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability Engineering & System Safety* 2014; 121: 43 – 60. DOI: http://dx.doi.org/10.1016/j.ress.2013.06.040.
13. George-Williams H and Patelli E. Efficient availability assessment of reconfigurable multi-state systems with interdependencies. *Reliability Engineering and System Safety* 2017; 15: 431–444. DOI:10.1016/j.ress.2017.05.010.
14. Aslett LJ, Coolen FP and Wilson SP. Bayesian inference for reliability of systems and networks using the survival signature. *Risk Analysis* 2015; 35(3): 1640–1651.
15. Reed S. An efficient algorithm for exact computation of system and survival signatures using binary decision diagrams. *Reliability Engineering & System Safety* 2017; 165: 257–267.
16. Feng G, Patelli E, Beer M et al. Imprecise system reliability and component importance based on survival signature. *Reliability Engineering & System Safety* 2016; 150: 116 – 125. DOI:http://dx.doi.org/10.1016/j.ress.2016.01.019.
17. Patelli E, Feng G, Coolen FP et al. Simulation methods for system reliability using the survival signature. *Reliability Engineering & System Safety* 2017; 167: 327–337.
18. Coolen FP and Coolen-Maturi T. Predictive inference for system reliability after common-cause component failures. *Reliability Engineering & System Safety* 2015; 135: 27 – 33. DOI:https://doi.org/10.1016/j.ress.2014.11.005. URL http://www.sciencedirect.com/science/article/pii/S0951832014002774.
19. Eryilmaz S, Coolen FP and Coolen-Maturi T. Marginal and joint reliability importance based on survival signature. *Reliability Engineering & System Safety* 2018; 172: 118 – 128. DOI:https://doi.org/10.1016/j.ress.2017.12.002. URL http://www.sciencedirect.com/science/article/pii/S0951832017311183.
20. Eryilmaz S, Coolen FP and Coolen-Maturi T. Mean residual life of coherent systems consisting of multiple types of dependent components. *Naval Research Logistics (NRL)* ; : n/a–n/aDOI:10.1002/nav.21782. URL http://dx.doi.org/10.1002/nav.21782.
21. Eryilmaz S. The concept of weak exchangeability and its applications. *Metrika* 2017; 80(3): 259–271. DOI:10.1007/s00184-016-0602-z. URL https://doi.org/10.1007/s00184-016-0602-z.
22. Samaniego FJ. *System Signatures and their Applications in Engineering Reliability*. Springer, New York, 2007.
23. Coolen FPA and Coolen-Maturi T. Generalizing the signature to systems with multiple types of components. In Zamojski W, Mazurkiewicz J, Sugier J et al. (eds.) *Complex Systems and Dependability*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-30662-4, pp. 115–130.
24. George-Williams H and Patelli E. A hybrid load flow and event driven simulation approach to multi-state system reliability evaluation. *Reliability Engineering & System Safety* 2016; 152: 351 – 367. DOI:http://dx.doi.org/10.1016/j.ress.2016.04.002.
25. Borgonovo E, Marseguerra M and Zio E. A monte carlo methodological approach to plant availability modeling with maintenance, aging and obsolescence. *Reliability Engineering & System Safety* 2000; 67(1): 61 – 73. DOI: http://dx.doi.org/10.1016/S0951-8320(99)00046-0.
26. Levitin G. *The Universal Generating Function in Reliability Analysis and Optimization*. Springer-Verlag London Limited, 2005.
27. George-Williams H and Patelli E. Maintenance strategy optimization for complex power systems susceptible to maintenance delays and operational dynamics. *IEEE Transactions on Reliability* 2017; PP(99): 1–22. DOI: 10.1109/TR.2017.2738447.