# Quantitative Analysis of Dynamic Fault Trees Based on the Coupling of Structure Functions and Monte Carlo Simulation

G Merle, J.-M Roussel, J.-J Lesage, V Perchet, N Vayatis

# Quantitative Analysis of Dynamic Fault Trees Based on the Coupling of Structure Functions and Monte Carlo Simulation

G. Merle*†[1], J.-M. Roussel[1], J.-J. Lesage[1], V. Perchet[2] and N. Vayatis[3]

[1]LURPA, ENS Cachan, 61 Avenue du Président Wilson, Cachan, 94230, France
[2]LPMA, Denis Diderot University, 75205, Paris, France
[3]CMLA, ENS Cachan, 61 Avenue du Président Wilson, Cachan, 94230, France

**This paper focuses on the quantitative analysis of Dynamic Fault Trees (DFTs) by means of Monte Carlo simulation. In a previous article, we defined an algebraic framework allowing to determine the structure function of DFTs. We exploit this structure function and the minimal cut sequences that it allows to determine, to know the failure mode configuration of the system, which is an input of Monte Carlo simulation. We show that the results obtained are in good accordance with theoretical results and that some results, such as importance measures and sensitivity indexes, are not provided by common quantitative analysis and yet interesting. We finally illustrate our approach on a DFT example from the literature.**

**Keywords:** Dynamic Fault Tree; quantitative analysis; Monte Carlo simulation

## 1 Introduction

Fault Tree Analysis (FTA) is one of the oldest and most diffused techniques in industrial applications, for the dependability analysis of large safety-critical systems.[1-3] FTA generally consists in two types of analyses. On the one hand, a qualitative analysis allows to determine the list of all the possible combinations of events that lead to the top event (TE), which are called the minimal cut sets of the FT. On the other hand, a quantitative level allows to determine the probability of occurrence of the TE of the FT, based on the time-to-failure probability distributions of its basic events.

The original FTs contain only Boolean OR/AND gates, and only the combination of failures of basic events is relevant and not their sequence. As they do not allow to model the failure mechanisms of sequence-sensitive dynamic systems, they are commonly called *Static Fault Trees* (SFTs). As the knowledge of the working or failing state of each component is sufficient to determine the state of the system, a Boolean model of events is generally retained for SFTs and Boolean gates can easily be modelled by means of Boolean operators OR and AND. An expression for the *TE* of the SFT can hence be determined as a function of basic events: this expression is called the *structure function* of the SFT. Such an expression can always be reduced to a minimal sum-of-product canonical form thanks to the theorems of Boolean algebras, and both the qualitative and quantitative analyses can be performed directly from this minimal canonical form.

Several attempts have been reported in the literature to remove some modelling limits of SFTs and include various kinds of temporal and statistical dependencies in the model. A Priority-AND (PAND) gate has been introduced by Fussel *et al.*[4] to model situations in which the failure of the gate occurs if the inputs fail in an

*Correspondence to: G. Merle, LURPA, ENS Cachan, 61 Avenue du Président Wilson, Cachan, 94230, France
†E-mail: guillaume.merle@gmail.com

order. This gate was later included in the Dynamic Fault Tree (DFT) model that Dugan *et al.* proposed.[5,6] The DFT is based on the definition of new gates that induce temporal as well as statistical dependencies: PAND, Functional Dependency (FDEP), Warm Spare (WSP) and Sequence Enforcing (SEQ). As the TE of DFTs can be engendered not only by combinations but also by sequences of event failures, the concept of minimal cut sets was hence extended to the concept of minimal cut sequences.[7] However, the lack of a behavioural model for these gates made impossible the determination of a structure function for DFTs, and other approaches – mainly based on state models – were developed to perform the analyses. On the one hand, qualitative temporal analysis[8] and zero-suppressed binary decision diagrams[7] were retained to perform the qualitative analysis of DFTs. On the other hand, continuous time Markov chains,[9,10] Stochastic Petri Nets[11,12] and temporal Bayesian networks[13] were retained to perform the quantitative analysis of DFTs. However, all these approaches present some limits in terms of accuracy of the results obtained or of time-to-failure distributions which can be considered.

To remove these limits, in a previous article, we presented an algebraic framework allowing to algebraically model dynamic gates and determine the structure function of any DFT.[14] We also showed that the minimal cut sets and sequences of DFTs can be determined directly from this structure function, in the same way that the minimal cut sets of SFTs can be determined directly from their structure function. Based on the structure function and on a probabilistic model of all dynamic gates, we also proposed in the work of Merle *et al.*[15] an analytical approach for the quantitative analysis of DFTs. The main advantage of this approach is that it allows to consider any kind of failure distribution for basic events, because the probabilistic models of dynamic gates do not depend on this distribution. Nevertheless, for large-scale systems, the underlying calculations are quite important and remain difficult to handle for practitioners. On the other hand, Monte Carlo simulation is often used for DFT quantitative analysis, and especially when the use of non-exponential distributions is needed for a realistic modelling of failures in a system.[16] Although Monte Carlo simulation is within the reach of practitioners (at least more than analytical calculus), it remains time-consuming when solving large-scale problems. In this paper, we propose a new approach that aims at cumulating the advantages of both approaches by performing Monte Carlo simulation onto the only minimal cut sequences extracted from the structure function and not onto the whole DFT.

This paper is organised as follows. The main approaches commonly used to perform the quantitative analysis of DFTs are reviewed in Section 2. Then, our approach based on the coupling of structure functions and Monte Carlo simulation is presented in Section 3. Finally, we illustrate our approach on a DFT example in Section 4.

## 2 State of the art

Many approaches have been envisaged to perform the quantitative analysis of DFTs. In the work of Tang and Dugan,[7] each dynamic gate of the considered DFT is replaced by the static gate corresponding to its logic constraints; the minimal cut sets of the resulting SFT are then generated by using zero-suppressed binary decision diagrams, and these minimal cut sets are expanded to minimal cut sequences by considering the time constraints. However, it can be noted that some constraints cannot be taken into account during this conversion of dynamic gates into static gates as this conversion leads to a super set of sequences for the qualitative analysis: we showed in the work of Merle[17] that, during the conversion of many Spare gates sharing a spare event into static gates, the behaviour of the spare event cannot be correctly taken into account. Coppit *et al.*[18] propose to convert the DFT into a failure automaton that models the changing state of the system as failures occur. This failure automaton can then be converted into a continuous time Markov chain (CTMC), and the solution of the corresponding set of differential equations allows to determine the failure probability of the TE of the DFT. These two approaches have been implemented in the Galileo tool.[6]

Other model-based approaches also allow to perform the quantitative analysis of DFTs. For instance, in the work of Montani *et al.*,[19] the whole DFT is converted into a dynamic Bayesian network, and the failure probability of the TE of the DFT can be determined by using inference algorithms. Finally, in the work of Bobbio and Codetta Raiteri,[11] the dynamic subtrees of DFTs are converted into a class of coloured Stochastic Petri Nets called Stochastic Well-formed Net (SWN). This SWN can be converted into a CTMC to determine the failure probability of the TE of the dynamic subtree, and this failure probability can then be cast back into the original DFT. These two approaches have been respectively implemented in the Windows[19] and Linux[20] version of the Drawnet tool.

All these approaches can provide a literal result, but the types of time-to-failure distributions that can be considered for basic events are limited. On the one hand, CTMC-based approaches can only take into account exponential distributions. On the other hand, there is no theory for exact Bayesian network inference with general distributions, and theory exists only in the case of Gaussian distributions[21] and mixtures of truncated exponentials.[22] In both cases, the state space becomes too large for calculation when the number of gate inputs increases.[16] This is the reason why Monte Carlo simulation has become more and more used to perform the quantitative analysis of DFTs over the years, as it allows to obtain an approximate result for any distribution of basic events whilst eliminating the statistical independence assumption. Initially applied to SFTs only,[23] the extension of Monte Carlo simulation to DFTs was considered for the first time by Marsaguerra *et al.*[24] Over the years, it has been used to determine the availability, reliability and importance measure estimations of complex systems[25-27] such as multi-state systems, that is, systems with dependencies between the system state and the state of its components.

Recently, Monte Carlo simulation was made able to solve dynamic gates. In the work of Durga Rao,[16] a tool called Dynamic Reliability with SIMulation (DRSIM) and based on the Monte Carlo simulation approach is presented; dynamic gates are implemented in this tool by taking into account the sequences according to which components fail (for the PAND gate) as well as by accommodating the standby behaviour of spare components (for Spare gates). The results obtained thanks to DRSIM are in good agreement with the results commonly obtained by analytical approches, even if this approach is not subject to state space explosion and provides results with a lower computational time. Miao *et al.*[28] particularly focus on some structures called ring-standby structures, which are standby models in which backup components cannot replace all non-backup ones. Such a structure can easily be modelled by means of many Spare gates having both shared and unshared spare events. However, the authors do not compare the results obtained with their approach with the results obtained with other approaches. Finally, a library object called MatCarloRe and based on Simulink was proposed by Manno *et al.*,[29] allowing the user to construct a DFT model of the process thanks to the graphical interface of the MATLAB simulator by using the library blocks. Each such block carries the logic of a DFT gate, and the Monte Carlo engine collects the outputs of many runs and their agglomerate to construct significant statistics of interest. However, this tool does not allow to calculate differential importance measures. Differential importance measures represent the impact that each parameter of the system (e.g. the failure rate of a basic event) has on the failure of the whole system and hence represent a really useful result to know what component(s) to improve in priority to improve the reliability of the system considered.

The main goal of our approach is to perform the quantitative analysis of DFTs and the sensitivity analysis of their top events by coupling Monte Carlo simulation with the structure function of DFTs that we introduced in the work of Merle *et al.*,[14] and hence with the knowledge of the cut sequences and minimal cut sets of the DFT (i.e. the results of the qualitative analysis of the DFT). As the algebraic framework that we presented in the work of Merle *et al.*[14] allows to model all dynamic gates and is based on temporal operators, we can use Monte Carlo simulation to simulate the sequences according to which components will fail and the results of the qualitative analysis will allow to know, for each random sequence, whether the system fails or not. This approach is presented in Section 3.

# 3 Quantitative analysis of DFTs based on the coupling of structure functions and Monte Carlo simulation

## 3.1 Structure function of DFTs

We defined in the work of Merle *et al.*[14] three temporal operators named non-inclusive BEFORE (BF, noted $\lhd$), SIMULTANEOUS (SM, noted $\bigtriangleup$), and inclusive BEFORE (IBF, noted $\trianglelefteq$):

- the operator SM was introduced to take into account the simultaneity of occurrence of intermediate events that may happen in any DFT containing repeated events[30];
- the operator BF was introduced to model a strict version of the PAND gate, as well as all Spare gates;
- finally, the operator IBF was built thanks to operators SM and BF to model the non-strict version of the PAND gate that we retained in the work of Merle *et al.*[14]

The operators BF and IBF allowed us to define a behavioural model for all dynamic gates FDEP (which can be modelled by means of Boolean operators as we demonstrated in the work of Merle *et al.*[31]), PAND and Spare. We demonstrated that such a model allows to determine the structure function of any DFT and that this structure function can be expressed under a minimal canonical form thanks to a set of theorems that we provided. Because this minimal canonical form is a sum of terms, each term – that we called a cut sequences set – is an algebraic expression, which represents a condition that must hold for a sequence of occurrences to be a cut sequence and which hence allows to determine a set of minimal cut sequences. For instance, if the minimal canonical form of the structure function contains the algebraic term $C \cdot (A \lhd C) \cdot (B \lhd C)$, it means that any failure sequence in which the basic events $A$ and $B$ fail before $C$ is a cut sequence: $[A, B, C]$ and $[B, A, C]$ hence are minimal cut sequences. In the same way, if an expression in the structure function does not contain temporal operators, it allows to determine minimal cut sets and hence minimal cut sequences. For instance, if the minimal canonical form of the structure function contains the algebraic term $A \cdot B$, it means that any failure sequence in which the basic events $A$ and $B$ fail is a cut sequence: $[A, B]$ and $[B, A]$ will hence be minimal cut sequences.

The knowledge of all these minimal cut sequences can then be used in Monte Carlo simulation to determine, for each random failure sequence, whether the system fails or not, as illustrated in Section 3.2.

## 3.2 Monte Carlo simulation

Our approach is illustrated in Figure 1. On the one hand, as explained in Section 3.1, the algebraic framework that we presented in the work of Merle *et al.*[30] allows to determine the minimal canonical form of the structure function of any DFT. Such a minimal canonical form allows to determine the minimal cut sequences of the DFT, and either this structure function of the DFT or the minimal cut sequences can be used to generate an oracle that will be able to state, for any sequence of occurrences of basic events, whether the top event of the DFT occurs or not. On the other hand, we use Monte Carlo simulation to generate random sequences of occurrences.

Monte Carlo simulation is a very valuable method widely used in the solution of real engineering problems in many fields.[16] It allows to estimate the reliability indices by simulating the actual process and random behaviour of the system in a computer model to create a realistic lifetime scenario of the system. The only required information for the analysis is as follows[29]:

- the probability density function (pdf) of the time to failure of each component – noted $f_{b_i}(t)$ in Figure 1 – and their parameters values;
- the mission time $T$ of the system;
- the system failure mode configuration (which is modelled here by the oracle of the DFT).

Components are then simulated for the specified mission time for depicting the duration of their available state, which is random and will depend on the pdf of time to failure.

For instance, if we consider a random variable $X$ with a Markovian time-to-failure distribution with failure rate $\lambda$, its pdf and cumulative distribution function (cdf) are given by the following expressions:

$$
\begin{aligned}
f(x) &= \lambda e^{-\lambda x} \\
F(x) &= \int_0^x f(t)dt = 1 - e^{-\lambda x},
\end{aligned}
\tag{1}
$$

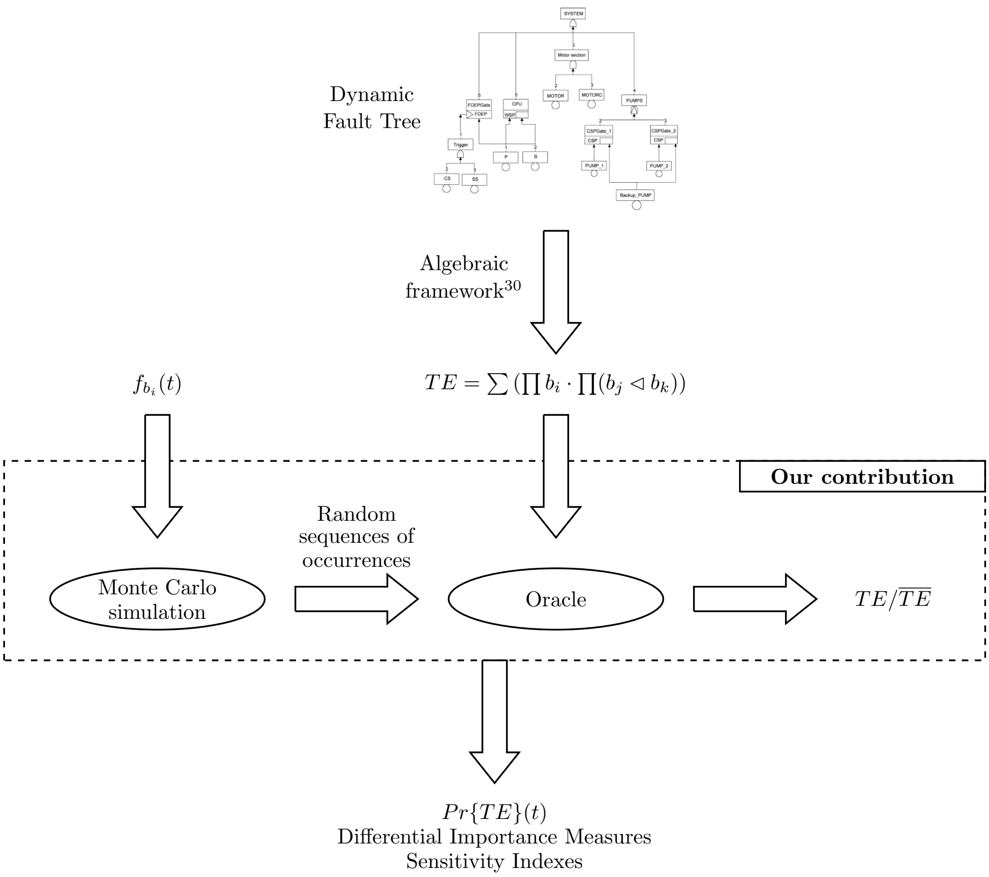

**Figure 1.** Monte Carlo simulation coupled with the structure function of DFTs.

when $x \geqslant 0$ and by $f(x) = F(x) = 0$ otherwise.

$x$ can then be expressed as a function of $F(x)$ as follows:

$$\begin{aligned} & F(x) = 1 - e^{-\lambda x} \\ \Leftrightarrow \quad & x = \frac{1}{\lambda} \ln\left(\frac{1}{1 - F(x)}\right) \end{aligned} \tag{2}$$

The pdf and cdf that we are considering in this paper are functions of time $t$ and a uniform random number can be generated by using any of the standard random number generators. Such a random number generator will hence allow to determine a random value for $F(t)$ in $[0;1]$, which can then be substituted in place of $F(t)$ in the expression $t = \dfrac{1}{\lambda} \ln\left(\dfrac{1}{1 - F(t)}\right)$ in order to obtain a random date of occurrence for the basic event considered. Similarly, if $X$ is a Weibull random variable with parameters $\lambda > 0$ (the scale parameter) and $k > 0$ (the shape parameter; when $k = 1$, the Weibull distribution corresponds to an exponential distribution), then its pdf and cdf are given by the following expressions:

$$\begin{aligned} f(x) &= \lambda k (\lambda x)^{k-1} e^{-(\lambda x)^k} \\ F(x) &= \int_0^x f(t)dt = 1 - e^{-(\lambda x)^k}. \end{aligned} \tag{3}$$

As a consequence, one readily obtains that

$$x = \frac{1}{\lambda}\left(\ln\left(\frac{1}{1-F(x)}\right)\right)^{\frac{1}{k}}$$

Finally, we recall that, with respect to these paremeters, the expectation of the random variable $X$ is $\lambda\Gamma\left(1+\frac{1}{k}\right)$, where $\Gamma(\cdot)$ is the usual Gamma function. For the specific choices of $k = 0.5$, 1 or 2, it is defined by $\Gamma(3) = 2$, $\Gamma(2) = 1$ and $\Gamma(1.5) = \frac{\sqrt{\pi}}{2}$.

In the same way, it can be noted that if a random variable $X$ has a uniform distribution and if $F$ is an invertible cdf, then the random variable $F^{-1}(X)$ has the cdf $F$.[32] This can be reproduced for all basic events in order to determine a random sequence of occurrences. This random sequence can then be reduced by comparing it with the mission time considered and by removing all the events that occurred after the mission time. Finally, these reduced random sequences of occurrence of basic events can be compared with the minimal cut sequences of the DFT to determine whether the top event of the DFT occurs or not. The failure probability of the system – noted $Pr\left\{TE\right\}(T)$ in Figure 1 – can then be determined as the ratio $\frac{n_{TE}}{N}$ between the number of times the top event occurs $n_{TE}$ and the number of simulations $N$.

The number of simulations $N$ that is needed to have a probability $\delta$ of returning a value that misses the correct index by more than $\varepsilon$ can be determined, thanks to Hoeffding's inequality[33], as

$$N \geqslant \frac{1}{2\varepsilon^2}\ln\left(\frac{2}{\delta}\right), \tag{4}$$

that is, the smaller $\delta$ will be, the more lucky we will be to obtain a result with the accuracy $\varepsilon$. It can be noted that Hoeffding's inequality allows to obtain a lower bound for the number $N$ as this probabilistic analysis is based on the worst case (which corresponds here to $Pr\left\{TE\right\} = \frac{1}{2}$).

Two types of analyses can also be carried out:

- a *global analysis*, which consists in analysing the differential importance measures of the parameters of the pdfs to know which component needs to be improved in priority;
- a *local analysis*, which consists in analysing the sensitivity with respect to a set of parameters, in order to know which components best represent the model.

These global and local analyses are shortly described in Sections 3.3 and 3.4, respectively.

## 3.3 Global analysis

Global analysis aims at knowing which parameters of the pdfs are important by determining how the failure probability of the system evolves depending on these parameters. If the system considered depends on $n$ parameters, for each parameter, the $(n-1)$ other parameters are left unchanged, whilst the value of the parameter considered varies inside a set of values to get an idea of its importance with respect to the global failure probability by simulation. Such an analysis can be performed in two ways:

- by doing a large number of simulations on a small set of values;
- by doing a small number of simulations on a large set of values.

The data obtained can then be classified in order to determine areas in which the probability is particularly high or low, and these areas can be ordered in order to know the relative importance of the parameters.

### 3.4 Local analysis

The main purposes of local analysis – i.e. of sensitivity analysis – are as follows:

- compare the mathematical model with the real system, to see whether the variables have the same impact in theory and in practice;
- minimize the error made on the output, by decreasing the error on influent variables or by modifying the model;
- simplify the model, by replacing non-influent variables by constants.

Theoretically, let $Y$ be the output variable and $X_1, \dots, X_n$ be the independent input variables of the system. If the relation between $Y$ and the inputs $X_i, i \in \{1, \dots, n\}$ is as follows:

$$Y = \beta_0 + \sum_{k=1}^{n} \beta_k X_k, \tag{5}$$

the variance of $Y$ can be determined as follows:

$$var\,[Y] = \sum_{k=1}^{n} \beta_k^2 var\,[X_k], \tag{6}$$

and we can hence determine the sensitivity indexes of variables $V_k$ as,

$$1 = \sum_{k=1}^{n} \beta_k^2 \frac{var\,[X_k]}{var\,[Y]} = \sum_{k=1}^{n} IS_k. \tag{7}$$

It can be noted that all these sensitivity indexes are positive numbers whose sum equals to 1. The importance of a variable can hence be determined by comparing its sensitivity index with respect to 1.

In the same way, if $Y = f(X_1, \dots, X_n)$ is a function of $n$ random variables $X_1, \dots, X_n$, there exists a family of mappings $f_{j_1,\dots,j_k}$, with $\{j_1, \dots, j_k\} \subset \{1, \dots, n\}$, that are orthogonal[1] with each other and such that

$$f(X_1, \dots, X_n) = f_0 + \sum_{k=1}^{n} f_k(X_k) + \sum_{k,l=1}^{n} f_{k,l}(X_k, X_l) + \dots + f_{1,\dots,n}(X_1, \dots, X_n). \tag{8}$$

From a probabilistic point of view, we hence have

$$Y = E\,[Y] + \left( \sum_{k=1}^{n} E\,[Y|X_k] - E\,[Y] \right) + \left( \sum_{k,l=1}^{n} \left( E\,[Y|X_k, X_l] - E\,[Y|X_k] - E\,[Y|X_l] \right) + E\,[Y] \right) + \dots, \tag{9}$$

which allows to determine the variance of $Y$ as

$$var\,[Y] = \sum_{k=1}^{n} V_k + \sum_{k,l=1}^{n} V_{k,l} + \dots + V_{1,\dots,n}, \tag{10}$$

where $V_k = var\,[E\,[Y|X_k]]$ is the part of $var\,[Y]$ which is due to $X_k$, $V_{k,l} = var\,[E\,[Y|X_k, X_l]] - var\,[E\,[Y|X_k]] - var\,[E\,[Y|X_l]]$ is the part of $var\,[Y]$ which is due to the interaction between $X_k$ and $X_l$ and which is not taken into account in $X_k$ and $X_l$, and so on.

This variance-based sensitivity analysis originated in the work of Cukier *et al.*[34] and Sobol.[35]

[1] According to Chastaing *et al.*,[38] two functions depending on random variables from $\mathbb{L}^2$ are orthogonal whenever all the variables involved in one of the functions also appear in the other.

# 4 Application to a DFT example

## 4.1 Presentation of the DFT example

The DFT example that we are going to use is the DFT of a Hypothetical Cardiac Assist System (HCAS) from the work of Boudali and Dugan,[13] which was inspired from a Cardiac Assist System found in the work of Vemuri *et al.*,[36] and whose structure can be found in the work of Ren and Dugan[37] and is shown in Figure 2, where TEDTS stands for transcutaneous energy and data transmission system.

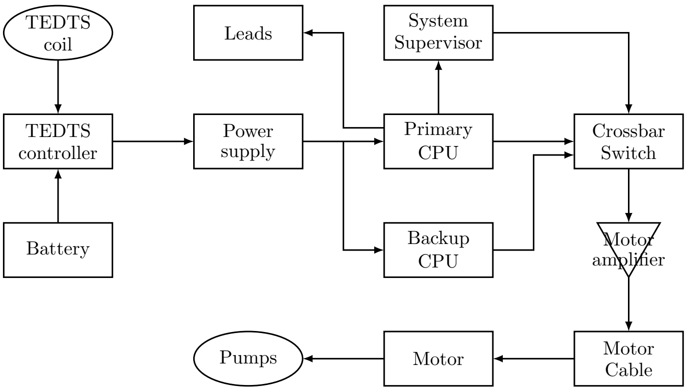

**Figure 2.** The Hypothetical Cardiac Assist System (HCAS).

The HCAS is designed to treat mechanical and electrical failures of the heart. The system can be divided into four modules: trigger, CPU unit, motor section, and pumps. The crossbar switch ($CS$) and the system supervisor ($SS$) represent the trigger, because the failure of either $CS$ or $SS$ triggers the failure of both CPUs. The CPU unit can be considered as a warm spare with a primary $P$ and a spare unit $B$ (which corresponds to the backup CPU). For the motor section to function, either the motor ($MOTOR$) or the motor cable ($MOTORC$) needs to be working. The pumps unit is composed of two cold spares, each having a primary pump ($PUMP\_1$ and $PUMP\_2$), and sharing a common spare pump ($Backup\_PUMP$). In order for the pumps unit to fail, all three pumps need to fail, and CSPGate_1 needs to fail before (or at the same time as) CSPGate_2.

The DFT that models the potential failure of the HCAS is shown in Figure 3.

## 4.2 Structure function and qualitative analysis of the DFT

The minimal canonical form of the structure function of the DFT in Figure 3 has been determined in the work of Merle *et al.*[14] and is

$$\begin{aligned} TE &= CS + SS + MOTOR \cdot MOTORC + P \cdot (B_d \lhd P) + B_a \cdot (P \lhd B_a) \\ &\quad + BP \cdot (P2 \lhd P1) \cdot (P1 \lhd BP) + P2 \cdot (P1 \lhd BP) \cdot (BP \lhd P2), \end{aligned} \tag{11}$$

where $P1$, $P2$, and $BP$ respectively stand for $PUMP\_1$, $PUMP\_2$, and $Backup\_PUMP$.

In Equation (11), the top event of the DFT is expressed as a sum of seven algebraic expressions. Amongst these seven algebraic expressions,

- three algebraic expressions do not contain the temporal operator $\lhd$: they hence are static expressions from which the minimal cut sets of the DFT can be extracted;
- four algebraic expressions contain the temporal operator $\lhd$: they hence are dynamic expressions from which the minimal cut sequences of the DFT can be extracted.

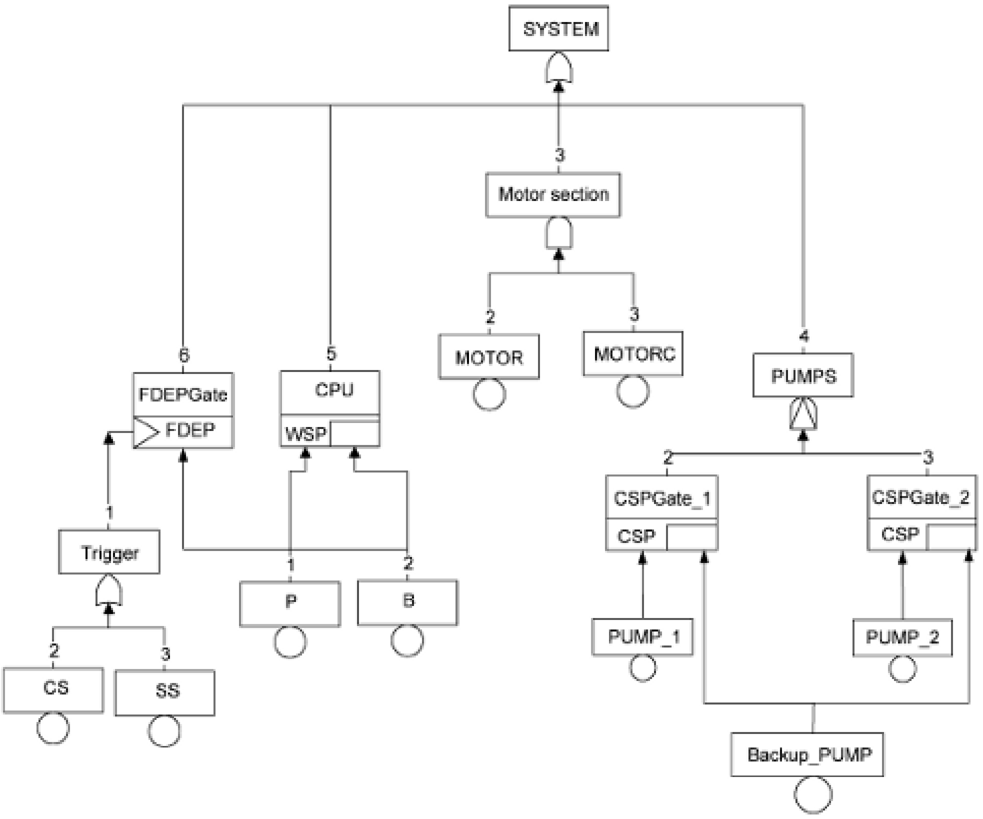

**Figure 3.** The DFT of the HCAS.

The minimal cut sets and sequences of the DFT can hence be determined as follows:

- the algebraic expressions $CS$, $SS$ and $MOTOR \cdot MOTORC$ allow to determine that $CS$, $SS$ and $MOTOR \cdot MOTORC$ are minimal cut sets for the DFT;
- the algebraic expression $P \cdot (B_d \lhd P)$ allows to determine that $[B_d, P]$ is a minimal cut sequence for the DFT;
- the algebraic expression $B_a \cdot (P \lhd B_a)$ allows to determine that $[P, B_a]$ is a minimal cut sequence for the DFT;
- the algebraic expression $BP \cdot (P2 \lhd P1) \cdot (P1 \lhd BP)$ allows to determine that $[P2, P1, BP]$ is a minimal cut sequence for the DFT;
- the algebraic expression $P2 \cdot (P1 \lhd BP) \cdot (BP \lhd P2)$ allows to determine that $[P1, BP, P2]$ is a minimal cut sequence for the DFT.

Sequences are noted between brackets and contain the basic events that occurred in the order in which they occurred. For instance, the sequence $[A, B]$ indicates that the basic events $A$ and $B$ failed and that $A$ failed before $B$. As we consider that basic events are statistically independent, two basic events cannot occur simultaneously and the case in which two basic events occur at the same moment in a sequence will hence never happen.

Our approach is based on Monte Carlo simulation and hence on the generation of random sequences of occurrences. As we aim at comparing these random sequences of occurrences with the minimal cut sequences of the DFT, we hence have to provide minimal cut sequences only and to convert the minimal cut sets of the DFT into minimal cut sequences:

- the minimal cut set $CS$ is equivalent to the minimal cut sequence $[CS]$;
- the minimal cut set $SS$ is equivalent to the minimal cut sequence $[SS]$;
- the minimal cut set $MOTOR \cdot MOTORC$ is equivalent to the two minimal cut sequences $[MOTOR, MOTORC]$ and $[MOTORC, MOTOR]$, as both $MOTOR$ and $MOTORC$ need to occur and cannot occur simultaneously as they are statistically independent.

We can hence conclude that the DFT in Figure 3 has eight minimal cut sequences:

- two minimal cut sequences of length 1: $[CS]$ and $[SS]$;
- four minimal cut sequences of length 2: $[B_d, P]$, $[P, B_a]$, $[MOTOR, MOTORC]$ and $[MOTORC, MOTOR]$;
- two minimal cut sequences of length 3: $[P2, P1, BP]$ and $[P1, BP, P2]$.

## 4.3 Monte Carlo simulation of the DFT example

In order to compare the results obtained by Monte Carlo simulation with the results obtained by Merle,[17] we retain exponential time-to-failure distributions for basic events with the failure rates given in Table I and with a dormancy of 0.5 for the spare event $B$. It can be noted that there is no methodological or computational obstacle that may prevent us from applying our approach to non-exponential distributions.

**Table I.** Failure rates of the basic events of the DFT of the HCAS, from the work of Boudali and Dugan[13]

| Basic component | Failure rate ($10^{-6}$) |
|---|---|
| $CS$ | 1 |
| $SS$ | 2 |
| $P$, $B$ | 4 |
| $P1$, $P2$, $BP$ | 5 |
| $MOTOR$ | 5 |
| $MOTORC$ | 1 |

$CS$: crossbar switch; $SS$: system supervisor; $P1$: $PUMP\_1$; $P2$: $PUMP\_2$; $BP$: $Backup\_PUMP$; $MOTORC$: motor cable.

We need to determine how many simulations are needed to obtain an accurate result for the failure probability of the system. According to Equation (4),

$$N \geqslant \frac{1}{2\varepsilon^2} \ln\left(\frac{2}{\delta}\right).$$

Let us target a result with an accuracy of $\varepsilon = 1\%$ and with $\delta = 5\%$ (i.e. we want to have a probability of 95% of obtaining a result with the accurary $\varepsilon$). We hence have

$$N \geqslant \frac{1}{2 \times 0.01^2} \ln\left(\frac{2}{0.05}\right) \approx 18,444.$$

Twenty thousand simulations will thus be sufficient to obtain a result with the expected accuracy. The result obtained is a failure probability of 36.19% at a mission time $T = 100,000$ h, which is in accordance with the failure probability of 36.35% obtained by Merle[17] at the same mission time, with an error of only 0.44%.

It can be noted that the results obtained with Monte Carlo simulation are determined within a few seconds or minutes, depending on the number of simulations. If we retain a probability of 95% of obtaining an accurate probability,

- a 1% accurate result is calculated within a few seconds with $20,000$ simulations;
- a 0.1% accurate result is calculated within a few minutes with $2,000,000$ simulations.

It can be noted that these computation times are similar with the ones obtained from the work of Boudali and Dugan[13] with Bayesian networks, for which the accuracy of the result – as well as the computation time – depends on the time granularity. However, the types of time-to-failure distributions that can be taken into account in Bayesian networks are limited by inference algorithms, like we said in Section 2, whereas Monte Carlo simulation can accommodate any such distribution.

## 4.4 Global analysis of the DFT example

As we consider exponential time-to-failure distributions, each basic event has one parameter, that is, its failure rate $\lambda$. The DFT in Figure 3 has nine basic events, so the system hence has nine parameters. As the failure rates of spare events in their dormant and active modes are linked by the dormancy $\alpha$ of the spare event, we just consider one failure rate (and hence one parameter) per spare event too.

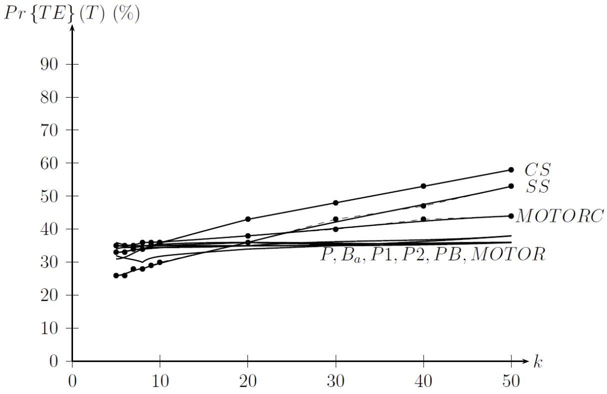

**Figure 4.** Graphical representation of the failure probabilities of Table II.

We can check which parameters are the most important with respect to the system by making each one of them vary whilst the eight other ones are kept unchanged. The failure probabilities obtained for each case are displayed in Table II, and they are represented graphically in Figure 4. We consider that each component $X$ from the first column fails with a variable failure rate $\lambda_X(k) = \dfrac{k \cdot \lambda_X}{10}$, where $\lambda_X$ is the initial failure rate from Table I and where the value of $k$ is given in the first row of Table II and can take values in $\{5, 6, 7, 8, 9, 10, 20, 30, 40, 50\}$, whilst the failure rates of the eight other basic events are kept unchanged. It can be noted that $\lambda_X(10) = \lambda_X$, that is, the new failure rate when $k = 10$ equals to the original one. These results were obtained by performing $10,000$ simulations, which gives them a probability of 75% of having an accuracy of 1%. On the other hand, the probability that every value of Table II has an accuracy of 2% is greater than 94%. The failure probabilities that are particularly high are indicated by bold italic letters, whereas the failure probabilities that correspond to the original values of the parameters of the system (i.e. the ones in Table I) are in bold letters.

On the one hand, it can be noted that the failure mechanisms of the pumps, motors and CPU units are working well, because the variation of the failure rate of their components has a minor impact on the failure

**Table II.** Estimated failure probabilities in the exponential model

| | 5 | 6 | 7 | 8 | 9 | **10** | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|---|---|---|---|---|
| $CS$ | 33 | 33 | 34 | 35 | 36 | **36** | 42 | 47 | 53 | ***57*** |
| $SS$ | 29 | 30 | 32 | 33 | 35 | **36** | 48 | ***57*** | ***65*** | ***71*** |
| $P$ | 34 | 35 | 34 | 35 | 36 | **36** | 40 | 44 | 46 | 47 |
| $B$ | 34 | 34 | 35 | 35 | 35 | **36** | 41 | 45 | 46 | 49 |
| $P1$ | 36 | 36 | 36 | 36 | 36 | **36** | 37 | 37 | 37 | 38 |
| $P2$ | 36 | 36 | 36 | 35 | 36 | **36** | 37 | 38 | 38 | 38 |
| $BP$ | 35 | 37 | 36 | 36 | 36 | **36** | 36 | 36 | 37 | 37 |
| $MOTOR$ | 35 | 36 | 36 | 35 | 36 | **36** | 38 | 39 | 39 | 40 |
| $MOTORC$ | 35 | 36 | 35 | 36 | 36 | **36** | 38 | 39 | 42 | 43 |

$CS$: crossbar switch; $SS$: system supervisor; $P1$: $PUMP\_1$; $P2$: $PUMP\_2$;
$BP$: $Backup\_PUMP$; $MOTORC$: motor cable.

probability of the system. On the other hand, it can be noted that the variation of the failure rates of the components of the trigger unit has a strong impact on the failure probability of the system with, for instance, a particularly low failure probability when $k = 5$ and a particularly high failure probability when $k = 50$ for $CS$ and $SS$. Consequently, if one component had to be improved, one had better choose it in the trigger unit.

One may also wonder whether it may be fruitful to improve a whole unit. We can study the impact of the improvement of all the components of a unit on the failure probability of the system. Let us consider that the improved failure rate of the components of the unit considered is decreased to $\lambda = 5 \cdot 10^{-7}$:

- if we decide to improve the motors unit, the failure probability of the system is decreased by 6%;
- if we decide to improve the pumps unit, the failure probability of the system is decreased by 3%;
- if we decide to improve the CPU unit, the failure probability of the system is decreased by 17%;
- if we decide to improve the trigger unit, the failure probability of the system is decreased by 39%.

All these results were obtained after performing $20,000$ simulations, and we hence have a probability of 95% that they have an accuracy of 1%. This analysis hence confirms that one had better focus on the trigger unit if a unit had to be improved.

We also run simulations to determine whether the choice of a Weibull distribution instead of an exponential distribution for the pumps unit would have an impact. The results are summarized in Table III with the specific choice of $k = 0.5$ and in Table IV for $k = 2$. We stress out the fact that in these simulations, we modified the shape parameter (from $k = 1$ to $k = 0.5$ or $k = 2$) and the scale parameter (from $\lambda$ to $\dfrac{\lambda}{2}$ or $\lambda \times \dfrac{2}{\sqrt{\pi}}$) to keep the life expectancy invariant.

As it was quite predictable (because of the local analysis detailed in the following section), these changes have very few impacts on the estimated failure probabilities. For illustration purpose only, we also modelled the case where the CPU unit has Weibull distribution, although it might not be a good description of the reality. The results are provided in Table V, and, as suspected, they show significant differences in the estimated failure probabilities.

## 4.5 Local analysis of the DFT example

The sensitivity indexes of all the components can be determined by means of two samples of size $N = 20,000$. Indeed, two samples are necessary to be able to calculate conditional expectations. The first order sensitivity indexes are given in Table VI.

**Table III.** Estimated failure probabilities with pumps unit following Weibull distribution with shape parameter k = 0.5

| | 5 | 6 | 7 | 8 | 9 | **10** | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|---|---|---|---|---|
| $CS$ | 33 | 33 | 34 | 36 | 36 | **36** | 43 | 48 | 53 | ***56*** |
| $SS$ | 29 | 32 | 32 | 34 | 35 | **36** | 49 | ***56*** | ***65*** | ***71*** |
| $P$ | 33 | 34 | 35 | 35 | 35 | **36** | 41 | 43 | 46 | 47 |
| $B$ | 33 | 33 | 34 | 35 | 36 | **36** | 41 | 42 | 47 | 48 |
| $P1$ | 36 | 36 | 36 | 36 | 37 | **37** | 37 | 36 | 37 | 37 |
| $P2$ | 36 | 37 | 36 | 36 | 36 | **36** | 37 | 37 | 37 | 37 |
| $BP$ | 37 | 36 | 37 | 36 | 36 | **37** | 37 | 37 | 38 | 38 |
| $MOTOR$ | 35 | 36 | 35 | 36 | 36 | **36** | 38 | 38 | 40 | 39 |
| $MOTORC$ | 35 | 35 | 37 | 35 | 36 | **36** | 38 | 40 | 43 | 44 |

$CS$: crossbar switch; $SS$: system supervisor; $P1$: $PUMP\_1$; $P2$: $PUMP\_2$;
$BP$: $Backup\_PUMP$; $MOTORC$: motor cable.

**Table IV.** Estimated failure probabilities with pumps unit following Weibull distribution with shape parameter k = 2

| | 5 | 6 | 7 | 8 | 9 | **10** | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|---|---|---|---|---|
| $CS$ | 32 | 32 | 33 | 34 | 34 | **36** | 41 | 47 | 52 | ***56*** |
| $SS$ | 29 | 29 | 32 | 33 | 35 | **35** | 47 | ***56*** | ***64*** | ***70*** |
| $P$ | 33 | 33 | 33 | 35 | 35 | **35** | 39 | 43 | 46 | 46 |
| $B$ | 32 | 33 | 34 | 35 | 35 | **35** | 40 | 44 | 46 | 48 |
| $P1$ | 35 | 34 | 35 | 35 | 35 | **35** | 36 | 37 | 36 | 37 |
| $P2$ | 34 | 34 | 35 | 36 | 35 | **35** | 36 | 36 | 37 | 36 |
| $BP$ | 35 | 35 | 35 | 35 | 35 | **35** | 36 | 36 | 36 | 36 |
| $MOTOR$ | 34 | 33 | 35 | 35 | 35 | **35** | 37 | 37 | 39 | 39 |
| $MOTORC$ | 33 | 34 | 35 | 35 | 35 | **36** | 37 | 40 | 41 | 43 |

$CS$: crossbar switch; $SS$: system supervisor; $P1$: $PUMP\_1$; $P2$: $PUMP\_2$;
$BP$: $Backup\_PUMP$; $MOTORC$: motor cable.

**Table V.** Estimated failure probabilities with pumps and CPU units following Weibull distribution with shape parameter k = 2

| | 5 | 6 | 7 | 8 | 9 | **10** | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|---|---|---|---|---|
| CS | 17 | 17 | 18 | 17 | 18 | **18** | 21 | 26 | ***32*** | ***39*** |
| SS | 15 | 15 | 16 | 17 | 18 | **18** | ***31*** | ***45*** | ***63*** | ***75*** |
| P | 15 | 15 | 16 | 17 | 17 | **18** | 23 | ***27*** | ***30*** | ***32*** |
| B | 14 | 16 | 16 | 17 | 17 | **18** | 24 | ***29*** | ***31*** | ***34*** |
| P1 | 18 | 18 | 18 | 17 | 18 | **18** | 19 | 21 | 20 | 20 |
| P2 | 18 | 18 | 18 | 17 | 18 | **18** | 19 | 19 | 20 | 20 |
| BP | 18 | 18 | 18 | 18 | 18 | **18** | 19 | 19 | 19 | 19 |
| MOTOR | 16 | 18 | 17 | 18 | 18 | **18** | 20 | 21 | 22 | 23 |
| MOTORC | 17 | 17 | 18 | 18 | 18 | **18** | 21 | 24 | 26 | ***29*** |

$CS$: crossbar switch; $SS$: system supervisor; $P1$: $PUMP\_1$; $P2$: $PUMP\_2$;
$BP$: $Backup\_PUMP$; $MOTORC$: motor cable.

It can be noted that both $CS$ and $SS$ have a major impact on the variance of the top event and hence on the failure probability of the system. $SS$ has the highest importance because of its twice lower life expectancy with respect to $CS$. This analysis confirms that the trigger unit should be the unit on which to focus.

Higher order sensitivity indexes are not represented for two reasons: basic computations show that there are more than 500 of them, and they are close to zero. This explains the missing 35% (split between these hundreds of terms) in Table VI.

**Table VI.** Sensitivity indexes of the components

| $CS$ | $SS$ | $P$ | $B$ | $P1$ | $P2$ | $BP$ | $MOTOR$ | $MOTORC$ |
|---|---|---|---|---|---|---|---|---|
| 18% | 39% | 5% | $\approx$ 0% | $\approx$ 0% | $\approx$ 0% | $\approx$ 0% | $\approx$ 0% | 3% |

$CS$: crossbar switch; $SS$: system supervisor; $P1$: $PUMP\_1$; $P2$: $PUMP\_2$; $BP$: $Backup\_PUMP$; $MOTORC$: motor cable.

## 5 Conclusion

In this paper, we presented an approach allowing to perform the quantitative analysis of DFTs. This approach is based on both Monte Carlo simulation, which allows to generate random failure sequences for any time-to-failure distribution of basic events, and the structure function of DFTs, which allows to determine the minimal cut sequences of DFTs. On the one hand, we have showed that our approach allows to obtain accurate quantitative results within a short calculation time. Any time-to-failure distribution can be accommodated as, from any such distribution, a random failure date can be generated for each basic event and hence random failure sequences. On the other hand, our approach also allows to obtain results that are quite uncommun – and yet useful – in quantitative analysis, such as the importance measures of the parameters of the distribution functions of basic events or the sensitivity of the system with respect to these parameters.

Ongoing work is now addressed to the determination of a scoring criterion allowing to classify configurations of parameters according to their criticity in terms of risk with respect to failure events.

## References

1. Henley EJ, Kumamoto H. Reliability Engineering and Risk Assessment. Prentice Hall, Englewood Cliffs, 1981.
2. Leveson NG. Safeware: System Safety and Computers. Addison-Wesley, 1995.
3. Stamatelatos M, Vesely W. Fault Tree Handbook with Aerospace Applications, vol. **1.1**. NASA Office of Safety and Mission Assurance, Washington DC, 2002; 205.
4. Fussell JB, Aber EF, Rahl RG. On the quantitative analysis of Priority-AND failure logic. *IEEE Transactions on Reliability* 1976; **R-25**(5):324-326, doi:10.1109/TR.1976.5220025.
5. Dugan JB, Bavuso SJ, Boyd MA. Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transactions on Reliability* 1992; **41**(3):363-377, doi:10.1109/24.159800.
6. Dugan JB, Sullivan KJ, Coppit D. Developing a low-cost high-quality software tool for dynamic fault-tree analysis. *IEEE Transactions on Reliability* 2000; **49**(1):49-59, doi:10.1109/24.855536.
7. Tang Z, Dugan JB. Minimal cut set/sequence generation for Dynamic Fault Trees. *Proc. of the Annual Reliability and Maintainability Symp.*, Los Angeles, CA, USA, 2004; 207-213.
8. Walker M, Papadopoulos Y. Qualitative temporal analysis: towards a full implementation of the Fault Tree Handbook. *Control Engineering Practice* 2009; **17**(10):1115-1125, doi:10.1016/j.conengprac.2008.10.003.

9. Boudali H, Crouzen P, Stoelinga M. A compositional semantics for Dynamic Fault Tree in terms of interactive Markov chains. *Proc. of the Int. Symp. on Automated Technology for Verification and Analysis (ATVA'07)*, Tokyo, Japan, 2007; 441-456.
10. Boudali H, Crouzen P, Stoelinga M. Dynamic Fault Tree analysis through input/output interactive Markov chains. *Proc. of the Int. Conf. on Dependable Systems and Networks DSN 2007*, Edinburgh, UK, 2007; 25-38.
11. Bobbio A, Codetta Raiteri D. Parametric fault trees with dynamic gates and repair boxes. *Proc. of the Annual Reliability and Maintainability Symp.*, Los Angeles, CA, USA, 2004; 459-465.
12. Codetta Raiteri D. The conversion of Dynamic Fault Trees to Stochastic Petri Nets, as a case of graph transformation. *Electronic Notes on Theoretical Computer Science* 2005; **127**(2):45-60, doi:10.1016/j.entcs.2005.02.005.
13. Boudali H, Dugan JB. A discrete-time Bayesian network reliability modeling and analysis framework. *Reliability Engineering and System Safety* 2005; **87**(3):337-349, doi:10.1016/j.ress.2004.06.004.
14. Merle G, Roussel JM, Lesage JJ. Algebraic determination of the structure function of Dynamic Fault Trees. *Reliability Engineering and System Safety* 2011; **96**(2):267-277, doi:10.1016/j.ress.2010.10.001.
15. Merle G, Roussel JM, Lesage JJ. Quantitative analysis of Dynamic Fault Trees based on the structure function. *Quality Reliability Engineering International* 2014; **30**(1):143-156.
16. Durga Rao K, Gopika V, Sanyasi Rao VVS, Kushwaha HS, Verma AK, Srividya A. Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment. *Reliability Engineering and System Safety* 2009; **94**(4):872-883, doi:10.1016/j.ress.2008.09.007.
17. Merle G. Algebraic modelling of Dynamic Fault Trees, contribution to qualitative and quantitative analysis. PhD thesis, École Normale Supérieure de Cachan, 2010.
18. Coppit D, Sullivan KJ, Dugan JB. Formal semantics of models for computational engineering: a case study on Dynamic Fault Trees. *Proc. of the 11th Int. Symp. on Software Reliability Engineering*, San Jose, CA, USA, 2000; 270-282.
19. Montani S, Portinale L, Bobbio A, Varesio M, Codetta-Raiteri D. DBNet, a tool to convert Dynamic Fault Trees into dynamic Bayesian networks. Università del Piemonte Orientale, Technical Report TR-INF-2005-08-02-UNIPMN, 2005.
20. Vittorini V, Franceschinis G, Gribaudo M, Iacono M, Mazzocca N. Drawnet: model objects to support performance analysis and simulation of systems. *Proc. of the 12th Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation*, Springer Verlag - LNCS, **2324**, 2002; 233-238.
21. Lauritzen SL, Jensen F. Stable local computation with conditional Gaussian distributions. *Statistics and computing* 2001; **11**(2):191-203, doi:10.1023/A:1008935617754.
22. Moral S, Rumí R, Salmerón A. Mixtures of truncated exponentials in hybrid Bayesian networks. *Proc. of the 6th European Conf. on Symbolic and Quantitative Approaches to Reasoning With Uncertainty*, Toulouse, France, 2001; 145-167.
23. Banks J, Carson J. Discrete-event System Simulation. Prentice Hall, Upper Saddle River, NJ, USA, 1984.
24. Marsaguerra M, Zio E, Devooght J, Labeau PE. A concept paper on dynamic reliability via Monte Carlo simulation. *Mathematics and Computers in simulation* 1998; **47**(2-5):371-382, doi:10.1016/S0378-4754(98)00112-8.
25. Marsaguerra M, Zio E. Monte Carlo estimation of the differential importance measure: application to the protection system of a nuclear reactor. *Reliability Engineering and System Safety* 2004; **86**(1):11-24, doi:10.1016/j.ress.2003.12.011.
26. Zio E, Marella M, Podofillini L. A Monte Carlo simulation approach to the availability assessment of multi-state systems with operational dependencies. *Reliability Engineering and System Safety* 2007; **92**(7):871-882, doi:10.1016/j.ress.2006.04.024.
27. Zio E, Podofillini L, Levitin G. Estimation of the importance measures of multi-state elements by Monte Carlo simulation. *Reliability Engineering and System Safety* 2004; **86**(3):191-204, doi:10.1016/j.ress.2004.01.009.
28. Miao Q, Zhang X, Ling D, Chen Z, Huang HZ. Reliability assessment of ring-standby structure based on Monte Carlo simulation. *Proc. of the 8th Int. Conf. on Reliability, Maintainability and Safety (ICRMS 2009)*, Chengdu, China, 2009; 1115-1118.

29. Manno G, Chiacchio F, Compagno L, D'Urso D, Trapani N. MatCarloRe: an integrated FT and Monte Carlo Simulink tool for the reliability assessment of Dynamic Fault Tree. *Expert Systems with Applications* 2012; **39**(12):10334-10342, doi:10.1016/j.eswa.2011.12.020.
30. Merle G, Roussel JM, Lesage JJ, Bobbio A. Probabilistic algebraic analysis of fault trees with priority dynamic gates and repeated events. *IEEE Transactions on Reliability* 2010; **59**(1):250-261, doi:10.1109/TR.2009.2035793.
31. Merle G, Roussel JM, Lesage JJ. Improving the efficiency of Dynamic Fault Tree analysis by considering gates FDEP as static. *Proc. of the ESREL'2010 Conf.*, Rhodes, Greece, 2010; 845-851.
32. Devroye L. Non-Uniform Random Variable Generation. Springer-Verlag, New-York, 1986.
33. Hoeffding W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* 1963; **58**(301):13-30.
34. Cukier RI, Fortuin CM, Schuler KE, Petschek AG, Schaibly JH. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I. Theory. *The Journal of Chemical Physics* 1973; **59**:3873-3878.
35. Sobol IM. Sensitivity analysis for non-linear mathematical models. *Mathematical Modeling and Computational Experiment* 1993; **1**:407-414.
36. Vemuri KK, Dugan JB, Sullivan KJ. Automatic synthesis of fault trees for computer-based systems. *IEEE Transactions on Reliability* 1999; **48**(4):394-402, doi:10.1109/24.814522.
37. Ren Y, Dugan JB. Design of reliable systems using static and Dynamic Fault Trees. *IEEE Transactions on Reliability* 1998; **47**(3):234-244, doi:10.1109/24.740491.
38. Chastaing G, Gamboa F, Prieur C. Generalized Hoeffding-Sobol decomposition for dependent variables - application to sensitivity analysis. *Electronic Journal of Statistics* 2012; **6**:2420-2448, doi:10.1214/12-EJS749.

*Authors' biographies*

**Guillaume Merle** received his MSc degree in Systems Engineering from Ecole Normale Supérieure de Cachan (France) in 2007 and his PhD degree also from Ecole Normale Supérieure de Cachan in 2010. Then he did post-docs at Tsinghua University, at the Chinese Academy of Sciences, and at Ecole Normale Supérieure de Cachan. He is currently an Associate Professor of Engineering Science at the Sino-French Engineering School of Beihang University (Beijing, China). His research interests focus on analysis and diagnosis of discrete event systems and dynamic reliability.

**Jean-Marc Roussel** has been an Associate Professor of automatic control at Ecole Normale Supérieure de Cachan since 1995. He obtained his PhD in Automatic Control of Discrete Events Systems. His research interests are formal analysis of control systems, reliability engineering, and system safety.

**Jean-Jacques Lesage** received his MSc degree from Univ. Paris 6 and his PhD degree from Ecole Centrale de Paris. He is currently a Full Professor of Automatic Control at Ecole Normale Supérieure de Cachan. His research interests focus on formal methods and models for synthesis, analysis, and diagnosis of discrete event systems, with applications to manufacturing systems, network automated systems, energy production, and ambient assisted living.

**Vianney Perchet** is Assistant Professor in the Probability and Statistics Department of Univ. Paris 7. His research interests range from game theory to statistics, along with machine learning. He obtained his BSc and MSc degrees in Mathematics and Economics at Ecole Normale Supérieure de Paris and graduated from the National School of Statistics and Economical Administration. He then received his PhD degree in Game Theory from Univ. Paris 6 and then did a post-doc at Ecole Normale Supérieure de Cachan.

**Nicolas Vayatis** received a BE degree from Ecole Centrale Paris, his MSc degree from Ecole Polytechnique in 1995, and his PhD degree also from Ecole Polytechnique in 2000. He is currently a Full Professor at the Department of Mathematics of Ecole Normale Supérieure de Cachan and head of the Centre de Mathématiques et de Leurs Applications. His research interests focus on the theory and applications of machine learning and the use of mathematical modeling techniques to contribute to data-driven decision-making systems for the industrial and medical sectors.