Dependability Analysis using Temporal Fault Trees and Monte Carlo Simulation

Ernest Edifor[1][0000-0001-9768-7360], Neil Gordon[2][0000-0001-6889-0781] and Martin Walker[3][0000-0002-6380-2313]

[1] Manchester Metropolitan University, Manchester, M15 6BH, UK
[2] University of Hull, Hull, HU6 7RX, UK
[3] Dymodian Systems Ltd, Hull, HU1 1TJ, UK
e.edifor@mmu.ac.uk

Abstract. The safety and reliability of high-consequence systems is an issue of utmost importance to engineers because such systems can have catastrophic effects if they fail. Fault Tree Analysis (FTA) is a well-known probabilistic technique for assessing the reliability of safety-critical systems. Standard FTA approaches are primarily static analysis techniques and as such cannot effectively model systems with dynamic behaviours, such as those with standby components or multiple modes of operation. There have been several efforts to address this limitation, one of which is Pandora, a temporal fault tree approach. Pandora uses three temporal gates — Priority-AND, Simultaneous-AND, and Priority-OR — to model the effects of sequences of events. Hitherto, Pandora was unable to perform a holistic evaluation of a full system that is repairable, taking account of useful system operating environment variables (such as time of operation, flow rate, etc.) or system data such as repair state and preventive maintenance. This paper aims to address these limitations. Algorithms to evaluate different system configurations have been generated and techniques for modelling and analyzing different system data in a simulation platform have been proposed. This paper extends the capabilities of Pandora so that it is capable of analyzing a modern system that features different failure modes, has diverse component failure distributions, considers the system's operation environment data, and models different system configurations. The outcome of such analysis enables analysts to understand the operation and dynamics of a system holistically and aids in the implementation of appropriate risk mitigating strategies.

Keywords: temporal fault tree analysis; dependability, Monte Carlo simulation.

1 Introduction

Safety-critical systems are high consequence systems that can have detrimental effects on the environment or human life if they should fail. Fault Tree Analysis (FTA) [1] is one of the Probabilistic Risk Assessment (PRA) techniques used to evaluate the reliability of systems. FTA is typically used to identify the combinations of component faults (basic events) that can lead to the occurrence of a system failure (top event).

Fault trees are based on Boolean logic and most commonly use OR gates and AND gates to connect events. Fault tree analyses can be performed both qualitatively (logically) and quantitatively (probabilistically). The logical analysis involves using Boolean algebra to derive minimal cut sets (MCSs). An MCS is the smallest logical combination of basic events that will trigger the occurrence of a top event. Qualitative analysis helps to identify critical and non-critical components of the system, as well as providing a better understanding of the causes of system failures. Quantitative analysis involves the mathematical evaluation of the top event probability and the determination of importance measures (components contribute to the top event occurrence). The quantitative analysis gives analysts an indication of how likely a system is to fail, as well as the contribution of each component to the system failure.

Despite its widespread use, traditional FTA has some drawbacks. Various efforts have been made to extend FTA. One relatively recent solution is Pandora [2], a temporal fault tree technique. It maintains the structure of FTA but extends its laws and semantics with the introduction of three temporal gates – Priority-AND, Simultaneous-AND, and Priority-OR. These new gates enable the representation and analysis of sequences of events in fault trees.

Pandora's logical analysis capabilities allow it to eliminate redundancies and contradictions in the fault tree to produce a set of minimal cut sequences (MCSQs), which are analogous to MCS but which retain information about the order of events. Pandora is capable of performing a comprehensive qualitative analysis. Analytical and simulation approaches have been proposed to allow Pandora to perform quantitative analysis. However, these techniques are restricted to at least one of the following: unable to perform full quantitative analysis, limited to an exponential distribution, limited to non-repairable events, unable to capture system environment data (such as flow-rate), limited to producing only reliability data.

In this paper, a Monte Carlo based solution is presented to enable the full quantitative analysis of Pandora. This solution includes the definition of a simulation procedure for modelling and evaluating different types and combinations of MCSQs. An alternative solution is also prescribed to allow the dependability analysis of Pandora to take place by using an established simulation software platform based on Monte Carlo simulation. The second solution involves the creation and modelling of temporal gates and behaviours on the simulation platform. The outcome of these solutions produces different quantitative data such as reliability, availability, mean-time-to-failure, and criticality measures. These enable the quantitative analysis of modern systems that feature different component failure distributions, have repair and maintenance regimes, feature different system configurations, and are responsive to changes in their operating environment.

The remaining sections of this paper are arranged as follows: Section 2 describes temporal gates and the mathematical models and evaluations of these gates using analytical and simulation approaches. Section 3 contains new Monte Carlo simulation algorithms and modelling techniques for evaluating different types of MCSQs in temporal fault trees. Section 4 shows the application of the proposed solutions to a hypothetical autonomous underwater vehicle. Concluding remarks are given in Section 5.

2 Related Work

Due to the popularity and long history of FTA, many research efforts have focused on expanding FTA to enhance its capabilities and encompass a greater range of systems. One of the best-known techniques is the Dynamic Fault Tree (DFT) technique, an extension of static FTA that was proposed to tackle computer-based systems in which the outcomes are affected by the order of occurrence of events [3]. Like FTA, DFT has seen several further developments, extensions, and enhancements for both qualitative and quantitative analysis [4,5], but it remains primarily a quantitative analysis technique. There are other approaches, such as the Temporal Fault Tree (TFT) technique by Palshikar [6]. A temporal fault tree is one in which temporal dependencies between events can be specified. Palshikar's extension of the classical FT introduces new operators and it is intended primarily to aid in the post-failure diagnosis of log data. Kabir [7] provides a critical review and evaluation of various FTA techniques.

2.1 Pandora

Pandora [2] is a more recent technique for analyzing temporal fault trees. It introduces three new gates: the Priority-AND (PAND), Simultaneous-AND (SAND), and Priority-OR (POR). The PAND (Priority-AND) is true if an input event occurs strictly before another input event in the left-to-right order (leftmost first). It only becomes true when its last input event occurs. The PAND is represented by the symbol '<', thus '*A*<*B*' indicates that *A* should have occurred before *B*.

The POR (Priority-OR) is true only if its first input event occurs before any other input event, but unlike the PAND, those other input events are not strictly required to occur. Thus whereas the PAND is a type of conjunction, the POR is a type of disjunction and represents a type of trigger condition. In Pandora, the symbol '|' represents POR; '*A*|*B*' means '*A* POR *B*', and is true if either *A* occurs and *B* does not, or both occur, with *A* occurring before *B*.

The SAND (Simultaneous-AND) gate is used to represent the situation where all input events occur at the same time. '&' is used to indicate a SAND gate: '*A*&*B*' means '*A* SAND *B*', and is true if *A* and *B* occur simultaneously. Throughout this paper, Boolean AND is represented with the symbol '•' and Boolean OR with the symbol '+'.

2.2 Quantitative Analysis using Analytical Approaches

In terms of quantitative analysis, techniques exist for the probabilistic evaluation of the Boolean gates (AND and OR) [1], PAND gate [7], and POR gate [8]. Nearly simultaneous events (that is, events that occur together within a small interval of time) can be evaluated using the pSAND according to a technique presented in [9]. Unless otherwise stated, it is assumed that E is a set of n basic events such that $E = \{X_1, X_2, ... X_{n-1}, X_n\}$ with corresponding failure rates $\lambda_1, \lambda_2, ... \lambda_{n-1}, \lambda_n$, where $0 \leq \lambda_i \leq 1$ respectively. Equations for the AND and OR gates can be found in [1]. Equations 1 to 3 are analytical formulae for the PAND, POR, and pSAND; it is assumed that all events have exponential failure distribution.

Henceforth, any reference to an analytical solution of PAND refers to the solution proposed by Fussell et al., [10] in (1), where $a_0 = 0$ and $a_m = -\sum_{j=1}^{m} \lambda_i$ for $m > 0$.

$$P\{X_n < X_{n-1} < \cdots < X_2 < X_1\}(\mathrm{t}) = \prod_{i=1}^{n} \lambda_i \sum_{k=0}^{n} \left[\frac{e^{(a_k t)}}{\prod_{\substack{j=0 \\ j \neq k}}^{n} (a_k - a_j)} \right] \tag{1}$$

The mathematical expression for calculating the probability of MCSQs with two or more POR gates at a time t is explained in [8]. The formula is shown in (2).

$$P\{X_1 | X_2 | \ldots | X_{n-1} | X_n\}(t) = \frac{\lambda_1 \left(1 - \left(e^{-(\sum_{i=1}^{n} \lambda_i) t} \right) \right)}{\sum_{i=1}^{n} \lambda_i} \tag{2}$$

Equation [9] is the formula for evaluating the pSAND gate for an MCSQs with two or more POR gates given that the interval between the occurrences of all events is d.

$$P\{X_1 \&_d X_2 \&_d \ldots \&_d X_n\}(t_0, t_1) = \sum_{i=1}^{n} \left(\left(1 - e^{\lambda_i t_0}\right) \bullet \left(\prod_{\substack{j=1 \\ j \neq i}}^{n} \left(1 - e^{\lambda_j (t_1 - t_0)}\right) \right) \right) \tag{3}$$

2.3 Quantitative Analysis using Simulation

A simulation is a means of learning something about the real world by replicating a scenario using a model. Simulations are used in situations where real-world scenarios are financially costly, could be dangerous, are overly complicated to design, or are too time-consuming to implement. Monte Carlo (MC) simulation is a popular simulation technique used in various fields such as chemistry, engineering, medicine, games, finance, and telecommunications. MC begins with modelling the system under study. Once this is done, the model is simulated or 'run' by generating random numbers for the model variables to create a unique 'instance' of the model. The system variables are generated several times, called trials, to create several instances of the model. These instances are examined for some common predetermined property, which eventually determines the behaviour of the model.

Simulation algorithms have been developed [11] for the PAND, POR and pSAND gates. There are techniques [11] for evaluating MCSQs with multiple events but one type of operator, MCSQs with multiple events and multiple different operators, and MCSQs with different events, operators and failure distributions. Unfortunately, these techniques struggle to holistically analyze real-world systems with all of the following features: allows for repairs, maintenance, and replacements, considers other system data apart from failure data and capture different system configurations. In this paper, a full quantitative analysis of temporal fault trees which overcome these limitations will be presented using algorithms for simulation and a technique for modelling temporal fault trees in the Goldsim software.

3 Dependability Analysis

For an MCSQ with input events X_1, X_2, ..., X_n, the input events could be basic or intermediate. Regardless of the type of input events an MCSQ has, an intermediate event can be considered an input event depending on the investigators level of abstraction. Given that an input event X_i is basic, $P(X_i)\{t\} = F(X_i)\{t\}$, where the function P is the probability of failure and F is the cumulative distribution function (CDF). However, if an input event X_i is an intermediate event, $P(X_i)\{t\}$ will have to be evaluated based on the individual sub-input events of X_i. To evaluate X PAND Y, where X has an exponentially distributed failure rate of λ and Y has Weibull failure distribution with α and β representing the scale and shape parameters respectively, the simulation condition can be generated as $R_A < =F_E(\lambda_A)\{t\}$ && $R_B <=F_W(\alpha_B, \beta_B)\{t\}$ && $TTF_E(\lambda_A,R_A)\{t\}< TTF_W(\alpha_B, \beta_B,R_B)\{t\}$. Where F_E and F_W are the CDFs of the events A and B respectively and TTF_E and TTF_W are the time-to-failure (TTF) of A and B respectively.

3.1 Evaluating all MCSQs

The top-event evaluation of a temporal fault tree with only exponentially distributed component failure modes is straightforward but a simulation approach is relatively slower to compute and produces estimates, although it is not restricted to any particular failure distribution. Combining both analytical and simulation approaches to harnesses the strengths of both techniques has been proven to be very useful [12]. Algorithm 1 is proposed for the top event evaluation of a fault tree with n MCSQs. *X[n]* represents the *n*th MCSQ, *static* represents events with no dynamic gates and *FTA* is a function that evaluates Boolean expressions [1]. *ANA* is a function that evaluates non-static MCSQs with components that have only exponentially distributed failure modes using their analytical equations (1), (2), and (3). *SIM* is a function that evaluates MCSQs with any combination of different failure distributions [11]. *EP* is a function that uses the Esary-Proschan formula in [13] to calculate the top event probability.

Algorithm 1. Evaluation of top-event of a dynamic system.

```
Require: X
Z ← 0
for i = 1 to n do
     if (X[i] is static) then
          Z ← Z + FTA (X[i])
     else
          if (X[i] is exp) then
               Z ← Z + ANA (X[i])
          else //not exponential distribution
               Z ← Z + SIM (X[i])
          end if
     end if
end for
return EP (Z)
```

3.2 Modelling in Goldsim

To ensure that temporal fault trees can capture repair and maintenance data and different system operating environment data [14] such as flow-rate, time of operation, etc., the Goldsim software [15] will be used. Goldsim is a software that allows the modelling and probabilistic analysis of complex systems based on Monte Carlo simulation. It has elements for representing static AND and OR Boolean gates. However, it has no preconstructed elements for the PAND, POR, or pSAND gates and it cannot perform qualitative analysis. Once a qualitative analysis has been performed with Pandora, it can be modelled in Goldsim.

Dependability analysis in Goldsim can be performed in one of two ways: using Fault Tree Analysis (FTA) or using Requirement Tree Analysis (RTA). Fig. 1 is an example of an FTA model of the PAND gate in Goldsim. The fuel cell is the primary power source so it is initially turned on while the diesel secondary source is turned off. The sensor needs to be turned on when the system starts so that it can activate the diesel engine if the fuel cell engine fails. When the fuel cell stops operating, the sensor activates the diesel system; when the fuel cell resumes operation, the diesel system is turned off. RTA is based on a set of conditions necessary for a system or component to succeed. RTA in Goldsim is rather simple to implement. For the entire dual-fuel engine to work, the input command must be issued, the engine and the fuel-cell or diesel subsystems must be working. RTA can be used without the manual construction and modelling of temporal gates.

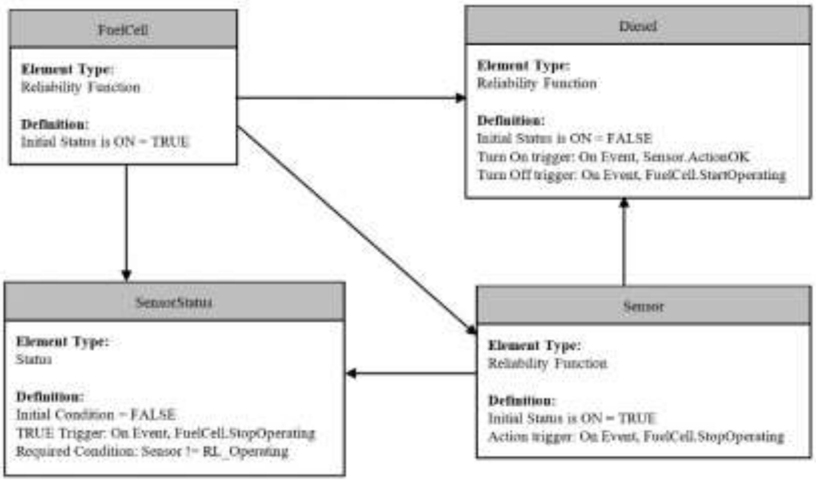
FuelCell
Element Type:
Reliability Function
Definition:
Initial Status is ON = TRUE
Diesel
Element Type:
Reliability Function
Definition:
Initial Status is ON = FALSE
Turn On trigger: On Event, Sensor.ActionOK
Turn Off trigger: On Event, FuelCell.StartOperating
SensorStatus
Element Type:
Status
Definition:
Initial Condition = FALSE
TRUE Trigger: On Event, FuelCell.StopOperating
Required Condition: Sensor != RL_Operating
Sensor
Element Type:
Reliability Function
Definition:
Initial Status is ON = TRUE
Action trigger: On Event, FuelCell.StopOperating

Fig. 1. Model of a hybrid fuel system in Goldsim.

It is well known that, unlike analytical approaches that produce exact results, simulation approaches produce approximated results. Unfortunately, there are no analytical approaches for performing the full quantification of Pandora, which considers different system configurations and failure distributions. The Goldsim simulation software uses state-of-the-art sampling algorithms to improve the accuracy of its results.

4 Case Study

To illustrate the above techniques, the authors have designed a hypothetical Autonomous Underwater Vehicle (AUV) to help solve one of the big concerns of the modern world – to collect microplastics from the ocean. The setup of the vehicle is such that it forms part of a collection of similar robots – referred to as a "shoal"– which is deployed from a mother sea vessel. The shoal functions as a large organism via artificial intelligence. A control centre on-board the vessel is kitted to control and monitor the shoal as a unit, or each seabot. For example, the shoal can be controlled to create a formation, to move to a different location, or to move to the mother vessel for safekeeping before a storm strikes. Each robotic AUV unit (called a seabot) functions autonomously but is capable of communicating with other seabots. Fig. 2 is a simplified version of a seabot.

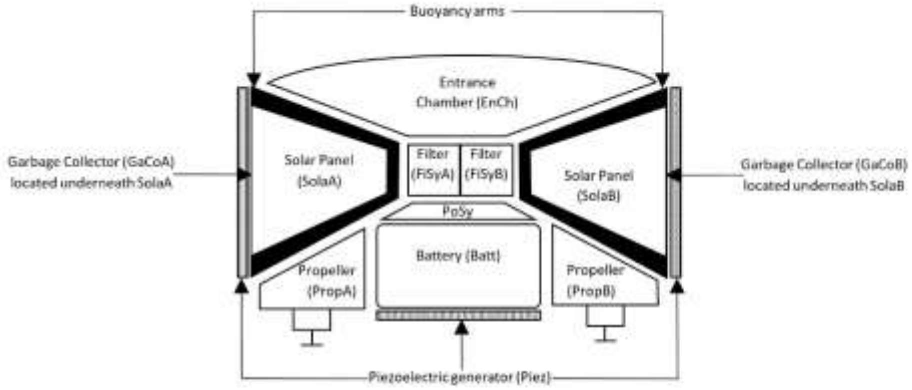

Fig. 2. An abstract model of the proposed seabot.

The seabot has a dish-like feature, known as the entrance chamber (EnCh) that collects seawater. The EnCh is designed with various dome-like features to prevent sea animals and plants from entering two filtering systems (FiSy) connected to it. The flow of seawater is facilitated by a pump (Pump) that is situated within each FiSy. The FiSy also contains a valve that allows water into it from the pump and prevents water from going back through the EnCh. A flow meter (FoMe) controls the rate of water the pump should allow depending on the reading of the level detector (LeDe).

The microplastic isolation system (MiSy) is responsible for separating the microplastics from the seawater within the FiSy and depositing the residue into either of the garbage collectors (GaCo) situated under the solar panels (SoPa). Each GaCo stores the microplastics as long as its level detector (GaDe) does not read full. When both GaCos are full, the seabot moves to the mother vessel for its GaCos to be replaced with empty ones. The GaDe are also able to detect the situation where its GaCo bursts. In the event where a GaCo bursts, the seabot will signal some of the seabots closest to it to assist in collecting the microplastics it is releasing back into the sea. The seabot will then move to the mother vessel for the defective GaCo to be replaced.

There are two onboard propellers or thrusters (Prop) that move the seabot backwards or forward and two buoyancy arms (Bouy) that are responsible for the floating and sinking of the seabot. Collision sensors are fixed on all sides of the seabot to allow it to avoid collision with other objects during navigation. All operations of the seabot are powered by a stack of batteries (Batt). During the day, the seabot operates afloat and harnesses solar power (Sola) to charge the Batts. During underwater movements, cilia boards located on the sides of the seabot agitate piezoelectric actuators (Piez) to generate electricity to charge Batts. In the event where the Batt fails, the seabot is powered by either the SoPa and/or the Piez if they can provide enough energy required. There are various controllers responsible for fault diagnosis (FaDi), filtering system (FiSy), communication system (CoSy), power management system (PoSy), and navigation system (NaSy).

To perform qualitative analysis, the data in Table 1 are assumed. From the table, λ represents the hazard rate. α and β represent the scale and shape parameters of a Weibull distribution respectively, and μ and δ are the mean and standard deviation of a lognormal distribution respectively. (F) and (R) stand for failure and repair data respectively. A full qualitative analysis and dependability analysis of the entire seabot system is outside the scope of this paper. This paper will consider only the operations necessary for collecting the microplastics from the EnCh and storing them in the GaCo. After a qualitative analysis using techniques in [2], the temporal fault tree can be represented by the following CSQ expression.

$$\begin{aligned}\mathit{Top\text{-}event} = {} & \mathit{EnCh} + (\mathit{FiSyA} \,.\, \mathit{FiSyB}) + (\mathit{GaCoA} \& \mathit{GaCoB}) + (\mathit{Batt} < \mathit{Sola}) \,.\\ & (\mathit{Batt}|\mathit{Piez}) \,.\, (\mathit{Sola}|\mathit{Piez}) + [\mathit{PoSy}|\mathit{Sola}] + [\mathit{PoSy}|\mathit{Batt}] + [\mathit{PoSy}|\mathit{Piez}]\end{aligned}$$

Table 1. System Operating Data (In Days or Per Day).

Entity	Failure Type	λ (F)	α (F)	β (F)	λ (R)	μ (R)	δ (R)
Seabot	General system failure	2.33E-5	-	-	2.0	-	-
Seabot	Preventive maintenance	2.74E-3	-	-	-	0.5	0.0
EnCh	Blocked/covered	1.37E-3	-	-	-	0.8	0.2
FiSyA	Internal failure	2.74E-3	-	-	-	1	0.5
FiSyA	Valve stuck closed	-	480	1.5	-	1	0.5
FiSyB	Internal failure	2.74E-3	-	-	-	1	0.5
FiSyB	Valve stuck closed	-	480	1.5	-	1	0.5
Batt	Internal failure	1.10E-3	-	-	-	1	0.25
Piez	Internal failure	1.34E-3	-	-	-	1	0.5
Sola	Internal failure	1.83E-3	-	-	-	1	0.75
PoSy	Internal failure	-	635	2	0.5	-	-
GaCoA	Valve stuck closed	-	522	1	-	0.5	0.25
GaCoA	Replacement	-	-	-	0.1	-	-
GaCoB	Valve stuck closed	-	522	1	-	0.5	0.25
GacoB	Replacement	-	-	-	0.1	-	-

It is assumed that the solar panels are turned on and off between 09.00 and 17.00 from January to March, 08.00 to 18.00 from April to June, 06.00 to 20.00 from July to September and 07.00 to 17.00 on the other months. The garbage collector has a maximum capacity of 100m3 and a rate of addition with an exponential distribution mean of 1.5m3/day. When both garbage collectors are full, the seabot moves to the mother shipping vessel for replacements. Annual preventive maintenance is scheduled for each seabot on the first of June. Using the proposed algorithms, the case study was modelled and run using Monte Carlo simulation. The simulation was computed for a system lifetime from 1 to 50 days with 100000 iterations per day and a time-step of 1 hour. Only failure data was used in the simulation. The result of the simulation is displayed in Fig. 3.

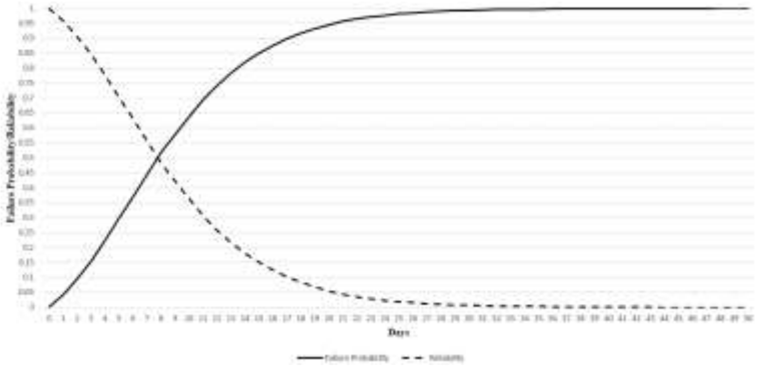

Fig. 3. Result of Monte Carlo simulation with only failure data.

It is clear that the probability of the top-event occurring increases with an increasing system lifetime. By the 50th day, it is expected that the top-event should have occurred – that is, the system should have failed. Modelling the same failure data, including the repair and maintenance data and system operating data in Goldsim using 5000 iterations over 50 days with a time-step of 1 hour produces the result in Fig. 4.

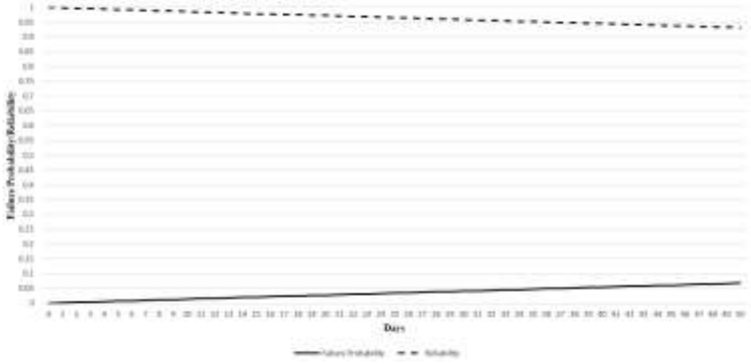

Fig. 4. Result of Monte Carlo simulation with all system data.

A lower number of iterations is used in Fig. 4 because a sampling technique (Latin Hypercube Sampling) is implemented. Given repair, maintenance, and failure data, the mean system reliability stays over 90% after the 50th day. Such results are expected because if components are maintained and repaired, the entire system fails far less. Other quantitative measures can be extracted from the Goldsim simulation. In Table 2, *FP*, *Rel.*, *TF*, and *TR* are the failure probability, reliability, mean-time-to-failure (MTTF), and mean-time-to-repair (MTTR) respectively. *FP*, *Rel.*, *MTTF* and *MTTR* retain their original definitions from [1]. *IA* is the inherent availability and *OA* is the operational availability [15].

Table 2. Seabot Reliability Analysis.

Entity	FP	Rel.	IA	OA	TF (days)	TR (days)
FiSyA	0.151	0.849	0.997	0.986	260	0.977
GaCoB	0.091	0.909	0.999	0.988	491	0.518
GaCoA	0.09	0.91	0.999	0.988	479	0.507
Seabot	0.068	0.932	0.989	0.989	696	0.791
Piez	0.064	0.936	0.999	0.988	746	0.986
Ench	0.057	0.943	0.999	0.989	724	0.782
Batt	0.053	0.947	0.999	0.988	904	1.011
Sola	0.038	0.962	0.999	0.423	1096	1
PoSy	0	1	1	0.002	-	-

The exhaustive analysis of these results is outside the scope of this paper. The mean reliability of the seabot over the entire duration of the simulation is 0.932 with 0.9261 and 0.9379 as the 5% and 95% confidence bounds respectively. The filtering systems are the least reliable components of the system; they are less reliable than the entrance chamber. The ratio of the contributions of each of the filtering systems to that of the entrance chamber leading to the seabot failure is 1:118. However, the entrance chamber fails less frequently with an MTTF of 724 days and takes a shorter time to repair with an MTTR 0.782. Meaning, even though the entrance chamber is more reliable than the filtering systems, it is a single point of failure and contributes more to the system failure so it needs to be improved if a system improvement is desired.

5 Conclusion

Two techniques have been proposed for the evaluation of temporal fault trees using simulation approaches. In the first approach, an algorithm (limited to non-repairable events) has been developed for evaluating all MCSQs in the temporal fault tree. The second approach, using Goldsim, allows for the modelling of dynamic systems featuring some temporal behaviours. This technique applies to almost any system with repairable, replaceable, maintenance, preventive-maintenance, failure and other system operating environment data.

An autonomous underwater vehicle system case study has been analyzed with the proposed techniques and a comparative analysis of both only failure data and all system data (using Goldsim) techniques has been discussed. As would be expected, the proposed technique featuring repairs and maintenance is more reliable than the technique to considers only failure data. Future works will focus on applying the proposed techniques to real-world systems; this could be done by aggregating and synthesizing data from different sourcing using various fourth industrial revolution technologies such as Big Data, the Internet of Things and cloud computing.

References

1. Vesely, W. E. *et al*: *Fault Tree Handbook with Aerospace Applications*. (NASA Office of Safety and Mission Assurance), 2002.
2. Walker, M: Pandora: A Logic for the Qualitative Analysis of Temporal Fault Trees. (University of Hull, 2009).
3. Dugan, J. B., Bavuso, S. J. & Boyd, M: Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems. *IEEE Trans. Reliab.* 41, 363–377 (1992).
4. Tang, Z. & Dugan, J. B: Minimal Cut Set/Sequence Generation for Dynamic Fault Trees. *Annu. Symp. Reliab. Maint. 2004 - RAMS* 207–213 (2004).
5. Merle, G., Roussel, J. & Lesage, J: Improving the Efficiency of Dynamic Fault Tree Analysis by Considering Gate FDEP as Static. *Reliab. Risk …* 1–7 (2010).
6. Palshikar, G. K: Temporal Fault Trees. *Inf. Softw. Technol.* 44, 137–150 (2002).
7. Kabir, S: An overview of fault tree analysis and its application in model-based dependability analysis. *Expert Systems with Applications* 77, 114–135 (2017).
8. Edifor, E., Walker, M. & Gordon, N: Quantification of priority-OR gates in temporal fault trees. in *Lecture Notes in Computer Science* 7612 LNCS, 99–110 (2012).
9. Edifor, E., Walker, M. & Gordon, N: Quantification of simultaneous-AND gates in temporal Fault Trees. in *Advances in Intelligent Systems and Computing* 224, 141–151 (2013).
10. Fussell, J. B., Aber, E. F. & Rahl, R. G: On the Quantitative Analysis of Priority-AND Failure Logic. *IEEE Trans. Reliab.* R-25, 324–326 (1976).
11. Edifor, E. E: Quantitative Analysis of Dynamic Safety-Critical Systems Using Temporal Fault Trees. (University of Hull, 2014).
12. Herrera, F. & Sander, I: Combining Analytical and Simulation-based Design Space Exploration for Time-Critical Systems. in *IEEE Specification & Design Languages* 1–8 (2013).
13. Esary, D. & Proschan, F: Coherent Structures with Non-Identical Components. *Technometrics* 5, 191–209 (1963).
14. Hong, Y., Zhang, M. & Meeker, W. Q: Big data and reliability applications: The complexity dimension. *J. Qual. Technol.* 50, 135–149 (2018).
15. Goldsim: Goldsim. *A Dynamic Simulation Approach to Reliability Modeling and Risk Assessment Using GoldSim* (2020). Available at: https://media.goldsim.com/Documents/WhitePapers/GoldSim_Reliability_and_PRA.pdf. (Accessed: 23rd January 2021)