Reliability Analysis of Sensorized Stamping Presses by Generalized Fault Trees

Eugénio M. Rocha
Department of Mathematics and Center of Research and Development in Mathematics and Applications (CIDMA), Campus de Santiago, Aveiro, 3810-193, Portugal
eugenio@ua.pt

Pedro Nunes and José Santos
Department of Mechanical Engineering and Center for Mechanical Technology and Automation (TEMA), Campus de Santiago, Aveiro, 3810-193, Portugal
pnunes@ua.pt, jps@ua.pt

Abstract

Reliability analysis and probability methods are of extreme importance for understanding the behavior of critical systems. In this scope, dynamic fault trees (DFTs) are consolidated graphical models, previously applied at known entities, such as NASA and TESLA. However, the (classical) DFTs analysis has a known issue; the fact that it assumes that the distribution of basic events (BEs) follows the exponential/Weibull distribution, which is often a rough approximation of real data distributions. Moreover, building a DFT model for a real system requires specialized knowledge, to infer the root causes of failures. In this work, a new formalism for the analysis of so-called generic fault trees (GFTs) is proposed which extends DFT to data-driven scenarios, based on the notion of h-approximation of the associated distributions, where leaves may contain an arbitrary compact support distribution or an h-truncation of a distribution. The approach is validated against known solutions in the literature, showing great accuracy. An optimization process is employed to generate the best structure for training a GFT that fits a given data and tested in real use case scenario of a stamping press of Bosch ThermoTechnology. The obtained GFT demonstrates a good fit, considering 3 different metrics.

Keywords
Generalized fault trees, Dynamic fault trees, Data-driven, Reliability analysis, Industry 4.0.

1. Introduction and Background

Reliability analysis and probability methods are of utmost importance to ensure the correct working of systems in several fields, such as industrial, medical, automotive and energy sectors. Regarding the industrial sector, efforts have been made in the predictive maintenance field, to reduce costs and improve the availability of equipment (Zonta et al., 2020). Traditional techniques often exploit Machine Learning (ML), and classification techniques to detect early failures on the equipment, however, these methods strongly depend on data patterns, and may fail when hard, or unexcepted failures occur not previously found on the trained data. Stamping presses are subjected to this kind of mechanical failures, which represent one of the main reasons for equipment unavailability or unreliability. This work was developed in the follow-up of the work developed by Coelho et al. (2021), where authors developed an innovative approach merging time-series segmentation, anomaly detection and ranking twelve machine/deep learning algorithms. Although the obtained results for some types of failures are quite good, the approach is quite ineffective for other types of failures. Here, we address such issues with the development of a method on reliability analysis, through the definition of an extension of dynamic fault trees (DFT), the so-called generic fault trees (GFT), and apply it to the same stamping press at Bosch ThermoTechnology. Furthermore, the proposed method enables the automatic generation of the tree structure, that describes the root cause of the press' mechanical failures.

Fault tree analysis (FTA) is a well-established technique, that has been exploited by known entities, such as NASA (Stamatelatos et al., 2002), to model and quantify the risk of failure in large systems (Kabir, 2017). A fault tree (FT) is a directed acyclic graph, in the shape of a tree, that has two types of nodes, events and gates. There are two types of events, the basic events (BEs), that are in the tree's leaves and represent the failure of some components or parts within the main system (Ruijters & Stoelinga, 2015), and the top event, which represents the failure of the complete system being considered. The gates characterize how the combination of BEs event's fails affect the top event's fails.

A traditional fault tree (FT), commonly, comprise the gates AND, OR and Voting. An event resulting from an AND gate fails if all the events in this gate fail. On the other hand, an event resulting from an OR gate fails if at least one of the events in the gate fails. This approach is simple; however, it does not represent the dynamic behavior of a system's failure. For example, a failure of a part may depend on the sequence of failures of sub-parts, and spares are not modelled properly within a FT approach. To overcome these problems, dynamic gates, such as the priority AND (PAND), the sequence enforcing (SEQ), the spare (SPARE) and the functional dependency (FDEP), have been employed (Dugan et al., 1992) to perform the so-called DFT analysis.

The PAND gate reaches a failure if both the basic events fail in a pre-assigned order (usually, from the left to the right). In Figure 1, a failure occurs if A fails before B, but does not occur if B fails before A. Different from the PAND, the SEQ gate forces events to fail in a specific order. In the case of Figure 1, B can only fail after A fails, and C can only fail after A and B fail.

SPARE gates model components that may be replaced by one or more spares. This gate fails when the number of healthy components is inferior to an established number. From Figure 1, if A fails, it may be replaced by S_1 or S_2. The SPARE gate may be divided in three distinct types: a Hot SPARE (HSP), when the spare parts have the same failure probability in normal and dormant state; a Cold SPARE (CSP), where the spare part cannot fail in dormant state; and the Warm Spare (WSP), where the spare can fail in both states, but with different probabilities of failure. Usually the failure probability, in the dormancy state, is smoothed by a dormancy factor.

The FDEP gate represents the functional dependency of some events (A, B, C), to a trigger event (T). It helps to model a scenario, when the working condition of some components is dependent on one component. For example, in a system with a single power supply (T), its failure will cause the failure of all components that depend on it (A, B, C). These events are dependent on T, and can have their individual failure, which does not imply the failure of T.

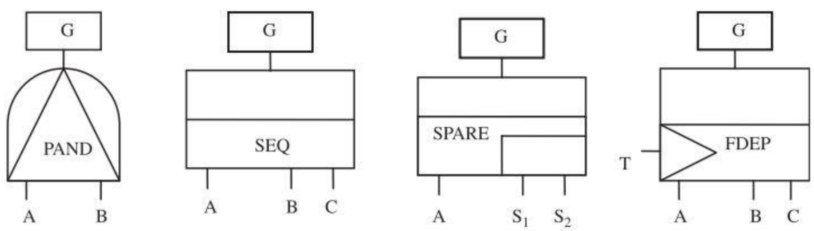


Figure 1: Dynamic gates.

2. Literature Review

There are several proposed methodologies to analyze a DFT, such as algebraic solutions, Markov models, Petri nets, Bayesian networks, Monte Carlo simulation, among others, (Aslansefat et al., 2020). Often, methods to solve a DFT require the mapping of the tree model, to a different one, that can be solved, analytically or via simulation. An example consists in converting the tree model into a continuous-time Markov chain (CTMC). Sihite & Kohda (2013) employed this approach to analyze the reliability of power transformer systems, while Volk & Junges (2018), proposed reduction techniques to exploit symmetries in the tree and reduce the Markovian's model state space. Aslansefat & Latif-Shabgahi (2020), employed Semi-Markov models, to generalize the BEs distribution to a Weibull distribution.

Other state space models are employed to solve a DFT, for example, Khakzad et al. (2013) employed discrete-time Bayesian networks, to model SPARE and SEQ gates. Kabir et al. (2018), explored and applied Petri nets and Bayesian networks to an aircraft fuel distribution system, while Jiang et al. (2021) proposed a modular approach based on binary decision diagrams. Besides having strong mathematical formalism behind these models, they follow the assumptions of exponential distribution for the failure's probability, which in general does not accurately represent real world problems. Moreover, these models suffer from the state explosion issue, i.e., the state space increases exponentially with the increasing number of gates. Some software tools and model checkers, such as Galileo (Sullivan et al., 2003), Coral (Hichem Boudali et al., 2007), DFTCalc (Arnold et al., 2013), or Storm (Dehnert et al., 2016), are popular to

© IEOM Society International

solve DFTs, because they are based on the CTMC model and try to mitigate the state space explosion, by applying techniques such as aggregation and modularization, to exploit symmetries in the tree, and reducing the state space.

Simulation methods have been employed to consider non-exponential distributions and to deal with the explosion of state space. Durga Rao et al. (2009), Chiacchio et al.(2020), and Xu et al. (2021) employed the Monte Carlo simulation to perform DFT analysis. On the other hand, Zhu et al., (2014) developed a stochastic approach, to solve the PAND gate, and Cheshmikhani & Zarandi (2015) proposed a stochastic method to solve DFTs, while, H. Boudali et al. (2009) developed a simulation tool and compared its results with other software tools, such as Galileo and Coral.

Algebraic formulas have been employed to calculate the probability of gates in FT analysis. However, regarding DFTs, expressions to deal with dynamic gates are more recent. Merle et al. (2010) proposed an algebraic framework for modeling priority dynamic gates, such as PAND and FDEP, and, in further works Merle et al. (2010 a), and Merle et al. (2011), proposed an algebraic formulation for the Spare gates. Meanwhile Elderhalli et al. (2019) used an algebraic approach together with a combination of a Lebesgue integral analysis and higher-order logic (HOL). Kabir et al. (2020), used a hybrid modular approach. They converted a DFT into a Petri net and then, divided it in subtrees, where the Monte Carlo simulation was employed to solve gates with non-exponential distributed events, the CTMC was used to solve gates whose BEs follow exponential distributions, and algebraic expressions were employed to solve static subtrees.

The solution for a DFT analysis may require complex analytical models, which often are not a good option to model real problems, since there are assumptions that events follow exponential or Weibull distribution. Simulation methods may overcome these issues, but they require large computational times to achieve accurate results. Algebraic approaches allow more efficiency in analyzing DFTs, however, they need a general method to deal with the real probability distributions of BEs. Model checker and hybrid models are still evolving in this area. Moreover, to build a DFT structure for a real case system is not an easy task, since it requires expert's knowledge, which may be expensive, or even impossible for all industrial systems. To the author's knowledge, there is no solution that generates a GFT automatically from sensor data. The novelty of this work is two-fold: we generalize the DFT analysis to the so-called GFT analysis, where BEs may follow an arbitrary compact support distribution. We also propose an optimization method to generate the best GFT structure for a given dataset, which is applied in a real dataset from a stamping press.

The remaining is organized as follows; Section 3 describes the proposed approach to solve numerically a GFT. Section 4 presents the optimization process, to fit the structure of a GFT to given data, while Section 5 validates the proposed approach against Storm's known solutions in the literature. Finally, Section 6 discusses a real use case, a stamping press at Bosch TermoTechnology, and Section 7 presents the conclusions and final remarks.

3. Generalized Fault Trees (GFT)

A GFT approach solves the main problem of (classical) DFT, that is assuming that the BEs follow the exponential distribution. The algebraic formalism and the numerical method based on a h-approximation method, are described in this section, namely, Subsection 3.1 exposes some definitions and concepts that are essential to understand the proposed approach, while Subsection 3.2 presents the formalism for the generalization of the gates.

3.1 Definitions and Concepts

As mentioned, a fault tree is a graphical formalism employed for reliability analysis, thus, mathematically, it has the following structure, $DFT= (V, E, R, L, f_L, O, f_O)$, where:

- V is the set of nodes/vertices;
- E is the set of arcs, i.e., $E = \{(v_i, v_j)\}, \forall\, v_i, v_j \in V$;
- R is the top node/root node;
- $L \subset V$ is the set of leaves;
- $f_L: L \to R^k$ is a function: $f_L(v) = D_v, v \in L$;
- O is the set of gates/operators, where, for each $\mathrm{op} \in \mathrm{O}, \mathrm{op}(\mathrm{D}_1, \dots, \mathrm{D}_\mathrm{k}) \in \mathfrak{D}$, where $\mathfrak{D}$ is the set containing the values for the probability mass function (PMF) of each node;
- $\mathrm{f_O}: \mathrm{V} \setminus \{\mathrm{L}\} \to \mathrm{O}$, is a function that associates an operator to each intermediate node.

© IEOM Society International

In the GFT approach, is achieved, the gate/operator receives the arbitrary PMF of BEs as input, and outputs a new distribution. In order to generalize the operators/gates for any distribution, it is necessary to establish some assumptions, such as:

1. All the BEs have discrete distributions equally spaced with Probability Mass Function (PMF), denoted by $\widehat{f_x}$, with step *h*, and with the same maximum support, i.e., the following expressions are verified:
    - $\mathrm{a} = \mathrm{x}_0 < \mathrm{x}_1 < \cdots < \mathrm{x}_\mathrm{n} = \mathrm{b}$;
    - $\exists \mathrm{h} > 0 \colon \mathrm{x}_{\mathrm{i}+1} = \mathrm{x}_\mathrm{i} + \mathrm{h}$.
2. The probability distributions of BEs are known.
3. The results may have an associated error that depends on the discretization step, *h*, but converge to zero, when $\mathrm{h} \to 0$.

Thus, consider a BE, *X*, and $\widehat{f_x} = [\overline{x}_0, \dots, \overline{x}_n]^T, \tilde{x} = \widehat{f_x}(x_i)$ its PMF. Assuming that $\mathrm{f_x}$ is the continuous version of the PMF, the cumulative density function (CDF) is given by equation 1,

$$P[X \leq X_K] = \int_{-\infty}^{X_K} f_x(\xi)\, d\xi\,, \;\; \forall \;\; x_K \in \;]a, b[\,, \qquad \textbf{(1)}$$

where $\mathrm{P[X \leq X_K]}$ is the CDF until the instant of time X_K, or, in other words, the probability that the BE has occurred at before, or at time X_K. Then, by solving equation 1 numerically by the composite trapezoidal rule, an arbitrary CDF can be approximated by equation 2,

$$P[X \leq X_K] = h \sum_{i=0}^{K} w_i \widehat{f_x}(x_i) \; + \; h^3 \left(-\frac{K}{12} f''(\theta)\right) + \; G_x\,, \quad \textbf{(2)}$$

where *h* is the discretization step, $\mathrm{G_x}$ is the CDF value for the lower bound of the considered support (*a*). For a small value of *a*, one can consider $\mathrm{G_x} \simeq 0$, $\mathrm{w_i}$ is given by equation 3, and $h^3 \left(-\frac{K}{12} f''(\theta)\right)$ is the associated approximation error,

$$w_i = \begin{cases} \frac{1}{2}, if i \in 0, n \\ 1, if \;\; 0 < i < n \end{cases} \;\; . \qquad \textbf{(3)}$$

3.2 GFT Numerical Algorithm

The generalization of a classical DFT to a GFT requires algebraic expressions and calculations, in order to use an arbitrary distribution in the tree's leaves. The algebraic expressions, and numerical method for each gate, are presented throughout the next paragraphs.

OR operator: As mentioned, the output of a gate is a new PMF, thus, let us consider *X* and *Y*, the time to failure or time between occurrences, of two BEs, and *Z*, the time to failure of the gate. Thus, one can represent $\mathrm{Z} = \min\{\mathrm{X}, \mathrm{Y}\}$, i.e., the top event occurs when one of the inputs occurs. This is traduced by equation 4, and, assuming that both events are independent, equation 5 is verified.

$$Z = min\{X, Y\} \Rightarrow P(Z > t) = P(min\{X, Y\} > t), \qquad \textbf{(4)}$$

$$P(Z > t) = P(X > t)P(Y > t). \qquad \textbf{(5)}$$

The CDF of a gate's output is represented as $\mathrm{P(Z \leq t)}$. For the case of the OR operator, it is obtained from equation 5. The output probability density function (PDF) is obtained by the derivative of the CDF. Thus, assuming a discrete distribution with step *h*, the PMF of an arbitrary CDF can be obtained approximately by equation 7, briefly because the PDF is the derivative of the CDF for a continuous function and the derivative can be approximated by a first order difference with step h.

$$P(Z \leq t) = P(X \leq t) + P(Y \leq t) - P(X \leq t)P(Y \leq t), \;\; \textbf{(6)}$$

$$\hat{f}(z_k) = \frac{P(Z \leq x_k) - P(Z \leq x_{k-1})}{h}\,. \qquad \textbf{(7)}$$

AND operator: The AND operator can be formalized analogously to the OR operator, such as, $\mathrm{Z} = \max\{\mathrm{X}, \mathrm{Y}\}$, i.e., the top event only occurs when both input events occur, which is expressed by equation 8. Assuming the independence of events, equation 9 is obtained. Then $\mathrm{P(Z \leq t)}$ can be calculated as shown in equation 10 and the respective PMF can be obtained by equation 7.

$$Z = max\{X, Y\} \Rightarrow P(Z > t) = P(max\{X, Y\} > t) \qquad \textbf{(8)}$$

$$P(Z > t) = P(X > t) + P(Y > t) - P(X > t)P(Y > t) \quad \textbf{(9)}$$

$$P(Z \leq t) = P(X \leq t)P(Y \leq t). \quad \textbf{(10)}$$

PAND operator: The PAND operator requires that both BEs fail, in a specific order (from left to the right). Let *X* and *Y* be the time between events of the left and right BEs respectively. According to Merle et al. (2010), and Elderhalli et al. (2019), the time between occurrences of Z (i.e., output of PAND operator) is given by equation 11 and $\mathrm{P}(\mathrm{Z} \leq \mathrm{t})$ is given by equation 12, whose integral can be calculated numerically by equation 2.

$$Z = \begin{cases} Y, if\ X \leq Y \\ +\infty, if\ X > Y \end{cases} \quad \textbf{(11)}$$

$$P(Z \leq t) = \int_0^t P(X \leq \xi)\, f_Y(\xi)\ d\xi. \quad \textbf{(12)}$$

HSP operator: The HSP can be formalized in the same way as the AND operator, since the spare part has the same probability of failure, in active and dormant state, the operators are equivalent.

CSP operator: The CSP operator, does not allow spare parts to fail in dormant state, then, let *Y* be the time to failure of the principal part and *X* the time to failure of the spare, according to (Merle, Roussel, & Lesage, 2010), and (Elderhalli et al., 2019), the time to failure *Z* of CSP operator is given by equation 13, and $P(Z \leq t)$ is given as shown in equation 14, where $\mathrm{f}_{(X_a|Y\,=\,\xi)}$ is the conditional density function for the active state of the spare part X_a, given that the main part failed at time ξ. In this case, the PDF of the spare part is dependent on the failure time of the main one.

$$Z = \begin{cases} X, if\ Y < X \\ +\infty, if\ Y \geq X \end{cases} \quad \textbf{(13)}$$

$$P(Z \leq t) = \int_0^t \left(\int_\xi^t \mathrm{f}_{(X_a|Y\,=\,\xi)}(v)\ dv \right) f_Y(\xi)\ d\xi. \quad \textbf{(14)}$$

WSP operator: The failure of a WSP gate may occur in active or dormant state, but in the dormant state, the failure probability is softened by a dormancy factor (Elderhalli et al., 2019), thus, for the spare part, there are two distributions, one for the active state, X_a and the other for the dormant state X_d. In real case scenarios, the definition of dormancy factor is not such relevant, as in theoretical problems, because in more realistic scenarios, the failure distribution for the dormant state can be obtained, thus does not need to be estimated using this factor.

FDEP operator: According to (Ejlali & Ghassem Miremadi, 2004), the time to failure of a FDEP gate is equal to the minimum, between the time to failure of the trigger event and time to failure of the dependent event, which corresponds to the OR operator, thus, one considers the FDEP equivalent to the OR operator.

4. Training and Prediction with a GFT

In real case scenarios, the structure of a GFT is often unknown, or only a part of it is known. However, the distribution of BEs and top event are known through the histograms obtained from the historical data. In this section, a method is discussed to find the most suitable tree's structure to describe the event's distribution.

4.1 Training

To retrieve the correct fault tree structure, first, the distribution of the top event is computed based on the histograms of data, and then, a full search is performed, to generate all the possible trees, from the list of gates and BEs. Algorithm 1 represents the process, where *T* is the set of all trees' structures, that are possible with the set of operators, *O*, and input BEs. The expression $\mathrm{RMSE}\big(\mathfrak{D}(\mathrm{R_t}), \mathfrak{D}(\mathrm{R})\big)$ represents the Root Mean Square Error (RMSE), between the known top event's distribution $\mathfrak{D}(\mathrm{R})$, and the distribution of a tree $\mathfrak{D}(\mathrm{R_t})$ from the set *T*. This metric is calculated according to equation 15, where *N* is the number of discretization points.

When the GFT structure is not known, it is not guaranteed that all available BEs have influence in the distribution of the root event, thus, the number of input BEs to be used, is defined at the beginning of search, and the best tree for all the combinations of events is generated. This enables not only to find the closest structure for a given problem, but also to obtain information related to the BEs that most influence the root event.

© IEOM Society International

Algorithm 1 Training with a GFT

INPUT: distribution of basic events $\mathfrak{D}(v)$, $\forall v \in L, G$, the real distribution of the top event, $\mathfrak{D}(R)$, and the set of operators, *O*
OUTPUT: GFT structure, *GFT*

```
Initialize distance d, d_0 = ∞
Compute top event's distribution, D(R_t)
Compute the set of all trees, T
for ∀ t ∈ T do
    Compute d_t = RMSE(D(R_t), D(R))
    if d_t < d_0 then
        d_0 = d_t
        GFT = t
    end if
end for
```

$$RMSE_{X,Y} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(x_j - y_j\right)^2},\ j \in \{1,2,\dots,N\}. \qquad \textbf{(15)}$$

4.2 Prediction

When knowing the GFT structure, it allows to perform a careful risk analysis and reliability, that may be qualitative or quantitative. The qualitative analysis is one of the major advantages that is achieved by the graphical representation of the GFT. Having the tree's structure, the interaction between components and the failure's cause are easily interpreted by engineers and stakeholders. One important parameter for this analysis is the minimal cut sequences (MCSQs), that represent the minimal ordered failure sequence of events that cause the failure of the root event (Kabir et al., 2018). The quantitative analysis is related to measures that can be obtained from the model, often, the most referred ones are reliability, and the mean time to failure (MTTF). Reliability is the probability of the top event's failure, after a given time, which, mathematically, corresponds to the CDF of the top event. On the other hand, the MTTF computes the expected time of the root's failure (Volk et al., 2016).

Software tools, such as Storm, can calculate these indicators, using a formal method, like PRISM (Kwiatkowska et al., 2011). However, Storm uses two different logics, one to write the fault tree, that follows the Galileo format, and another to perform the quantitative analysis (PRISM). Moreover, these tools require the tree structure to be known and well defined previously, while the proposed method enables to find the structure from the generic distribution of BEs, which generate new knowledge from the system or part to be analyzed. For example, it helps to find out what BEs are responsible for the occurrence of top event, enabling a careful qualitative analysis, and timely schedule of maintenance interventions by engineers and technicians.

5. GFT Examples - Validation

In this section, one presents examples of application of the proposed method, and compares the results of the CDF of the top event obtained by the proposed approach and by Storm, for the exponential distribution case, and with the analytical expressions for other distributions. Note that the support considered for analysis was kept, [0,3.5].

5.1 The Exponential Distribution Case

For BEs following the exponential distribution, operators can be validated with Storm (Dehnert et al., 2016), a software tool already mentioned in Section 2. The operators AND, OR, and PAND, are compared with the results obtained by this tool, using metrics such as the mean absolute error (MAE), the RMSE, and L_∞, that are calculated by equations 15-17. Note that the assumptions that the operators FDEP and OR, and HSP and AND are equivalent, also confirmed numerically, using Storm.

To study the influence of h in the results, three different discretization steps were tested. The obtained results are summarized in Table 1. As can be seen, the results improve by refining the discretization step, and even for the bigger step, the results are acceptable. It is important to note that, besides the presented example, the approach was tested for a mesh of parameters λ_A and λ_B, between 0.2 and 3.0, with equivalent results.

© IEOM Society International

$$MAE_{X,Y} = \frac{1}{N}\sum_{j=1}^{N}\left|x_j - y_j\right|, \ j \in \{1,2,\dots,N\} \quad \textbf{(16)}$$

$$d_{X,Y} = \max\left(\left|x_j - y_j\right|\right), \ j \in \{1,2,\dots,N\} \quad \textbf{(17)}$$

Table 1: The exponential distribution case $\boldsymbol{A \sim Exp(\lambda = 1.8)}$, $\boldsymbol{B \sim Exp(\lambda = 0.4)}$.

Operator	step (*h*)	MAE	RMSE	L_∞
PAND	0.1000	5.7838E-03	7.1933E-03	1.3339E-02
	0.0500	1.3203E-03	1.6202E-03	3.0688E-03
	0.0050	1.2055E-05	1.4589E-05	2.8177E-05
AND	0.1000	1.8224E-02	2.1200E-02	3.2037E-02
	0.0500	4.1330E-03	4.7554E-03	7.2208E-03
	0.0050	3.7945E-05	4.3239E-05	6.5910E-05
OR	0.1000	1.5153E-02	1.6482E-02	2.2473E-02
	0.0500	3.6010E-03	3.8164E-03	5.1470E-03
	0.0050	3.4101E-05	3.5550E-05	4.7021E-05

5.2 The Normal Distribution Case

Storm only supports BEs that follow an exponential distribution, thus, for the normal distribution case, the validation was made, by comparing the results of the proposed approach, with the analytical results, by applying the algebraic expressions analyzed in Section 3.2. Note that generally the min or the max of two random variables following normal distributions is not a normal distribution and, for the PAND operator, the CDF expression requires the integration of the product between the CDF and the PDF of BEs with normal distributions (see equation 12), which cannot be computed exactly, thus the real results for this operator, were calculated, using the h-approximated method described by equation 2 to compute the integral, with a h-step of 0.0005. The results presented show minimal differences between the obtained results and the ground truth obtained analytically, and that the convergence error decreases with the refining of *h*, as shown in Table 2.

Table 2: The normal distribution case $\boldsymbol{A \sim N(\mu = 1.2, \sigma} = 0.3)$, $\boldsymbol{B \sim N(\mu = 2.2, \sigma} = 0.7)$.

Operator	step (*h*)	MAE	RMSE	L_∞
PAND	0.1000	4.7787E-03	5.689E-03	8.9312E-03
	0.0500	1.1394E-03	1.401E-03	2.2351E-03
	0.0050	1.0962E-05	1.367E-05	2.2239E-05
AND	0.1000	1.9191E-03	2.698E-03	5.2018E-03
	0.0500	4.8147E-04	6.661E-04	1.3073E-03
	0.0050	4.6489E-06	6.567E-06	1.3198E-05
OR	0.1000	7.8138E-03	1.220E-02	2.2587E-02
	0.0500	1.7630E-03	2.937E-03	7.1923E-03
	0.0050	1.7885E-05	2.915E-05	7.1664E-05

5.3 Finding a GFT Structure

To proceed with a first validation the training of a GFT, we use a well-known example about a motorbike failure in the literature, illustrated in Figure 2, was chosen. The four BEs, generally denoted by *A*, *B*, *C*, *and D*, follow exponential distributions, with parameters $\lambda_A = 2.0, \lambda_B = 0.8, \lambda_C = 1.2$ and $\lambda_D = 1.8$. The results were compared with the results provided by the software Storm, (Dehnert et al., 2016). Storm can solve a FT, where all the BEs follow the exponential distribution, as the one proposed in this example, by transforming it into the corresponding CTMC.

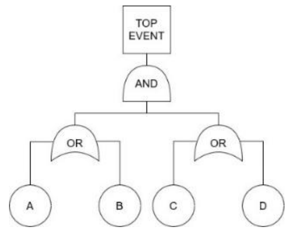
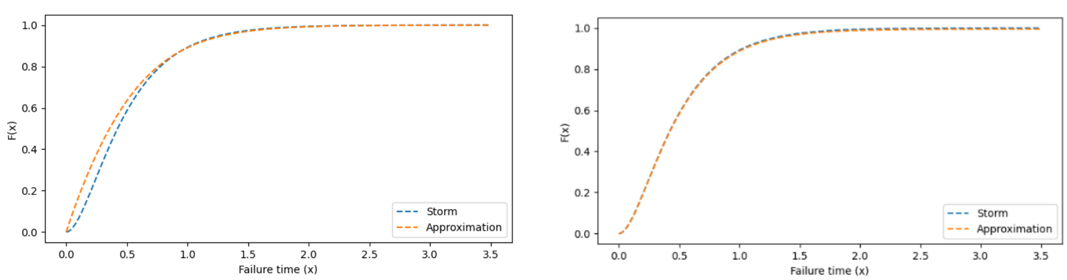


Figure 2: Skeleton of the tree used to train.

The tests were performed with two different discretization steps, namely, 0.0050 and 0.0005. The obtained results are represented in Figure 3. As can be seen, the output with the first step was a rough approximation of the real tree, while, by refining the step, the output quite corresponds to the real tree, where the approximation is much more precise. The outcome of the first training experiment can occur due to distinct factors, but the more obvious is the fact that the accuracy of the proposed numerical method strongly depends on the discretization step, *h*, thus, a larger *h* allow a bigger error so it is a cause of this issue, which can be confirmed by the fact that by refining the step, the approximation's accuracy improves significantly.

Some aspects must be accounted, the first is that the method of finding the structure of a tree, only having the root and the leaves fixed, may derive a tree that do not correspond exactly to the expected real tree. Nevertheless, such is not necessarily a drawback, since we show that both trees are “sufficiently” close numerically. Thus, different trees may produce similar same top event's distributions when fixing an error bound. The advantage of this approach is to allow the use of simple tree structures to approximate quite complicate real tree. On the other hand, besides the quantitative issues, in risk analysis and reliability, it is important to add a qualitative component which accounts for the sensibility of the main events that potentially cause a systems' failure is crucial and add previous acquired knowledge about the problem. However, most of the times, part of the structure of a DFT is somehow inferred by the engineering experts, thus, by exploiting this knowledge, only some subtree's must be found, mitigating the complexity problem of compromising the system's qualitative analysis.



(a) ***OR***(B, ***AND***(C,***OR***(A,D))) with *h*=0.0050. (b) ***AND***(***OR***(A,B),***OR***(C,D)) with *h*=0.0005.

Figure 3: Best approximations for the CDF reference values.

6. Real Use Case

6.1 System and Events Description

To illustrate the idea of the GFT approach, we use a real dataset from a stamping press from Bosch ThermoTechnology. The main goal is to generate the most suitable GFT structure that describes the mechanical faults of this equipment. The available data is collected from a Programmable Logic Controller (PLC), or from external sensors retrofitted to the press. A total of 25 variables are collected and monitored, however, only 7 are considered to have influence in mechanical failures, namely the oil temperature, oil level, motor speed, shaft vibration, the hydraulic pressure on the left and on the right connecting rod supercharge relief system, and, the tool height, since it is the differential characteristic between tools.
The dataset used in this use case comprises the measurements across 17-hours period, and was also employed in a previous work developed by Coelho et al. (2021). Before generating the histograms to build the GFT, the data was preprocessed and enriched. This process was made in the work developed Coelho et al. (2021), and was exploited for this use case. Since the pipeline and the platform used to collect the data is not the focus of this work, only a brief

description of the data processing will be made, for a good comprehension of this document. To have detailed information about the data preprocessing and general pipeline of the employed platform, we point the reader to (Coelho et al., 2021). Below, we describe the general steps performed before using the data:

- Real-time data segmentation of *Working* and *Stopped* states, to filter the relevant data (*Working state*) from the noisy data (*Stopped state*) and transition between states.
- Feature enrichment with anomaly detection (AD).
  - It was used the Principal Component Analysis (PCA) with 2 principal components (PCs);
  - The Copula-Based Outlier Detection (COPOD) algorithm was applied to the 2 PCs.
- The time series were aggregated in a 10-minute time window, and the aggregation primitives were calculated for all features, except the tool height.
- The most relevant features were selected, based on the correlation matrix, i.e., the features with more than 90% correlation were discarded.
- Two shifted features are created for each feature, containing its value for the two previous time steps.

The process described above produced 18 features, however, according to Coelho et al. (2021), some features had more importance than other in the XGBoost algorithm applied in their work. To speed up the GFT training process, only the top-6 features, which are described in Table 3, were used. Note that the presented variables are continuous, except *max_anomaly*, and the histograms must represent the time between the occurrence of events to model a GFT. Thus, the features were discretized in a maximum of 5 levels; Outlier Low (OL), Low (L), Normal (N), High (H) and Outlier High (OH), based on the features' quartiles. Each feature then generates a maximum of 5 new events, enabling the construction of a GFT.

Table 3: Description of the most relevant features.

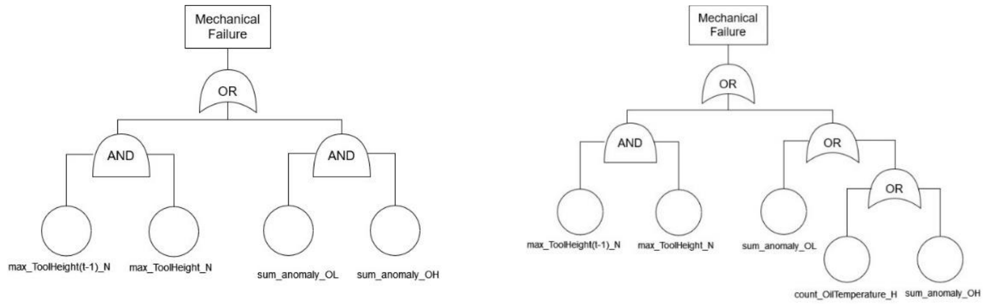
Feature	**Description**
sum_anomaly	Number of anomalies detected at the current time window.
count_OilTemperature	Number of oil temperature readings at the current time window.
max_anomaly	Maximum value for AD (1, if any anomaly was detected within time window, 0 otherwise).
consezeros_oilTemperature	Consecutive null count values for the oil temperature at the current time window
max_toolHeight	Maximum tool height at the current time window.
max_toolHeight(t-1)	Maximum tool height at previous time window.

6.2 Numerical Results

The exploited dataset was labeled in the format of a classification problem by Coelho et al. (2021), where the label assumes binary values, representing the absence of mechanical failures "0" or the existence of these failures "1", which is the root event. Then, the GFT was trained to generate the structure that best fits this distribution from the input BEs. The h-step used was 10 minutes, corresponding to the size of the aggregation time window. The process was executed considering 4 and 5 BEs. Spare operators, CSP, WSP and HSP were not considered since the idea of spare parts does not make sense in the context of the input BEs considered in this use case.

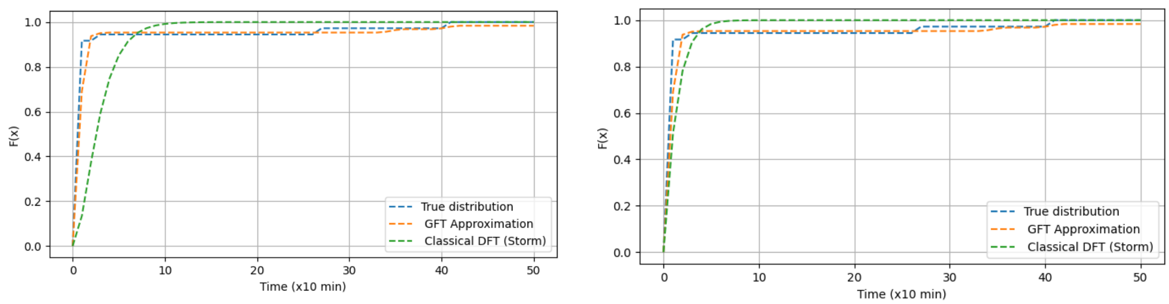
The obtained structures are represented in Figure 4, and the corresponding cumulative probability of failure is shown in Figure 5. The green curve was obtained with the storm software, by approximating the BEs distribution to an exponential, with $\lambda = 1/E[X_i]$, where $E[X_i]$ is the mean, or expected value of the random variables X_i. By analyzing the curves and the indicators presented in Table 4, can be concluded that the structure that best fits the data was obtained by the GFT method, with 4 BEs, which corroborates the fact that the approximation of BEs distribution commonly made in traditional methods is a rough approximation. Surprisingly, increasing the number of BEs did not contribute to improving the model, which may occur because not all the events from a dataset are needed to model properly the best GFT structure and just add noise to the model. However, it does not mean that the best structure only contains 4 BEs, a full search, considering all the possible number of BEs should be performed to claim it. Having the CDF of the root event, several questions may be made to the model, for example, what is the maximum time that can be elapsed between failures, until the failure probability reaches 80%? The response for this question is represented as $T_{p=0.8}$, in the last row of Table 4. The results of the classical DFT approach are clearly overestimated, when compared to the true distribution (8.7273 minutes), and the GFT with 4 BEs revealed the best results, showing to be a consistent method, even when new variables are added.

© IEOM Society International



(a) 4 BEs. (b) 5 BEs.

Figure 4: Best GFT structures obtained for the use case.



(a) 4 BEs (b) 5 BEs

Figure 5: Failure probability obtained from the best GFTs.

Table 4: Performance of the GFT training.

Approach	MAE	RMSE	L_∞	$T_{p=0.8}$ (minutes)
GFT 4 BEs	**1.5114E-02**	**3.2622E-02**	**2.1571E-01**	**13.8658**
DFT 4 BEs	6.6423E-02	1.4965E-01	7.8240E-01	45.0648
GFT 5 BEs	1.5572E-02	3.3924E-02	2.2518E-01	14.4155
DFT 5 BEs	4.2709E-02	7.1277E-02	4.0374E-01	21.4979

7. Conclusion

In this work, we addressed a key issue of classical DFT analysis, the fact that it assumes that the distribution of BEs follows the exponential/Weibull distribution. With the proposed GFT approach, BEs may follow an arbitrary compact support distribution or an h-truncation of a distribution**.** The numerical method was validated against traditional tools, such as Storm or analytical expressions. Furthermore, a full search process to find the best tree's structure that fits the data was validated. This method was applied to a real use case, a stamping press from Bosch ThermoTechnology, where all BEs were generated from sensor data. The training of the GFT to obtain the best tree structure allows the selection of BEs that influence the occurrence of the top event. In the case of Bosch ThermoTechnology's stamping press, these BEs are: *sum_anomaly_OL*, *sum_anomaly_OH*, max*_toolHeight_N* and *max_toolHeight(t-1)_N*. A simple structure, with only 4 BEs can represent the true distribution of mechanical failures, with a good accuracy, considering metrics such as MAE, RMSE and $\mathbf{L_\infty}$. Moreover, the proposed GFT demonstrated to be more effective in a real use case, when compared to the traditional DFT analysis where the distribution of BEs is approximated to the exponential distribution.

However, this model has some limitations, namely, a full search, considering all the possible number of BEs is too exhaustive, and demands high computational capabilities, thus a new method to accelerate the search, such as meta-heuristic algorithms will be considered in future works. Furthermore, the exploited dataset with 10-minute aggregations is short, and the approach should be tested in larger datasets. The process of GFT training can be accelerated if a part of the GFT structure is known and provided by technicians or experts. This knowledge, when it exists, should be exploited by the model. Finally, there are mathematical demonstrations, namely, the expressions of approximation errors for each operator, that were not presented here since they are out of the scope of this document but will be published in a more technical/theoretical article.

Acknowledgements
The present study was partially developed in the scope of the Project Augmented Humanity (PAH) [POCI-01-0247-FEDER-046103], financed by Portugal 2020, under the Competitiveness and Internationalization Operational Program, the Lisbon Regional Operational Program, and by the European Regional Development Fund. The first author was partially supported by the Center for Research and Development in Mathematics and Applications (CIDMA), through the Portuguese Foundation for Science and Technology, reference UIDB/04106/2022. The second author has a PhD grant supported FCT – Fundação para a Ciência e a Tecnologia, I.P. for the PhD grants ref. 2020.06926.BD. The second and third authors would like to acknowledge the University of Aveiro, FCT/MCTES for the financial support of TEMA research unit (FCT Ref. UIDB/00481/2020 \& UIDP/00481/2020) and CENTRO01-0145-FEDER-022083 - Regional Operational Program of the Center (Centro2020), within the scope of the Portugal 2020 Partnership Agreement, through the European Regional Development Fund.

Data availability statement
The data sets used in this work are confidential information of Bosch company manufacturing system, so they are not publicly available.

References
Arnold, F., Belinfante, A., Van der Berg, F., Guck, D., & Stoelinga, M. (2013). *DFTCalc: A Tool for Efficient Fault Tree Analysis* (pp. 293–301). https://doi.org/10.1007/978-3-642-40793-2_27

Aslansefat, K., Kabir, S., Gheraibia, Y., & Papadopoulos, Y. (2020). Dynamic Fault Tree Analysis. In *Reliability Management and Engineering* (Issue December, pp. 73–112). CRC Press. https://doi.org/10.1201/9780429268922-4

Aslansefat, K., & Latif-Shabgahi, G. R. (2020). A Hierarchical Approach for Dynamic Fault Trees Solution through Semi-Markov Process. *IEEE Transactions on Reliability*, *69*(3), 986–1003. https://doi.org/10.1109/TR.2019.2923893

Boudali, H., Nijmeijer, A. P., & Stoelinga, M. I. A. (2009). DFTSim: A simulation tool for extended dynamic fault trees. *42nd Annual Simulation Symposium 2009, ANSS 2009, Part of the 2009 Spring Simulation Multiconference*, 9–16.

Boudali, Hichem, Crouzen, P., & Stoelinga, M. (2007). *Coral : a tool for Compositional Reliability and Availability analysis † Input / Output Interactive*. *004527*(July), 1–20.

Cheshmikhani, E., & Zarandi, H. R. (2015). Probabilistic analysis of dynamic and temporal fault trees using accurate stochastic logic gates. *Microelectronics Reliability*, *55*(11), 2468–2480. https://doi.org/10.1016/j.microrel.2015.06.047

Chiacchio, F., Iacono, A., Compagno, L., & D'Urso, D. (2020). A general framework for dependability modelling coupling discrete-event and time-driven simulation. *Reliability Engineering and System Safety*, *199*(January), 106904. https://doi.org/10.1016/j.ress.2020.106904

Coelho, D., Costa, D., Rocha, M., & Almeida, D. (2021). Predictive maintenance on sensorized stamping presses by time series segmentation , anomaly detection , and classification algorithms. *International Conference on Industry 4.0 and Smart Manufacturing*.

Dehnert, C., Junges, S., Katoen, J.-P., & Volk, M. (2016). *The Probabilistic Model Checker Storm (Extended Abstract)*. http://arxiv.org/abs/1610.08713

Dugan, J. B., Bavuso, S. J., & Boyd, M. A. (1992). Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transactions on Reliability*, *41*(3), 363–377. https://doi.org/10.1109/24.159800

Durga Rao, K., Gopika, V., Sanyasi Rao, V. V. S., Kushwaha, H. S., Verma, A. K., & Srividya, A. (2009). Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment. *Reliability Engineering and System Safety*, *94*(4), 872–883. https://doi.org/10.1016/j.ress.2008.09.007

Ejlali, A., & Ghassem Miremadi, S. (2004). FPGA-based Monte Carlo simulation for fault tree analysis. *Microelectronics Reliability*, *44*(6), 1017–1028. https://doi.org/10.1016/j.microrel.2004.01.016

Elderhalli, Y., Hasan, O., & Tahar, S. (2019). A Methodology for the Formal Verification of Dynamic Fault Trees Using HOL Theorem Proving. *IEEE Access*, *7*, 136176–136192. https://doi.org/10.1109/ACCESS.2019.2942829

Jiang, G.-J., Li, Z.-Y., Qiao, G., Chen, H.-X., Li, H.-B., & Sun, H.-H. (2021). Reliability Analysis of Dynamic Fault Tree Based on Binary Decision Diagrams for Explosive Vehicle. *Mathematical Problems in Engineering*, *2021*, 1–13. https://doi.org/10.1155/2021/5559475

Kabir, S. (2017). An overview of fault tree analysis and its application in model based dependability analysis. *Expert Systems with Applications*, *77*, 114–135. https://doi.org/10.1016/j.eswa.2017.01.058

Kabir, S., Aslansefat, K., Sorokos, I., Papadopoulos, Y., & Konur, S. (2020). A Hybrid Modular Approach for Dynamic Fault Tree Analysis. *IEEE Access*, *8*, 97175–97188. https://doi.org/10.1109/ACCESS.2020.2996643

Kabir, S., Walker, M., & Papadopoulos, Y. (2018). Dynamic system safety analysis in HiP-HOPS with Petri Nets and Bayesian Networks. *Safety Science*, *105*(February), 55–70. https://doi.org/10.1016/j.ssci.2018.02.001

Khakzad, N., Khan, F., & Amyotte, P. (2013). Risk-based design of process systems using discrete-time Bayesian networks. *Reliability Engineering and System Safety*, *109*, 5–17. https://doi.org/10.1016/j.ress.2012.07.009

Kwiatkowska, M., Norman, G., & Parker, D. (2011). PRISM 4.0: Verification of Probabilistic Real-Time Systems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 6806 LNCS* (pp. 585–591). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-22110-1_47

Merle, G., Roussel, J.-M., & Lesage, J.-J. (2010). Improving the Efficiency of Dynamic Fault Tree Analysis by Considering Gate FDEP as Static. *European Safety and Reliability Conference(ESREL 2010)*, 845–851. https://hal.archives-ouvertes.fr/hal-00516896

Merle, G., Roussel, J., Lesage, J., Merle, G., Roussel, J., Dynamic, J. L., Tree, F., Based, A., Merle, G., Cachan, E. N. S., Roussel, J., & Cachan, E. N. S. (2011). Dynamic Fault Tree Analysis Based On The Structure Function. *Annual Reliability and Maintainability Symposium 2011 (RAMS 2011)*.

Merle, G., Roussel, J. M., Lesage, J. J., & Vayatis, N. (2010). Analytical calculation of failure probabilities in dynamic fault trees including spare gates. *European Safety and ReliabilityConference (ESREL 2010), September*, 794–801.

Ruijters, E., & Stoelinga, M. (2015). Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. In *Computer Science Review* (Vol. 15, pp. 29–62). Elsevier Ireland Ltd. https://doi.org/10.1016/j.cosrev.2015.03.001

Sihite, J. F., & Kohda, T. (2013). Assessing the reliability of power transformer by quantitative fault tree analysis. *Advanced Materials Research*, *694 697*, 901–906. https://doi.org/10.4028/www.scientific.net/AMR.694-697.901

Stamatelatos, M., Caraballo, M. J., Vesely, W., Dugan, J., Fragola, M. J., Minarick, M. J., Railsback, M. J., & Jsc, N. (2002). *Fault Tree Handbook with Aerospace Applications*.

Sullivan, K. J., Dugan, J. B., & Coppit, D. (2003). The Galileo fault tree analysis tool. *Digest of Papers. Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing (Cat. No.99CB36352), December*, 232–235. https://doi.org/10.1109/FTCS.1999.781056

Volk, M., & Junges, S. (2018). by Model Checking Techniques. *IEEE Transactions on Industrial Informatics*, *14*(1), 370–379.

Volk, M., Junges, S., & Katoen, J.-P. (2016). *Advancing Dynamic Fault Tree Analysis - Get Succinct State Spaces Fast and Synthesise Failure Rates* (pp. 253–265). https://doi.org/10.1007/978-3-319-45477-1_20

Xu, Z., Guo, D., Wang, J., Li, X., & Ge, D. (2021). A numerical simulation method for a repairable dynamic fault tree. *Eksploatacja i Niezawodnosc*, *23*(1), 34–41. https://doi.org/10.17531/EIN.2021.1.4

Zhu, P., Han, J., Liu, L., & Zuo, M. J. (2014). A stochastic approach for the analysis of fault trees with priority and gates. *IEEE Transactions on Reliability*, *63*(2), 480–494. https://doi.org/10.1109/TR.2014.2313796

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers and Industrial Engineering*, *150*(August), 106889. https://doi.org/10.1016/j.cie.2020.106889

Biographies

Eugénio Rocha is an Associate Professor at the Department of Mathematics, University of Aveiro, Portugal, with a PhD and MSc in Mathematics and graduation in Computer Science (artificial intelligence). He is director of the Masters in Mathematics and Applications, and vice-director of the Masters in Data Science, (co-)coordinates two thematic lines of the research unit CIDMA, has more than 90 publications in pure mathematics or their applications to different topics in Biological Systems and Epidemiology, Civil Engineering, Computational Chemistry and Nanotechnology, Econometrics, Education, and Mechanical Engineering. Participated in more than 30 projects (two with National/European prizes). Recently, he has coordinated projects involving data-driven smart solutions related with applications of Machine Learning and Big Data in the scope of the Industry 4.0.

Pedro Nunes is a PhD student in the Department of Mechanical Engineering, at University of Aveiro, Portugal. He concluded his MSc in Mechanical Engineering at the same institution, in 2018. He was a research fellow between 2018-2020 at TEMA (Centre for Mechanical Technology and Automation), published works in the context of bike routing, multi-objective routing, and sensorization of soft mobility vehicles. His areas of interest include industrial IoT, Industry 4.0 and meta-heuristic algorithms.

José Paulo Oliveira Santos is a professor at the University of Aveiro, graduated in Electronic Engineering and Telecommunications at the University of Aveiro, obtained his Master's Degree in "Computer and Communications Engineering - branch of computer networks" at the University of Coimbra and his PhD in Industrial Automation at the University of Aveiro. Since 1995, he has taught several disciplines, namely, Drive and Command Technologies, Integrated Production Control, Intelligent Manufacturing Systems (IMS), Industrial Informatics, Automation II, Computer Numerical Command (CNC), Introduction to Informatics, Programming Data Structures and Algorithms, Electrotechnology, Industrial Automation, and Flexible Manufacturing Systems (FMS). Has participated and coordinated several projects between the University of Aveiro and several enterprises. He supervised more than 100 theses, several in industrial companies and was Director of the Master in Industrial Automation Engineering.

© IEOM Society International