A Conceptual Framework to Incorporate Complex Basic Events in HiP-HOPS

Sohag Kabir, Koorosh Aslansefat, Ioannis Sorokos, Yiannis Papadopoulos, and Youcef Gheraibia

Department of Computer Science and Technology, University of Hull, Hull, UK
{s.kabir, k.aslansefat-2018, i.sorokos, y.gheraibia, y.i.papadopoulos}@hull.ac.uk

Abstract. Reliability evaluation for ensuring the uninterrupted system operation is an integral part of dependable system development. Model-based safety analysis (MBSA) techniques such as Hierarchically Performed Hazard Origin and Propagation Studies (HiP-HOPS) have made the reliability analysis process less expensive in terms of effort and time required. HiP-HOPS uses an analytical modelling approach for Fault tree analysis to automate the reliability analysis process, where each system component is associated with its failure rate or failure probability. However, such non-state-space analysis models are not capable of modelling more complex failure behaviour of component like failure/repair dependencies, e.g., spares, shared repair, imperfect coverage, etc. State-space based paradigms like Markov chain can model complex failure behaviour, but their use can lead to state-space explosion, thus undermining the overall analysis capacity. Therefore, to maintain the benefits of MBSA while not compromising on modelling capability, in this paper, we propose a conceptual framework to incorporate complex basic events in HiP-HOPS. The idea is demonstrated via an illustrative example.

Keywords: Fault Tree · Markov Process · Model-based safety analysis · HiP-HOPS · Reliability · Real time analysis

1 Introduction

By performing safety and reliability analysis of systems, it is possible to know how they can fail and what is the probability that they will operate without any failure for a specific time period. Fault Tree Analysis (FTA) [32] is a widely used top-down deductive approach for reliability analysis. Using FTA, it is possible to understand the potential causes of system failure and the probability of that failure.

Although FTA is primarily performed manually by analysts, the emergence of model-based safety analysis (MBSA) [26] has greatly reduced required manual effort by proposing ways for automating FTA. Among different available MBSA approaches, Hierarchically Performed Hazard Origin and Propagation Studies (HiP-HOPS)[22] offers multiple state-of-the-art functionalities, supported by a

tool. HiP-HOPS can automatically generate fault trees and Failure Mode and Effects Analyses (FMEAs) from system models. HiP-HOPS also supports multi-objective optimisation of system models and semi-automatic allocation of safety requirements to system components in the form of Safety Integrity Levels (SILs). These features automate some of the processes for the ASIL allocation specified in ISO 26262 [24].

Safety analysis through traditional FTA cannot model dynamic behaviour of systems by taking into account the complex interactions and dependencies between system components. To model complex dependencies and dynamic system behaviour, classical static fault trees are extended as dynamic fault trees by introducing dynamic gates. However, this approach is rarely used in industry due to the complexity associated with the analysis of such models and lack of training [35]. Moreover, there is lack of support for model-based analysis of DFTs.

Generally, during reliability analysis via a static or dynamic fault tree, system components are assumed to have various states of nominal operation or of failure. Component failure behaviours are defined accordingly either as probability of failure or failure rate or distribution of time of failure. However, components in practical systems can operate in multiple states and can have complex failure behaviour. For instance, in [17, 31, 25], Trivedi *et al.* illustrated the concepts of reliability and availability analysis of systems by considering the complex failure behaviour of components. In doing so, they utilised the modelling capability of Markov chains and used exponentially distributed data.

In [4, 1], Markov chain-based complex behaviour modelling of system components were considered and basic events of fault trees were proposed to be substituted by such state-space models. Zixian et al. [36] combined a Markov model and fault tree for the analysis of time-independent and dependent failure behaviour of components in medical industry. In their approach, failure of all medical equipment was modelled using a single type of Markov model and the human error was modelled using another type of Markov model. This implies that all physical components are assumed to have same failure behaviour, however, this may not be true in all practical applications. Recently, Zeller and Montrone [35] proposed a component-oriented concept of Markov chains to incorporate the Markov chain-based model of basic events in Component Fault Trees. At the same time, Nguyen *et al.* [20] used stochastic reward nets instead of Markov chains to model the complex behavior of basic events in fault trees. Note that none of the above concepts was proposed in the context of MBSA and they are only applicable to exponentially distributed data. Moreover, all the approaches focus only on design time (offline) analysis, hence there is no provision for incorporating runtime evidence about components' states in the analysis to update the belief about system reliability and/or availability.

Currently, reliability analysis through HiP-HOPS lacks an appropriate component concept that would allow to model the complex failure behaviour of a component as a Basic Event (BE) in the form of a separate subgraph. HiP-HOPS neither offers the modelling of multi-state components nor can the annotation of

the BEs incorporate complex degradation behaviour and repair actions. Therefore, considering the advantages provided by HiP-HOPS, and MBSA in general, in this paper we propose a conceptual framework to incorporate the concept of complex basic event in the HiP-HOPS. Note that our goal is not to model the state-space-based failure behaviour of a system due to complex interactions between its components. Instead, we aim to model the behaviour of some selected components of the system using state-space-based methods. As part of reliability analysis using HiP-HOPS, in the proposed framework, firstly we identify the components that have complex failure behaviour. Subsequently, we propose to model the failure behaviour of the basic events associated with such components using a Semi-Markov Process (SMP). Then, offline reliability analysis is performed based on the parameters available at design time. The framework can also perform real-time analysis during system operation by monitoring and providing evidence in the state-space models of the basic events. In summary, our work is different from other existing works and is advantageous because:

- It considers component-level complex behaviour modelling in the context of model-based safety analysis.
- Via the use of SMP, the proposed framework could analyse systems with both exponentially and non-exponentially distributed data.
- In addition to performing design time offline analysis, the approach has the capability to perform runtime analysis.

2 Background

2.1 Reliability analysis in HiP-HOPS

Hierarchically Performed Hazard Origin and Propagation Studies (HiP-HOPS) is a state-of-the-art software reliability analysis method. The tool and the surrounding methodology [21] have evolved as a body of work over the past decade, incorporating further techniques for design [23] and dependability requirement optimization [27, 28], temporal fault tree analysis [13], integration with the EAST-ADL [5] and AADL [19] modeling languages, uncertainty analysis [14] and more.

At the core of the HiP-HOPS approach is its ability to perform semi-automatic Fault Tree Analysis (FTA). FTA is a deductive, top-down analysis approach, applied extensively across numerous industries involved in dependability-critical systems development. See [18] for an extensive but older review and [11] for a more recent one. In FTA, system-level 'failures' (undesirable events, depending on the context) are modeled as the root of tree structures, whose leaf nodes represent basic events which cannot be further analyzed in the context of the analysis. Between the root (aka 'top-event') and the leaves of the tree, logic gates link and propagate the logic that governs the tree. Traditionally, gates used have been Boolean AND and OR gates, however more advanced options have also been explored in the literature e.g. temporal [33] and fuzzy logic events and operators, first seen in [29].

In HiP-HOPS, the user begins by annotating a model of the system architecture with mostly local (per system element) failure behaviour information. This information describes any basic events associated with the given element and the logic with which they are propagated from the element's inputs to its outputs. As the information is limited to the boundaries of this black-box view of each element, users do not need to break their modeling workflow to cross-reference potentially complex relationships with other, distant elements in the system architecture hierarchy. In addition to the qualitative failure logic, HiP-HOPS allows to associate failure and repair rates with basic events, which can be used for reliability analysis in later stages. It is important to note that, in HiP-HOPS, all the quantitative information provided is under the assumption that the failure a BE represents can either occur or not. In other words, the state of a BE is binary and can thus only represent up to one failure class per BE.

Once the annotation of a system model is complete, the HiP-HOPS tool can be invoked, automatically synthesizing local fault trees for each system element. The algorithm combines the local fault trees into a merged one, which is then minimized using logical rules to eliminate redundant sub-trees and so forth. Once the resulting minimal fault tree is complete, it can be analysed qualitatively and quantitatively. In the former case, the necessary and sufficient combinations of basic events, known as the minimal cut set, can be determined. Minimal cut sets are useful for directly identifying single points of failure as single-member cut sets as well as other critical combinations of low-level failures that can cause systemic failure. For quantitative analysis, the basic events of the fault tree can be assigned probabilities in various forms, most often failure rates according to some assumed distribution. The logical operators found in the fault tree structure can be used to combine the probabilities of linked basic events. For instance, AND and OR gates would combine, respectively, via multiplication and addition, the probability of basic events occurring, assuming the events are considered independent.

2.2 Complex failure behaviour modelling of a system

In traditional fault-tree-based reliability analysis, systems and their components are usually considered to have two types of states: *working* and *failed*. To facilitate reliability analysis, each of such elements can have their probability of failure or failure rate or distribution of time of failure or steady-state or instantaneous (un)availability defined. At the same time, if a component/system can be repaired then a repair rate is defined. However, modern large-scale complex systems have the capacity to work in different states and have complex repair processes. A component in such a system can work as a primary at a particular point in time, and in another instance the same component can work as a secondary or spare. Moreover, if a component acts as a spare, it can be in different modes of spare such as cold, warm, and hot spares.

A component does not necessarily transition directly from a working state to a failed state, and vice versa. The complete failure of a component may occur following a complex degradation process and recovery from failure may also involve complex repair processes. For instance, a battery, when fully charged,

may be considered as a fully operational component. From this mode, the battery can fail directly. The battery may be discharged to different levels over the course of operation. Consider each of the distinct charged levels such as 75%, 50%, and 25% as a distinct mode of operation. From each of these modes the battery can transition to the failed mode. The battery can also shift from one mode of operation to another mode either through further discharging or by recharging.

Such multi-modal operation capability of systems and their components gives rise to different dynamic failure characteristics like priorities among events and functionally dependent events. However, using the classical fault tree approach, it is not possible to model such complex dynamic behaviour. Expressiveness of traditional fault trees has been enhanced through different extensions of fault trees such as Dynamic Fault Trees [8] and State/Event Fault Trees [16]. These approaches are mostly useful in modelling dependencies and priorities among events. For a quantitative analysis, these models are usually transformed to state-space-based models like Markov chains or Semi Markov Process and [3] Petri Nets [15]. This leads to a state-space explosion problem, which limits their applicability to large-scale industrial systems. In addition to this, Markov models, the most widely used approach for dynamic reliability analysis, are applicable only to systems consisting of components with exponentially distributed lifetime.

2.3 Reliability modelling using Semi Markov Process

Semi-Markov Process (SMP) has been widely used in reliability evaluation of industrial systems [30]. The SMP has the ability to consider non-exponential probability distributions that can be counted as an advantage in comparison to other state-space methods. In this paper, three SMP parameters of $(p, P, F(t))$ are considered, where: p is the initial probability distribution vector, P is conditional transition probabilities matrix and $F(t)$ describes matrix of distribution functions of sojourn times in state i^{th}, when j^{th} state is next.

Considering X_i, $\forall_i = 0, 1, 2, \ldots$ as random variables, the time-homogeneous SMP X is determined by a vector of initial state probabilities $p(0) = \lceil P\{X_0 = i\}\rceil = \lceil 1, 0, \ldots, 0\rceil$ and the matrix of conditional transition probability $P(t) = \lceil P_{ij}(t)\rceil$ is computed by Eq. (1).

$$P_{ij}(t) = P\{X(t) = j | X(0) = i\} \quad i, j \in States \tag{1}$$

The $P_{ij}(t)$ matrix provided in previous equation can be satisfied by Kolmogorov-Feller's equation in Eq. (2) [34].

$$P_{ij}(t) = \delta_{ij}[1 - G_i(t)] + \sum_{k \in S} \int_0^t P_{kj}(t - x) dQ_{ik}(x) \tag{2}$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise, G_i is the distribution of the sojourn time in state i described by Eq. (3) [6], and $Q_{ij}(t)$ describes the kernel matrix by Eq. (4) [9].

$$G_i(t) = P\{S_i \leq t \mid X_0 = i\} = \sum_{j=1}^{i} Q_{ij}(t) \tag{3}$$

where $S_i, i = 0, 1, 2, \ldots$ is the state of the system at time t.

$$Q_{ij}(t) = P\{X_1 = j, S_i \leq t \mid X_0 = i\} \tag{4}$$

The solution of Eq. (2) can be found by applying the Laplace Stieltjes Transformation (LST) in Eq. (5) [10]. Note that for non-exponential failure distributions such as Weibull and Gamma, some approximation algorithm is needed (refer to [7, 34, 2]).

$$\tilde{p}_{ij}(s) = \delta_{ij}[1 - \tilde{g}_i(s)] + \sum_{k \in S} \tilde{q}_{ik}(s)\tilde{p}_{kj}(s) \tag{5}$$

Eq. (5) in matrix form can be rewritten as follows:

$$\tilde{p}(s) = \lceil 1 - \tilde{g}(s) \rceil + \tilde{q}(s)\tilde{p}(s) \tag{6}$$

Hence, it can be rewritten as Eq. (7).

$$\tilde{p}(s) = \lceil 1 - \tilde{q}(s) \rceil^{-1} \tilde{g}(s)) \tag{7}$$

In the above equation (7), the inverse of $1 - \tilde{q}(s)$ can be replaced by the summation of powers of $\tilde{q}(s)$. Eq. (7) can then be rewritten as Eq. (8). This equation is useful for singular kernel matrices.

$$\tilde{p}(s) = \Big(\sum_{n=0}^{\infty} \tilde{q}(s)^n \Big) \tilde{g}(s) \tag{8}$$

Having solved Eq. (8) with taking the inverse LST of $\tilde{p}(s)$, the unconditional state probabilities in time domain are determined as follows:

$$P(t) = P(0)P(t) \tag{9}$$

Finally, the reliability of a system can be achieved through summation of probability of operational state in the SMP.

3 Proposed Approach

Fig. 1 shows the framework of the proposed approach. As seen in the framework, the Annotation, Synthesis, and Analysis phases of the HiP-HOPS methodology are extended with new activities. Additionally, a new phase for real-time evaluation is added in the new approach. In this new approach, the annotation phase of the HiP-HOPS is extended by introducing an additional check. If, for quantitative analysis, the failure rate and repair rate are not sufficient to model a BE, then a complex BE must be defined instead. In that case, a suitable state-space based model is selected to model its behaviour. As a result of this, the logical annotations of the components do not change.

Consider the architecture of a system in Fig. 2. Each component is annotated with its failure behaviour according in HiP-HOPS' format. For instance, the

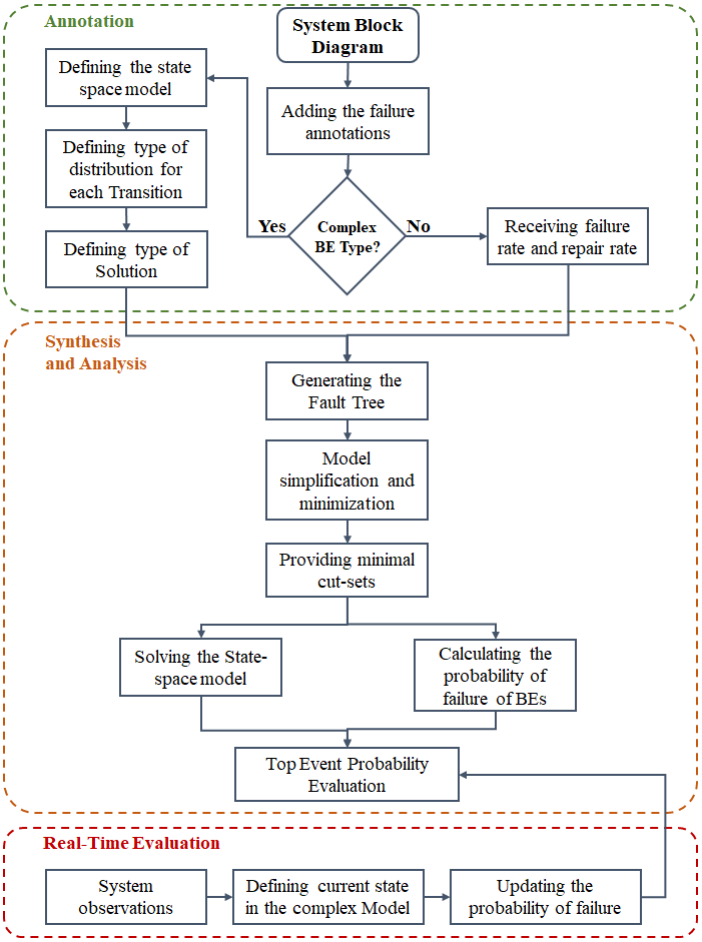


Fig. 1. Framework of the proposed approach

annotation of component C3 is: `O-C3 = O-C1 OR I-C3`. This means that the component C3 will fail to produce any output ('O' stands for the 'Omission' failure class) if there is no output from C1 (`O-C1`) or if there is an internal ('I') failure of C3 (`I-C3`). As mentioned earlier, the annotation would remain the same in the current approach. For quantitative analysis, HiP-HOPS uses λ_{C1} and μ_{C1} as the failure and repair rates of component C1. However, if the component C1 has a complex failure behaviour then this kind of data cannot be used. For this reason, the proposed approach would use a state-space-based model to represent the failure behaviour of C1. For instance, Fig. 2 shows a semi-Markov process based complex failure behaviour of component C1's.

The synthesis and analysis steps would produce both qualitative and quantitative results. For qualitative results, following the procedure described in section 2.1, fault trees would be generated first and minimal cut sets would be generated

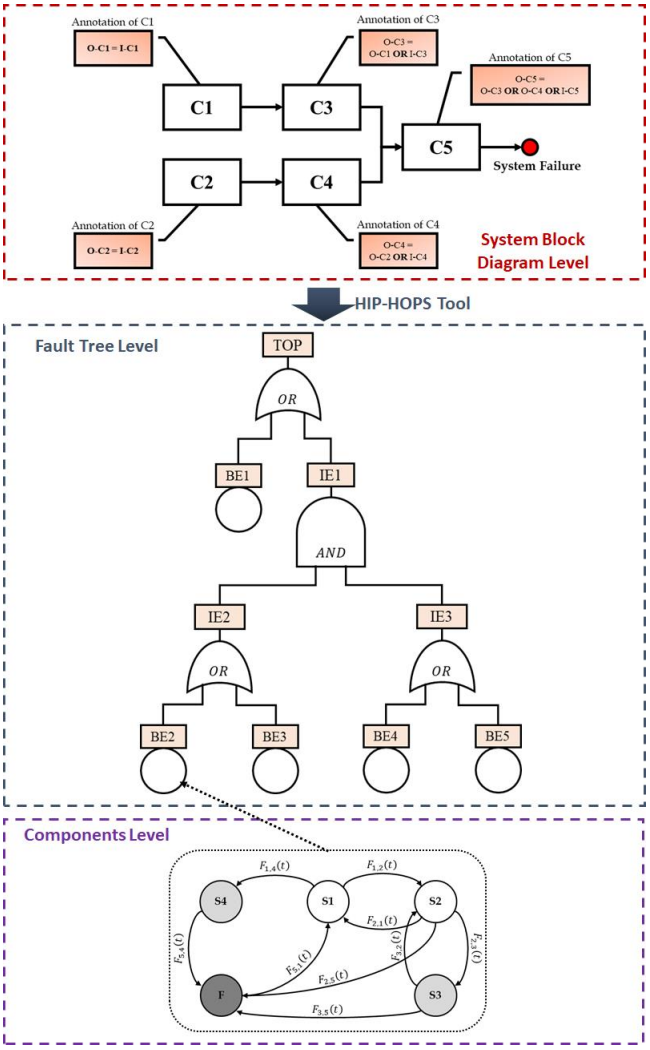


Fig. 2. An Example system architecture with failure annotations

next. However, as we currently have a Semi-Markov (SM) model for the complex BEs, we cannot perform the quantitative analysis as described in section 2.1 until we solve the SMP models. Therefore, the first step of quantitative analysis in the proposed approach is to solve the SMP-based models of the complex BEs to obtain the failure probability of the BEs. The process of solving SMP-based models are discussed in section 2.3. Afterwards this data will be used to obtain the probability of the top event of the fault tree. Additional analysis such as criticality analysis of BEs can also be performed.

Note that all the above analyses are performed at design time. However, the proposed framework also introduces real-time analysis via HiP-HOPS. For

real-time analysis, fault trees created during design time and the observations about system operation are used to update the knowledge about the system failure probability and criticality of BEs. As the complex BEs have state-space models of failure behaviour, the basic idea of this phase is to utilise the real-time operational knowledge of the system to place observations in the state-space models. Thus, during run-time, the approach can identify in which state a BE is in, which was not possible to determine at design time. Based on this new knowledge, the probability of each complex BE will be updated, which will eventually be propagated to update the belief about system failure probability.

Consider the failure behaviour model of component C1 in Fig. 2. At design time, it is not possible for analysts to know in which state the component will work during operation. As a result, the design time analysis will calculate the probability of the BE associated with this component by solving the state-space of the model of Fig. 2. However, at run-time, based on the observation of the system operation, the analysts may find that the component is working in state **S4**. Due to this new knowledge, a modified state-space would be solved for the model to obtain a new failure probability of the BE.

4 Illustrative Example

To illustrate the idea of safety analysis of systems with complex BEs via HiP-HOPS, we use a simplified version of the oxygen sensing and generation unit of an Automatic Pond Oxygen Management System first presented in [12], and shown in Fig. 3. The role of this system is to continuously sense the oxygen level of a pond and if the oxygen level falls below certain level then the system will automatically generate oxygen. The system contains two oxygen level sensing blocks, A and B. Each of these blocks contains a battery and an oxygen sensor. The battery keeps the sensor alive and the sensor senses the pond's oxygen level. Readings from both blocks are fed to the Decision Making (DM) block. Based on these readings, the DM can decide whether to generate oxygen or not. Note that although both block A and B work simultaneously, input from at least one of them is necessary to make a decision by the DM. When the DM finds that it is necessary to generate oxygen, it uses the oxygen generator (OG) unit to generate oxygen. During operation the OG draws power from the power supply.

For a model-based analysis of this system using HiP-HOPS, the architecture of the system was annotated by taking into account the failure behaviour of each of the system components. A fault tree was automatically generated based on this annotated architecture and shown in Fig. 4. Table 1 shows the ID and description of the basic and intermediate events of the fault tree. In this study, basic events 1, 4, and 6 were considered as complex basic events. The SMP-based failure behaviour models of these BEs are shown in Fig. 5. Parameters associated with these models and failure rates of other BEs are shown in Table 2.

Without loss of generality we evaluate the reliability of the system of Fig. 3 for a mission time of 500 hours. To illustrate the effectiveness of proposed framework, we have created some scenarios as shown in Table 3. As can be seen,

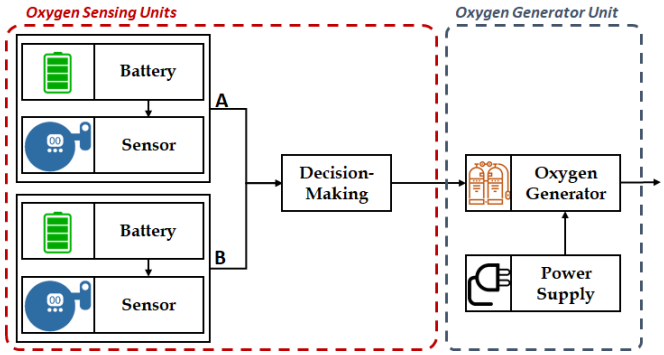


Fig. 3. An Example System

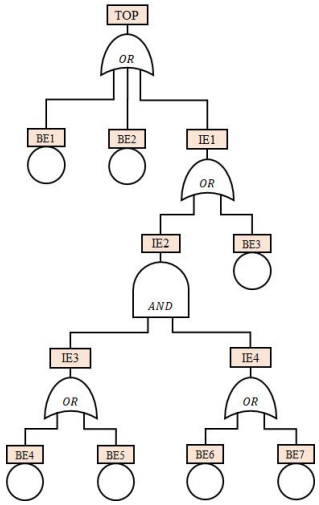


Fig. 4. Fault tree of the system in Fig. 3

at time interval $[0, 100]$ no observation has been provided for the states of the system components. As a result, analysis performed within this interval is like an offline analysis. At time interval $[101, 200]$, it is observed that the battery system is in state D2 and at time interval [201, 500], the power system has been observed to be in state S4. Fig. 6 shows the reliability of the battery and power systems with and without observation. The changes in reliability of these systems due to real time monitoring is clearly reflected in the figure. For instance, for battery system and the power system, the reliability declined steadily until 100 hours and 200 hours, respectively. After 100 hours and 200 hours, respective reliability per each system drops sharply and then continue to decline steadily again. That means because of our real time observation of the battery and power system states, our knowledge about the reliability of these systems is updated accordingly, which is not possible with design time analysis. Fig. 7 shows the

Table 1. ID and description of the basic and intermediate events of the fault tree

Event ID	Event Description
TE	No oxygen generated when required
IE1	No outputs from decision making block
IE2	No output from oxygen level sensing blocks
IE3	No output from oxygen level sensing block A
IE4	No output from oxygen level sensing block B
BE1	Power supply failure
BE2	Internal failure of oxygen generator
BE3	Internal failure of decision making block
BE4	Battery in oxygen level sensing block A failed
BE5	Sensor in oxygen level sensing block A failed
BE6	Battery in oxygen level sensing block B failed
BE7	Sensor in oxygen level sensing block B failed

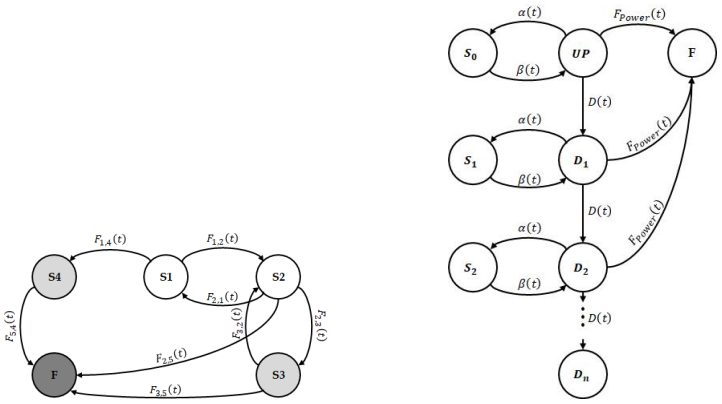
Table 2. Parameters for the BEs and their SMP-based models in Fig. 5

BEs	Parameters	BEs	Parameters
BE1	$F_{1,2}(t) = 1 - e^{-0.00065t}$	BE2	$\lambda = 0.00023$
	$F_{2,1}(t) = 1 - e^{-0.00073t}$	BE3	$\lambda = 0.00023$
	$F_{2,3}(t) = 1 - e^{-0.00633t}$	BE4, BE6	$\alpha(t) = 1 - e^{-0.00078t}$
	$F_{2,5}(t) = 1 - e^{-0.00044t}$		$\beta(t) = 1 - e^{-0.00082t}$
	$F_{3,2}(t) = 1 - e^{-0.00075t}$		$D(t) = 1 - e^{-0.00064t}$
	$F_{3,5}(t) = 1 - e^{-0.00044t}$		$F_{Power}(t) = 1 - e^{-0.00285t}$
	$F_{1,4}(t) = 1 - e^{-0.00860t}$	BE5	$\lambda = 0.00015$
	$F_{4,5}(t) = 1 - e^{-0.00088t}$	BE7	$\lambda = 0.00091$

Table 3. Experimental settings

Mission time	Real time observation
$t = [0, 100]$	No Observation
$t = [101, 200]$	State D2 in the SMP of Battery has been observed
$t = [201, 500]$	State S4 in the SMP of Power System has been observed

reliability of the whole system for 500 hours mission time. The effects of the observing the operating states of battery and power system on the reliability of the whole system is clearly visible in the figure. This real-time analysis feature not only helps us to update our belief about the system reliability, but also allows us to perform a meaningful analysis by taking into account the real operational status of the system.



(a) Failure behaviour of BE1

(b) Failure behaviour of BE4 and BE6

Fig. 5. State-based behaviour of BEs 1, 4, and 6

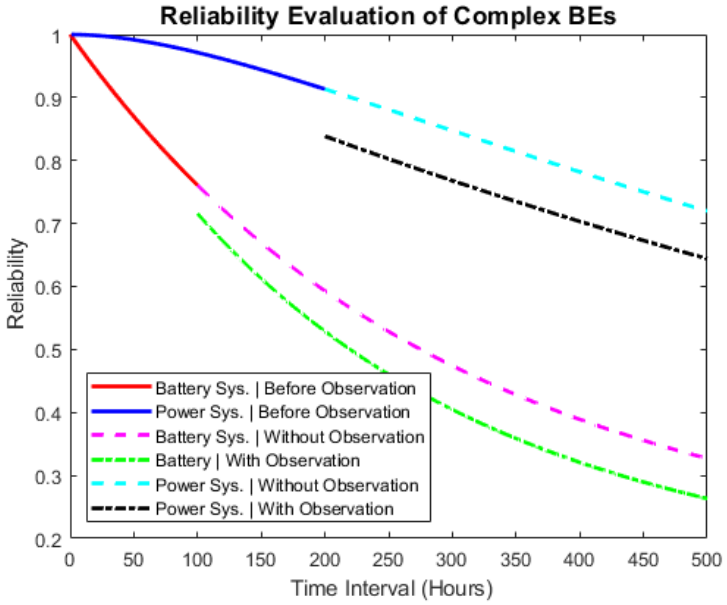


Fig. 6. Reliability of Battery and Power systems with and without observation

5 Conclusion

In this paper, we have presented a framework for incorporating the concept of SMP-based complex behaviour modelling of system components in HiP-HOPS. The framework retained all the functionality provided by HiP-HOPS while offering a simple way for modelling the failure behaviour of complex systems. Thus,

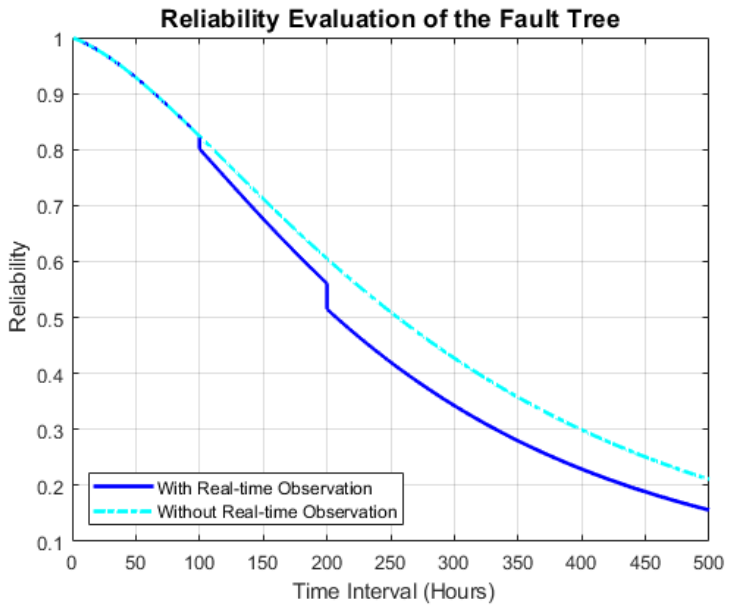


Fig. 7. Reliability of the whole system with and without observation

it enables fast, modular and compositional MBSA of such complex systems. The SMP-based basic event modelling supports distribution-independent analysis of system. Moreover, the proposed framework enables us to perform evidence-based runtime systems analysis.

The current approach focuses solely on the quantitative analysis part of the HiP-HOPS approach. In the future, we plan to explore the qualitative analysis aspects by considering the complex behaviour of basic events. Currently, the effectiveness of the approach is evaluated via a small illustrative example. In future work, more detailed evaluation using large-scale industrial systems will be pursued to illustrate the advantage of our proposed framework in MBSA of complex systems.

Acknowledgements

This work was supported by the DEIS H2020 Project under Grant 732242.

References

1. Adler, R., Forster, M., Trapp, M.: Determining configuration probabilities of safety-critical adaptive systems. In: 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07). vol. 2, pp. 548–555. IEEE (2007)

2. Aslansefat, K.: A novel approach for reliability and safety evaluation of control systems with dynamic fault tree. MSc. Thesis, Abbaspur Campus, Shahid Beheshti University (2014)
3. Aslansefat, K., Latif-Shabgahi, G.: A hierarchical approach for dynamic fault trees solution through semi-markov process. IEEE Transactions on Reliability pp. 1–18 (2019). https://doi.org/10.1109/TR.2019.2923893
4. Bouissou, M., Bon, J.L.: A new formalism that combines advantages of fault-trees and markov models: Boolean logic driven markov processes. Reliability Engineering & System Safety **82**(2), 149–163 (2003)
5. Chen, D., Mahmud, N., Walker, M., Feng, L., Lönn, H., Papadopoulos, Y.: Systems modeling with EAST-ADL for fault tree analysis through HiP-HOPS. IFAC Proceedings Volumes **46**(22), 91–96 (2013)
6. Cochran, J.: Wiley Encyclopedia of Operations Research and Management Science. John Wiley and Sons Ltd (2010)
7. Distefano, S., Longo, F., Trivedi, K.S.: Investigating dynamic reliability and availability through state–space models. Computers & Mathematics with Applications **64**(12), 3701–3716 (2012)
8. Dugan, J.B., Bavuso, S., Boyd, M.: Dynamic fault-tree models for fault-tolerant computer systems. IEEE Transactions on Reliability **41**(3), 363–377 (1992)
9. Fricks, R., Telek, M., Puliafito, A., Trivedi, K.S.: Markov renewal theory applied to performability evaluation. Tech. rep., North Carolina State University. Center for Advanced Computing and Communication (1996)
10. Insua, D., Ruggeri, F., Wiper, M.: Bayesian analysis of stochastic process models, vol. 978. John Wiley & Sons (2012)
11. Kabir, S.: An overview of fault tree analysis and its application in model based dependability analysis. Expert Systems with Applications **77**, 114–135 (2017)
12. Kabir, S., Azad, T., Walker, M., Gheraibia, Y.: Reliability analysis of automated pond oxygen management system. In: 18th International Conference on Computer and Information Technology (ICCIT). pp. 144–149. IEEE (2015)
13. Kabir, S., Walker, M., Papadopoulos, Y.: Dynamic system safety analysis in HiP-HOPS with Petri nets and Bayesian networks. Safety science **105**, 55–70 (2018)
14. Kabir, S., Walker, M., Papadopoulos, Y., Rüde, E., Securius, P.: Fuzzy temporal fault tree analysis of dynamic systems. International Journal of Approximate Reasoning **77**, 20–37 (2016)
15. Kabir, S., Yazdi, M., Aizpurua, J.I., Papadopoulos, Y.: Uncertainty-aware dynamic reliability analysis framework for complex systems. IEEE Access **6**(1), 29499 – 29515 (2018)
16. Kaiser, B., Gramlich, C., Förster, M.: State/event fault treesa safety analysis model for software-controlled systems. Reliability Engineering & System Safety **92**(11), 1521–1537 (2007)
17. Kim, D.S., Ghosh, R., Trivedi, K.S.: A Hierarchical Model for Reliability Analysis of Sensor Networks. In: 2010 IEEE 16th Pacific Rim International Symposium on Dependable Computing. pp. 247–248 (Dec 2010)
18. Lee, W.S., Grosh, D.L., Tillman, F.A., Lie, C.H.: Fault tree analysis, methods, and applications a review. IEEE transactions on reliability **34**(3), 194–203 (1985)
19. Mian, Z., Bottaci, L., Papadopoulos, Y., Biehl, M.: System dependability modelling and analysis using aadl and hip-hops. IFAC Proceedings Volumes **45**(6), 1647–1652 (2012)
20. Nguyen, T.A., Min, D., Choi, E., Tran, T.D.: Reliability and availability evaluation for cloud data center networks using hierarchical models. IEEE Access **7**, 9273–9313 (2019)

21. Papadopoulos, Y., Maruhn, M.: Model-based synthesis of fault trees from matlab-simulink models. In: 2001 International Conference on Dependable Systems and Networks. pp. 77–82. IEEE (2001)
22. Papadopoulos, Y., McDermid, J.A.: Hierarchically performed hazard origin and propagation studies. In: International Conference on Computer Safety, Reliability, and Security. pp. 139–152. Springer (1999)
23. Papadopoulos, Y., Walker, M., Parker, D., Rüde, E., Hamann, R., Uhlig, A., Grätz, U., Lien, R.: Engineering failure analysis and design optimisation with hip-hops. Engineering Failure Analysis **18**(2), 590–608 (2011)
24. Papadopoulos, Y., Walker, M., Parker, D., Sharvia, S., Bottaci, L., Kabir, S., Azevedo, L., Sorokos, I.: A synthesis of logic and bio-inspired techniques in the design of dependable systems. Annual Reviews in Control **41**, 170–182 (2016)
25. Ramezani, Z., Latif-Shabgahi, G.R., Khajeie, P., Aslansefat, K.: Hierarchical steady-state availability evaluation of dynamic fault trees through equal markov model. In: 2016 24th Iranian Conference on Electrical Engineering (ICEE). pp. 1848–1854. IEEE (2016)
26. Sharvia, S., Kabir, S., Walker, M., Papadopoulos, Y.: Model-based dependability analysis: state-of-the-art, challenges, and future outlook. In: Software Quality Assurance, pp. 251–278. Elsevier (2016)
27. da Silva Azevedo, L., Parker, D., Walker, M., Papadopoulos, Y., Araujo, R.E.: Assisted assignment of automotive safety requirements. IEEE software **31**(1), 62–68 (2014)
28. Sorokos, I., Papadopoulos, Y., Azevedo, L., Parker, D., Walker, M.: Automating allocation of development assurance levels: an extension to hip-hops. IFAC-PapersOnLine **48**(7), 9–14 (2015)
29. Tanaka, H., Fan, L., Lai, F., Toguchi, K.: Fault-tree analysis by fuzzy probability. IEEE Transactions on reliability **32**(5), 453–457 (1983)
30. Trivedi, K.S., Bobbio, A.: Reliability and availability engineering: modeling, analysis, and applications. Cambridge University Press (2017)
31. Trivedi, K.S., Kim, D.S., Ghosh, R.: System availability assessment using stochastic models. Applied Stochastic Models in Business and Industry **29**(2), 94–109 (2013)
32. Vesely, W., Dugan, J., Fragola, J., Minarick, Railsback, J.: Fault Tree Handbook with Aerospace Applications. Tech. rep., NASA office of safety and mission assurance, Washington, DC (2002)
33. Walker, M., Papadopoulos, Y.: Qualitative temporal analysis: Towards a full implementation of the fault tree handbook. Control Engineering Practice **17**(10), 1115–1125 (2009)
34. Zajac, M., Kierzkowski, A.: Attempts at calculating chosen contributors with regard to the semi-markov process and the weibull function distribution. Journal of Polish Safety and Reliability Association **2** (2011)
35. Zeller, M., Montrone, F.: Combination of component fault trees and markov chains to analyze complex, software-controlled systems. In: 2018 3rd International Conference on System Reliability and Safety (ICSRS). pp. 13–20. IEEE (2019)
36. Zixian, L., Xin, N., Yiliu, L., Qinglu, S., Yukun, W.: Gastric esophageal surgery risk analysis with a fault tree and markov integrated model. Reliability Engineering & System Safety **96**(12), 1591–1600 (2011)