2016

Temporal Logic for System Safety Properties and Hazard Monitoring

Francesca M. Favaro
J. H. Saleh

TEMPORAL LOGIC FOR SYSTEM SAFETY PROPERTIES AND HAZARD MONITORING

Francesca M. Favarò[a], Joseph H. Saleh[b]

[a] *Department of Aviation and Technology, San Jose State University, San Jose CA*
[b] *School of Aerospace Engineering, Georgia Institute of Technology, Atlanta GA*

In recent years, there has been a growing interest in the use of temporal logic (TL) in a variety of technical areas, such as robotics and safety-critical computational system. TL provides a formal language for the verification of requirements and for specification logic, to ensure the desired performance and behavior of the system. In this work we propose the application of temporal logic for risk analysis and system safety and in support of accident prevention and risk mitigation. We first provide an introduction to temporal logic and the use of temporal operators. We then examine the types of properties that can be expressed in TL and provide a set of four safety principles, formulated at a high-level of abstraction, based on the notions of accident sequence and hazard level/escalation. These safety properties, once expressed in TL, can be monitored during the design and operation of systems for compliance and be verified on-line and off-line. The verification of whether the system satisfies or violates the TL safety properties along with the monitoring of emerging hazards provide an important feedback for designers and operators to recognize the need for and trigger safety interventions. The present work augments the traditional perspective in risk analysis and its reliance on (conditional) probabilities as the basic modeling ingredient with the notion of temporal contingency, a novel dimension by which hazards are prioritized and ranked based on the temporal vicinity of their associated accident(s) to being released. This approach offers novel capabilities, complementary to PRA, and rich possibilities for further contributions toward accident prevention and improved risk management.

I. INTRODUCTION

In recent years, there has been a growing interest in the use of temporal logic (TL) in a variety of technical areas, such as robotics and safety-critical computational system. TL provides a formal language for the verification of requirements and for specification logic, to ensure the desired performance and behavior of the overall system. An increasing number of applications have adopted it, including for example the expression of specifications for automated motion planning problems for a variety of vehicles such as ground-based robots, UAVs, and drones[1], or the specifications of software program semantics capable of dynamically adapting to changing external conditions[2]. In robotics, temporal logic provides a convenient language for the expression of both usual control specifications (e.g., reachability and stability analyses) as well as more complex time-dependent specifications (e.g., sequencing and obstacle avoidance), to express the behavior expected from the system[3]. Once a specification is provided in TL, checks and controls are implemented to ensure that such behavior is followed.

With the increasing demand of highly automated processes and systems, the reliance on the correct and safe functioning of embedded software components is growing rapidly[4]. While computer science and software engineering heavily rely on the use of temporal logic, risk analysis and system safety speak a different (analytical) language. Probabilistic tools, Boolean logic and propositional calculus are well established in the risk and safety community (e.g., the use of Boolean logic in the gates of a fault tree or the use of predicate logic for probability calculations). By leveraging the TL formalism, a non-traditional choice for the risk analysis and system safety domain, the approach we propose offers novel capabilities, complementary to PRA, and rich possibilities for further contributions toward accident prevention and improved risk management.

The motivation for the introduction of temporal logic to the risk community is twofold. First, temporal logic can serve as a bridge between the risk/safety community and the computer science community. Having a common formal language is likely to generate useful synergies between these two communities, and it can stimulate a useful in-depth dialog between them (beyond the current superficial modeling of software problems in Probabilistic Risk Assessment). Some authors have expressed concerns regarding the "still very much hardware-orientated" character of risk analysis, advocating new models to account for this shift in the nature of processes[5]. With the increased development of software-intensive systems, there is a need to leverage automation to support risk assessment and management; the introduction of temporal logic for risk analysis and system safety can serve a useful purpose and first step towards this aim. Secondly, temporal logic makes use of "time operators" that allow expressing ideas of succession, change, and constancy over time, ideas central to risk analysis and to the notion of accident sequence, and that are implicitly included in most risk analysis tools. Temporal logic allows the explicit expression of these notions, translating the event-based path dependency (implicit in risk analysis) into time-based considerations. This translation augments the current perspective in traditional risk analysis with its reliance on (conditional) probabilities as the basic modeling ingredient with the notion of temporal contingency, a novel dimension by which hazards are prioritized and ranked based on the temporal vicinity of their associated accident(s) to being released.

This work is part of a larger effort whose objective is to explore the use of TL for risk analysis and system safety applications. The two companion articles are (Ref 6, 7). The first article develops the framework for model-based system safety and relates it to temporal logic; the second examines in details applications of temporal logic for accident prevention. In this work, we introduce and analyze the temporal logic formulation of four general safety properties, which are applicable to any engineering system. The definition of the safety properties is intimately related to the development of quantifiable metrics for accident escalation, which we refer to as the (analytic) hazard level. The verification of whether the system satisfies or violates the TL safety properties along with the monitoring of emerging hazards provide an important feedback for designers and operators to recognize the need for, rank, and trigger safety interventions. The verification of satisfaction of a safety property, now seen as a constraint on the admissible behavior of the system, acts as a decision-support tool towards improvement of the system design and its inherent reliability (when executed off-line) and towards improved accident prevention and risk mitigation for real-time response to emerging hazards (when executed on-line).

The remainder of this paper is organized as follows. Section 2 provides a brief introduction to temporal logic, time operators, and the verification process. Section 3 introduces the notion of hazard monitoring, and presents a set of high-level safety principles together with their temporal logic formalization. Section 4 explores some of the capabilities of temporal logic for risk analysis and system safety applications. Section 5 concludes this work.

II. TEMPORAL LOGIC AND ITS USE FOR VERIFICATION PURPOSES

Temporal logic (TL) is an extension of classical logic, which adds temporal modalities to the expression of a formula's truth content (for the historical development of TL see (Ref. 8)). TL adds operators that are related to time to the pool of operators from classical logic[9]. Combined with standard propositional logic, TL provides a formal and precise language in which computational and dynamical properties of systems can be described and analyzed. The possibility to include a temporal dimension in a logical formula makes TL an ideal candidate to overcome some of the time-related limitations of traditional PRA highlighted by several authors[5, 6, 10], and for the specification of key properties of systems whose behavior is time dependent, including software systems.

II.A The Operators of TL

In addition to the operators of classical logic (e.g., Boolean operators "and $\wedge$", "or $\vee$"; the existence operator "$\exists$"; the implication operator "$\rightarrow$"), temporal logic makes use of operators that allow expressing ideas of succession, change, and/or constancy over time[11]. Through the use of those temporal operators, TL allows the specification and the automatic verification of compliance with a broad range of important system properties such as *safety*, *liveness*, and *impartiality*[4].

Table 1 - Temporal operators, based on (Ref. 9)

Operator	Description
□ (f)	f is true in all future instants of time
◊ (f)	f is true at some point in the future
O(f)	f is true in the next instant of time
f U g	f is true until g is true
f R g	f releases g from being true

The basic temporal operators of TL are presented in Table 1 (in this work we only make use of the first three). Additional details on their definitions can be found in (Ref. 4, 9, 12). These basic operators can be extended with annotations allowing the expression of real-time constraints[9]. For instance, the expression "$\Diamond_{>t_i}(f)$" implies that f will be true at some point in the future after t_i. Also, all the operators can be extended to be true for past times (instead of future ones), and are denoted by "blackening" the corresponding symbol (e.g., ■ (f) for always true in the past). Other operators or logical connectives used hereafter are described in Table 2.

Table 2 – Logical Connectives of classical logic

Symbol	Read as
∃	There exists
→	Implies
∧, ∨	And, Or
≜	Is defined as
¬	Not

The underlying nature of time in temporal logic can be either linear or branching. In the linear perspective, for each instant of time there is only one direct successor and one direct predecessor, whereas in the branching one time has a "tree-like structure" where alternative future courses can be considered for each instant of time[4]. In this work we deal with linear temporal logic, which allows a simple perspective for the relative ordering of events (branching temporal logic is left as a fruitful venue for future work). Consider for instance two mutually exclusive events A and B that occur in a particular temporal order: first A and then B. This situation can be expressed by the TL formula "A ∧ O(B)" which is read as "at the present time A is true and in the next instant of time B is true" as represented in Figure 1. The real-time constraints previously mentioned can assist in specifying particular time intervals of interest.

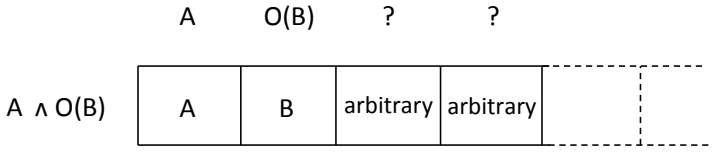

Figure 1 – Representation of "A ∧ O(B)"

TL formulae make use of a specific preference order for the logic operators: unary operators (those that require only one input argument, e.g., "O") bind stronger than the binary ones (those that require two input arguments, e.g., "∧"). The parenthesis in the formula "A ∧ O(B)" can thus be omitted. For more complex cases, parentheses are used to ensure the correct understanding and execution of the formula.

II.B The Verification Process

TL provides an intuitive and mathematically precise notation for expressing properties that relate different system states at different times[4]. In general, a TL formula can be intuitively thought of as providing one of the following: (i) a constraint on possible transitions between system states; (ii) a constraint on the set of states that can be accepted at the next instant of time; (iii) a description of system invariants, which are properties that should remain unchanged for the entire life of the system (e.g., many safety requirements that are expressed in the form "condition A never occurs" are considered invariants, as they describe a condition that should hold for all states of the system at all times)[9].

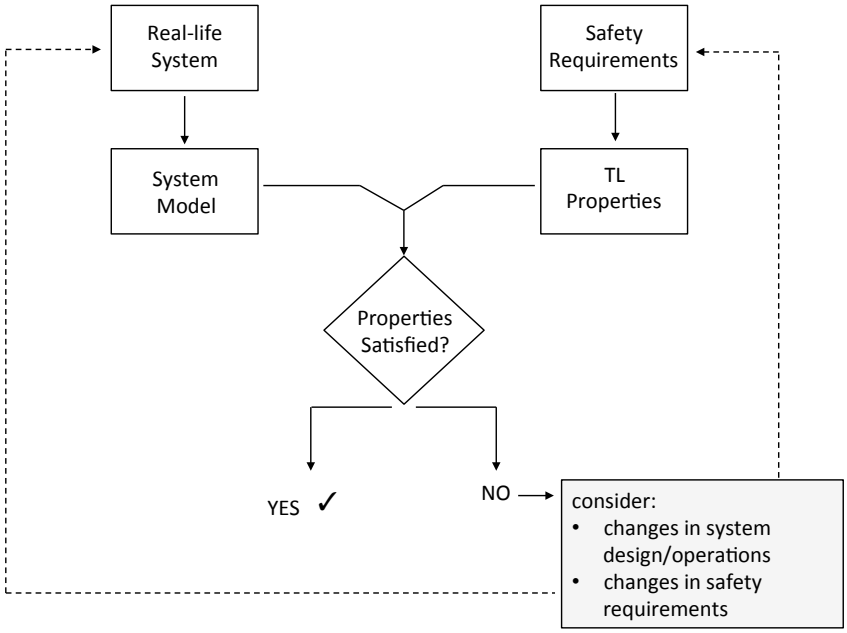

Figure 2 – Schematic representation of the verification process

Two ingredients are needed for the verification process: the first is the translation in TL of a system requirement (for our purposes a safety requirement); the second is a model for the system under consideration. The verification effort aims at checking the system compliance with the specified TL properties. This process can be achieved through direct monitoring of the system behavior or through formal verification techniques that involve mathematical abstraction (for more details see (Ref. 6, 7)). The verification process is schematically represented in Figure 2.

Figure 2 shows that if the compliance check is not satisfied, changes in the system design or in the system operating procedures and possibly in the safety requirements should be considered. The violation of one or more properties provides an important feedback for the operators/designers in both off-line and on-line applications. If the verification/monitoring process is executed off-line (in a simulation environment), it can serve a useful purpose during the design and development stages of the system: violations of specific TL safety properties provide a useful feedback to designers and management to trigger changes in the current system design and layout of operating procedures. Conversely, if the verification/monitoring process is executed on-line (during operations), it provides a useful feedback to the operators to guide safety interventions based on the understanding of which TL properties are closer to being violated. The approach helps assess the effectiveness of measures taken to address various risks, and it supports the identification of measures that are not yet implemented in the system design and vulnerabilities in the system, towards improved accident prevention and risk mitigation strategies. We will revisit these considerations in Section 4.

In Section 3 we examine a set of four safety principles, formulated at a high-level of abstraction, based on the notions of accident sequence and hazard level/escalation, which we introduce next. These safety properties, once expressed in TL, can be monitored during the design and operation of systems for compliance and be verified on-line and off-line, following the process previously outlined.

III. TL FORMULATION OF SYSTEM SAFETY PRINCIPLES

We conceive of an accident as resulting from the absence or inadequate enforcement of safety constraints[13]. Intuitively, the most simple safety constraint is of the form

$$\square \neg A \tag{1}$$

where A represents the occurrence of an accident or adverse event, and hence the expression reads: "the accident A never occurs". A similar approach was presented in (Ref. 14), where A represented a generic top-event of a fault-tree to be avoided. In order to operationalize a property such as (1), the first step is to understand how to further qualify and quantify A. As we discuss shortly, the introduction of the notion of hazard level will allow the translation of Eq. (1) into the quantifiable form provided by Eq. (6). The analytical definition of the hazard level provides an index to quantify the "hazardousness" associated to the system internal condition, and hence a quantification of the unfolding of accident A.

III.A The Notions of Accident Sequence and Hazard Level

An accident sequence can be viewed as a string of events, starting from an initiating event (IE) that leads the system into off-nominal conditions of operations and leading to an accident (A). For instance, in Figure 3 the string that starts with the initiating event IE_1 and terminated in the accident state A_k is written as:

$$s_{1,k} = \ IE_1 e_2 e_3 \ldots e_n A_k \tag{2}$$

where each event (e) in the sequence provided by Eq. (2) presents one subscript that identifies its position inside the string s. As indicated in Figure 3, multiple possible paths exist between different initiating events and accident states. The conditional probability of accident A_k occurring given the occurrence of the initiating event IE_i can be written as $p(A_k|IE_i)$. This conditional probability is the sum over all paths starting from IE_i and leading to A_k, and it is a key ingredient in PRA.

At a *local* level, given that an accident sequence has been initiated, the conditional probability that it will further advance or escalate is reflected in the conditional probability

$$p(e_{i+1}|e_i) \ \text{ or generally } \ p(e_k|e_i) \ \text{ for } k > i \tag{3}$$

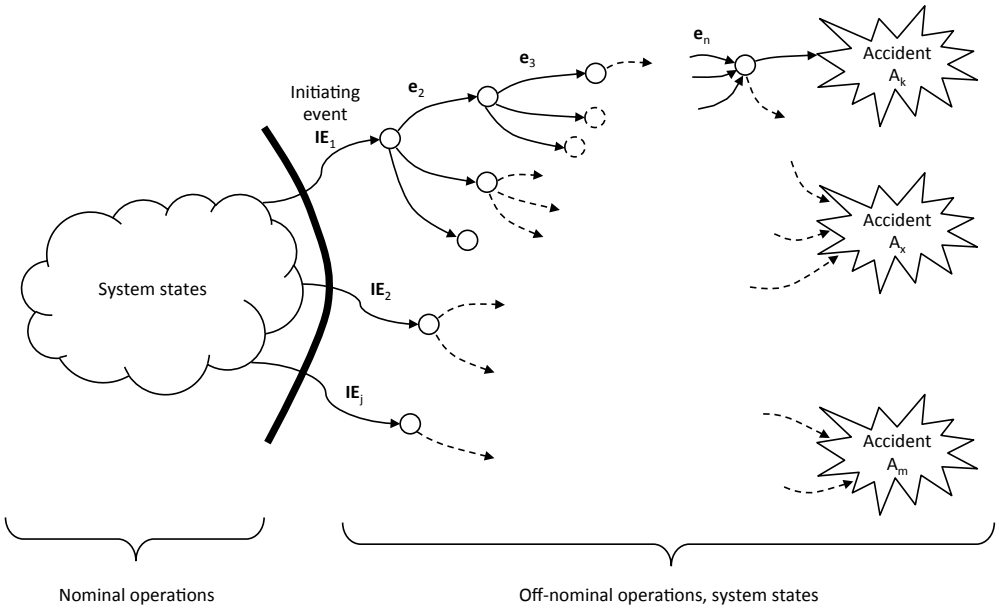

Figure 3 – Schematic representation of an accident sequence

The idea of an accident sequence and of the conditional probabilities associated with its escalation helps define and convey the notion of hazard level (H). Intuitively, the hazard level can be conceived of as the closeness of an accident to being released[15]. It is thus related to the extent an accident sequence has advanced: the further the sequence has escalated, the more hazardous the situation is for a given accident A_k. We can then represent an accident sequence evolution in time through the dynamic behavior of the hazard level, as seen in Figure 4. We define the following *dynamics* of hazard level[a]:

- Hazard Escalation: $dH/dt > 0$
- Hazard De-escalation: $dH/dt < 0$
- Hazard Constancy: $dH/dt = 0$

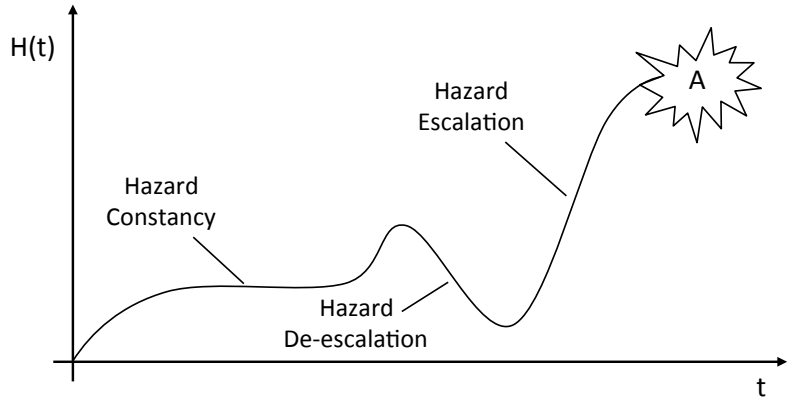

Figure 4 – Example of the hazard level dynamics

Traditional quantitative risk analysis involves the computation of the conditional probability associated with each scenario that leads to the occurrence of accident A. At its core risk analysis is the imagination of failure, and a significant effort is required to conceive of the many possible ways accidents can unfold. For each accident scenario, a probability like the one of Eq. (4), based on the scenario expressed by Eq. (2), needs to be computed.

$$p(s_{1,k}) = \; p(IE_1) \cdot p(e_2|IE_1) \cdot p(e_3|e_2) \ldots p(A_k|e_n) \tag{4}$$

The definition of the hazard level function is based on the state-space representation of the system[6, 7]. As noted in Figure 2, the approach proposed requires a model for the system description (i.e., a model for the system dynamics). Our approach falls then under the category of model-based safety analysis, (see (Ref. 16, 17) and references therein). Model-based safety analysis was first introduced to provide a more formal approach for analysis techniques that had traditionally been performed manually. In this work we chose to use the state-space representation to describe the system dynamics[6, 7], since accounting for the system state provides valuable information on the internal condition of the system, as we see in the following example.

Consider an oil tank that progressively fills up with raffinate. Monitoring for the accident "loss of containment (LoC) through tower overflow", the hazard level can be approximated by the height of the raffinate in the tower. In this 1-D case, a dimensionless hazard level for the considered accident is defined as:

$$H_{LoC}(t) = \frac{h(t)}{h_{max}} \tag{5a}$$

[a] Note that discontinuities (e.g., jumps) in the hazard level function H(t) may exist. In those cases, the definition of the hazard level dynamics can be interpreted in a discrete sense as $\Delta H/\Delta t$. In the practical implementation of the verification process (which is executed in a simulation environment), the definition of the derivative of the hazard level is always discretized.

where the height of raffinate in the tower h(t) represents a particular state variable of the dynamical system (oil tank and raffinate flows). In Eq. (5a) the height of the raffinate at time t is divided by the maximum achievable height before overflow occurs, so that the resulting hazard level is dimensionless. The situation H(t) = 1 indicates then overflow of the tower or the onset of the accident "loss of containment". Furthermore, given the current hazard level, and knowing the net flow into the tower[b], the remaining time before the LoC accident is released, assuming no changes are made to the operation of the system, can be simply expressed as follows:

$$\Delta T_{LoC} = \frac{h_{max} - h(t)}{\dot{h}(t)} = \frac{1 - H_{LoC}(t)}{\dot{H}_{LoC}(t)} \tag{5b}$$

The knowledge of these two "coordinates" of a hazard, $H_{LoC}(t)$ and ΔT_{LoC}, provides an important feedback for operators and decision-makers to manage the risk of loss of containment in real time. Furthermore, when other potential accidents are identified and their associated hazard coordinates are estimated, the result is a portfolio of hazard coordinates, which roughly translates into "how hazardous is a particular situation" and "how much time is left before their corresponding accident occur"[c]. This collective information can then be displayed dynamically in a "hazard temporal contingency map" (Fig. 5) to support operators' sensemaking and help them prioritize attention and defensive resources for safety interventions and accident prevention[d].

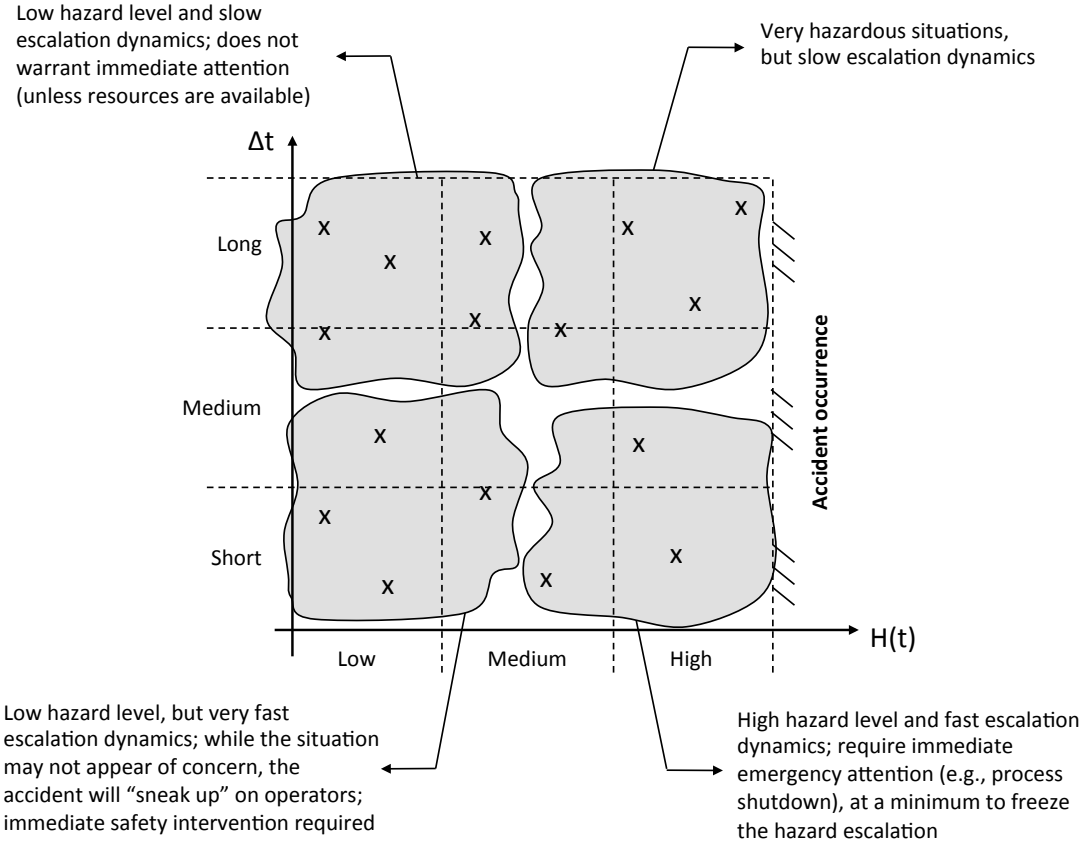

Figure 5 – Hazard temporal contingency map

[b] Typically measured with proper instrumentation of the inflow and outflow valves of the tower.
[c] Details on how to generalize the estimation of the remaining time before the accident release are provided in [Favarò & Saleh, 2016a].
[d] Trends over time and uncertainty bars in the estimates of both hazard coordinates can also be assessed and displayed.

The knowledge of the system state, in this example the height of the oil in the tower, is what allows the definition of an index for accident escalation. The introduction of a metric like the hazard level allows translating the qualitative Eq. (1) into a quantifiable constraint for the system behavior of the following form:

$$\square[\mathrm{H(t)} < \mathrm{H_A}] \tag{6}$$

where H_A represents the hazard level associated with the accident occurrence. The constraint posed by Eq. (6) is independent of the path that the system follows to reach the H_A threshold. In other words, constraints predicated on the values of the hazard level are agnostic to the series of events that led to that particular value of the hazard level, and are hence independent on the specific path followed by the system. The monitoring of the hazard level corresponding to a particular constraint together with the estimation of the remaining time before the accident occurrence provide an important feedback to operators to guide safety interventions. The constraint expressed by Eq. (6) is simple and intuitive; more complex cases, which have the advantage of providing detailed feedback during the verification process, are provided next.

III.B System Safety Principles

The hazard level provides a key ingredient for the definition of the safety properties we want a system to verify. In this work, we provide the TL formalization of four system safety principles, originally proposed in (Ref. 18). These principles, formulated at a high-level of abstraction, are domain-independent and can be applied/adapted to any engineering system. Before presenting their formalization, we provide a summary of their definition based on both traditional conditional probability and on the notion of hazard level (more details can be found in the original reference).

- *Fail-safe:* Given a function performed or implemented by a particular item in a system, the failure of said item should result in operational conditions that (i) block an accident sequence from unfolding, and/or (ii) freeze the dynamics of hazard escalation in the system, thus preventing potential harm or damage. The effects of this principle can be expressed as follows:

$$\mathrm{e_f}\text{: failure of the item/function of interest at time } \mathrm{t_f}$$

$$\begin{cases} \dfrac{\mathrm{dH}}{\mathrm{dt}} = 0 \quad \text{for } \mathrm{t} > \mathrm{t_{e_f}} & \text{(i.e., no further hazard escalation, see Fig. 3)} \\ \quad and & \\ \mathrm{p(e_{f+k} | e_f)} = 0 & \mathrm{e_{f+k}} \in s \text{ following } \mathrm{e_f} \end{cases} \tag{7}$$

- *Safety Margins*: This principle requires first an estimation of a critical hazard threshold for accident occurrence, $\widehat{H}_{crit}$, and an understanding of the dynamics of hazard escalation in a particular situation. Secondly, it requires that features be put in place, including feedback loops (to the automation and/or to the operators) to maintain the operational conditions and the associated hazard level H(t) at some "distance" away from the estimated critical hazard threshold or accident-triggering threshold. This buffer distance is expressed in terms of the safety margin (*SM*) as:

$$[\widehat{\mathrm{H}}_{\mathrm{crit}} - \mathrm{H(t)}] \geq SM \tag{8}$$

When equation (8) is satisfied as an equality for a particular value of H(t), we term the corresponding hazard level the "operational upper limit"(OUL).

- *Defense-in-Depth:* This fundamental safety principle requires that: (i) multiple lines of defenses or safety barriers be placed along potential accident sequences (identified through risk analysis); (ii) safety should not rely on a single defensive element (hence the "depth" qualifier in defense-in-depth); (iii) successive barriers should be diverse in nature and include technical, operational, and organizational safety barriers. The various safety barriers have different objectives and perform different functions. The first set of barriers, or line of defense, is meant to

prevent an accident sequence from initiating. It implies that safety features are devised and put in place such that the probability of an accident-initiating event (IE) is minimized:

$$\min[\mathrm{p(IE)}] \tag{9}$$

Should this first line of defense fail in its prevention function, a second set of safety defenses should be in place to block the accident sequence or minimize the likelihood of further escalation

$$\min[\mathrm{p}(e_{i+k}|e_i)] \quad \forall \mathrm{i,k\ for}\ e_i, e_{i+k} \in s \ \text{and}\ k \geq 1 \tag{10}$$

Finally should the first and second lines of defense fail, a third set of safety defenses should be in place to contain the accident and minimize its potential adverse consequences (*PAC*) :

$$\min\left(PAC \mid A\right) \tag{11}$$

- *Observability-in-Depth:* This principle is meant to eliminate the potential for safety blind spots—the concealment of hazardous states or event occurrence—in system design and operation, and to support operators' situational awareness. It requires that all safety-degrading events or states that safety barriers are meant to protect against be observable. The principle implies, among other things, that various features be put in place (i) to minimize the gap between the actual hazard level H(t) and the hazard level estimated by the operator $\hat{H}(t)$, and (ii) to ensure that at the hazard levels associated with the breaching of various safety barriers, the two hazard levels coincide, as indicated in Eq. (12).

$$\begin{aligned} &e_{b_i}: \text{ breach of safety barrier } b_i \\ &\begin{cases} \min \Delta H \ \Leftrightarrow \ \min\left\|\hat{H} - H(t)\right\| \\ \textit{and} \\ \Delta H_{b_i} = 0 \qquad \forall i \end{cases} \end{aligned} \tag{12}$$

One objective of observability-in-depth is to support the identification and sensemaking of emerging hazardous conditions. This in turn helps the understanding of what potential accident sequences that might follow, to guide decision-making and safety interventions by operators and management (more details follow in Section 4). The depth qualifier refers to the ability to observe accident pathogens and hazardous events as far back as possible (upstream in the causal chain) in an accident sequence, and hence to provide ample time for safety interventions and addressing emerging risks.

III.C TL Formalization

In this subsection, we make use of the logical operators presented in Tables 1 and 2, and of the elements defined in Eqs. (7-12). Equations (13-16) provide the TL formulae describing the safety principles introduced in the previous section. For each TL formula we provide a detailed explanation of how to read and interpret the syntax.

Each of the TL formulae presented next constitute a constraint on the system behavior. Once a model for the system is obtained, these requirements are checked and controlled for compliance/satisfaction according to the process show in Figure 2. The formulation of each TL formula is predicated on the hazard level function H(t). Note that multiple hazard level functions can be used for the properties definition (different hazard level function for each principle). We will revisit this point shortly.

- *Fail-safe:*

$$\mathrm{FS} \ \triangleq \{\, e_f(t = t_{ef}) \rightarrow [\Box(H(t) < H_{crit}) \wedge \Box_{t>tef} \neg \frac{dH}{dt} > 0] \tag{13}$$

The fail-safe principle revolves around the notion of an accident-triggering threshold (with corresponding hazard level H_{crit}). It is then fundamental for the correct implementation of the principle that a local failure event (e_f in Eq. 13) does not induce a breaching of such threshold and that the hazard level dynamics is not an escalating one. Equation (13) reads as follows: "If the failure event e_f occurs at time t_{ef}, then it is always true that the hazard level does not reach the critical level and it is true for all instants of time following t_{ef} that the hazard level does not escalate". As previously noted, Eq. (13) provides a quantifiable constraint that can be formally verified for compliance during system operations or during the design stages. The translation of a qualitative/descriptive safety principle into a quantitative definition is the fundamental step that allows the verification process of Figure 2. The violation of a safety principle like the one expressed in Eq. (13) provides useful insight towards several ends. Firstly, when different hazard level functions are used for each safety principle, the violation of a specific TL formula tells the operator which hazard level to monitor more closely (for complex systems several functions such as the one expressed in Eq. (5a) can be considered). Mapping the specific hazard level of interest into a diagram like the one of Figure 5 supports the on-line management, ranking, and recognition of the need for safety interventions. Additionally, the specific principle violated provides an important feedback for off-line considerations as well. For instance, if Eq. (13) is violated, this means that for that specific hazard the fail-safe principle was not correctly implemented. Changes in the layout of the available safety barriers, in the system design, and in the operating procedures can be put in place to overcome the lack of compliance identified by the TL formula violation.

- *Safety Margins:*

$$SM \triangleq \{ H_{OUL}(t = t_1) \rightarrow [\exists T: \Box_{t<t1+T}(H(t) < H_{crit})]\} \quad (14)$$

Central to the definition of the safety margins principle is a minimum required time T that ensures that a good time-window for operators' intervention can be established in between the time at which the operational upper limit is met and the time at which the accident triggering threshold is reached. Equation (14) reads as follows: "If the operational upper limit is reached at time t_1 then it is true that there exists a time T greater or equal to a pre-specified time-window such that for all instants of time before t_1+T the critical hazard level is not reached". A corollary of Eq. (14) is the need to embed in the system features that "slow down" the hazard escalation process, to buy the operators more time for safety interventions before an accident unfolds. As noted previously in regards to the fail-safe principle, changes in the system design and in the barriers layout (including alarms and warning systems to indicate that the operational upper limit has been met) can be considered to ensure compliance with Eq. (14).

- *Defense-in-Depth:*

$$PR \triangleq \{ \Box(H(t) < H_{crit})\} \quad (15a)$$

$$BL \triangleq \{ \blacklozenge (H(t) = H_{crit}) \rightarrow [\Diamond \frac{dH}{dt} \leq 0 \ \wedge \Box^{+}(H(t) < H_A)]\} \quad (15b)^{e}$$

$$MIT \triangleq \{ \blacklozenge (H(t) = H_A) \rightarrow [PAC_{|A} < \max(PAC)]\} \quad (15c)$$

Each equation provides the formalization of one of the three functions embodied by the defense lines: prevention; blocking; mitigation. Equation (15a) reads as follows: "It is always true that the hazard level does not reach the accident triggering threshold". This condition ensures that prevention barriers are put in place to maintain the system within its safe operating conditions. When this condition (Eq. 15a) is violated, Equation (15b) picks up the slack with the blocking function; it assumes that the first line of defense has been breached and the accident-triggering threshold has been reached. It reads as follows: "If at some point in the past the accident triggering threshold is met, then it must follow that at some point in the future the hazard level dynamics is frozen or de-escalating and that for all future instants of time the accident hazard level is not met". The same considerations apply to Eq. (15c) formalizing the last line of defenses, those that embody the mitigation function. Equation (15c) reads as follows: "If at some point in the past the hazard level associated with the accident unfolding is met, then the potential adverse consequences associated with the accident release are less than those of the worst-case scenario". Similar considerations to the ones

[e] In Eq. (15b) the operator $\Box^{+}$ indicates all future instants of time.

highlighted for the previous principles also apply to defense-in-depth. Ingenuity is required to consider what improvements/re-design in the system layout and operations are required in case this principle is violated.

- *Observability-in-Depth (OID):*

$$\text{OID1} \triangleq \{\, \square\neg\left(\left\|H(t) - \widehat{H}(t)\right\| > \varepsilon\right)\} \tag{16a}$$

$$\text{OID2} \triangleq \{\, \square[e_{b_i}(t = t_i) \rightarrow \frac{d\left(H(t) - \widehat{H}(t)\right)}{dt} \leq 0]\} \tag{16b}$$

As previously noted, the separate notions of the actual hazard level and of the estimated hazard level play a central role in this principle. The definition of the OID principle is predicated on their values and on the absence of a gap between the two (actual versus estimated hazard levels), as indicated in Eq. (16a), which reads as follows: "It is never the case that the actual and the estimated hazard level differ from each other of more than an admissible pre-set tolerance ε". The second ingredient to the OID principle derives from the feedback provided by safety barriers that are breached during the dynamics of hazard escalation. Equation (12) required a zero gap between the actual and the estimated hazard level after each barrier breaching. As this may not always be realistic (for instance due to the transients in change in the hazard level functions), this consideration is relaxed in Eq. (16b), which reads as follows: "It is always true that if a barrier is breached at time t_i, then the discrepancy between the actual and the estimated hazard level is either constant or decreasing". Violations of these properties imply a degraded situational awareness for the operator, who is left blind to, or with an inaccurate estimation of the hazard escalation dynamics. Violations of the OID principle identify the need to improve or redesign the system instrumentation and sensoring layout, in order to provide the operator with a better estimation of the actual internal states of the system, especially as they pertain to hazardous conditions.

Equations (13-16) translate the verbal and qualitative definition provided in the previous section into quantifiable and formally verifiable safety properties. As noted previously, the verification of multiple TL properties in real-time provides information on which hazard levels to monitor more closely. The hazard level monitoring in turns provides an important feedback to recognize the need for, rank, and trigger safety interventions. The violations of specific safety properties also warrant the need additional off-line interventions on the system design. The next section addresses some of the details behind the verification process and the insights that derives from the violation of the TL properties and hazard level monitoring.

IV. VERIFICATION OF SAFETY PROPERTIES AND HAZARD MONITORING

The TL formulae presented in the previous section act as safety constraints, which are then checked and verified for satisfaction. If the TL formulae are satisfied, then the verification process works towards ensuring the safe behavior of the system. What happens if they are violated? As seen in Figure 2, this condition is the starting point for a feedback loop that returns important information for re-engineering the system design, the barriers layout, the operating procedures, and the system instrumentation, just to mention a few possible safety interventions that can be triggered by said feedback. The process previously described is summarized in Figure 6.

When the verification of properties satisfaction fails, the first line of action is understanding which of the requirements specified in TL is being violated. As mentioned in Section 3, it is possible to set up different hazard level functions for each principle (H_1 to H_n in Figure 6). Note that violations of the TL formulae presented in Eq. (13-16) occur and trigger alarms and warnings *before* the accident threshold H_A is reached, warranting the close monitoring of the corresponding hazard level so that safety interventions can be ranked, prioritized, and executed in a timely fashion (according to the hazard temporal contingency map presented in Figure 5). Additionally, **the violation of the safety principles provides diagnostic information regarding missing/inadequate safety features embedded in the system**. On the one hand, monitoring the dynamical behavior followed by H(t) up to the point in which the TL formula was violated helps answering important questions regarding ineffective safety features and missing/ineffective feedback that would have allowed the operator to acknowledge a potentially dangerous situation beforehand (e.g., we could ask: "Why did hazard escalation occur at that point? Why weren't barriers devised to prevent that?"). On the other hand, monitoring the current value of H(t) is necessary to achieve an estimation of the time-window available for the operator before the system reaches the H_A threshold. Safety interventions will then be focused on managing and lowering the value of H(t) as far away from H_A as possible (at least below H_{crit} - SM). The whole process provides useful feedback and insights for various stakeholders, from management to designers, operators, and technicians, to guide safety interventions both on-

line (towards accident prevention and/or mitigation) and off-line (towards re-design and re-engineering of safer systems).

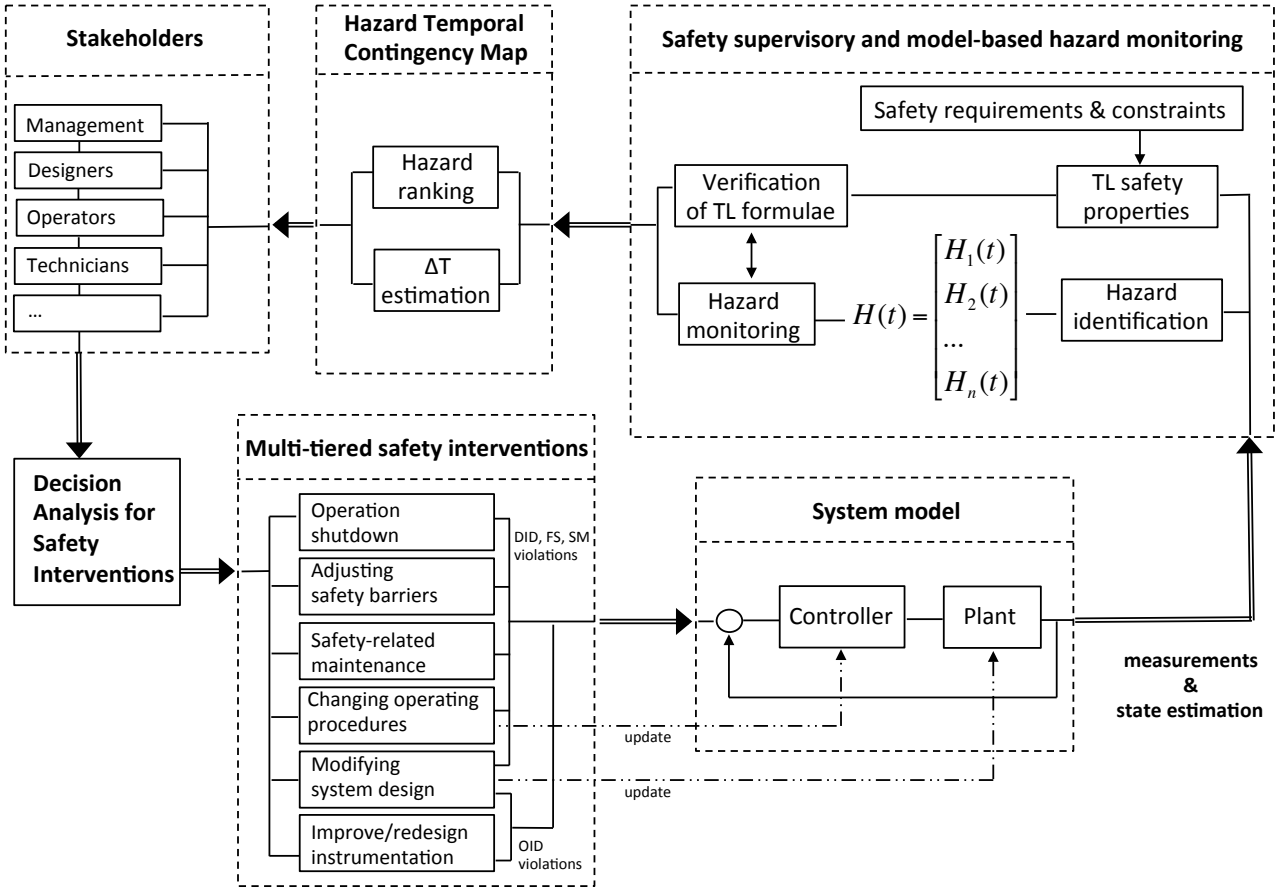

Figure 6 – Hazard Informed System Monitoring for Safety Interventions – process map

On a practical level, the implementation of monitoring devices should allow the operator to closely monitor for satisfaction or violation of each TL property, and to obtain detailed information regarding the dynamics and the current value of the corresponding hazard level. This process provides useful feedback to the operators, who can then understand which TL properties are closer to being violated and hence which situations warrant immediate attention. In practical applications, a control panel such as the one of Figure 7 can support the operators' monitoring effort, informing them of which safety requirements are satisfied and which are not, and allowing, for instance by double-clicking on the safety principle of interest, to monitor the corresponding hazard level(s) and visualize the estimation of the remaining time-window before an accident occurs.

The monitoring of the hazard level and the verification of the TL safety properties can assist in a dynamic and real-time accident prevention and risk management, where hazards are monitored, prioritized, and ranked based on their closeness to being released. This helps with the identification of emerging risks and towards a "temporal ranking of hazards", where accidents that are closer to being released warrant more attention and timely intervention. We term this dimension *temporal contingency*. The notion of temporal contingency provides a complementary perspective to the traditional notion of risk in Probabilistic Risk Assessment (PRA), augmenting traditional perspectives by adding a temporal dimension (a "time-to-accident" metric) to the estimation of the hazard associated to a particular situation. This process is independent on the specific path followed by the system, shifting the attention away from the calculation of conditional probabilities, and translating the event-based perspective of traditional PRA into time-based considerations.

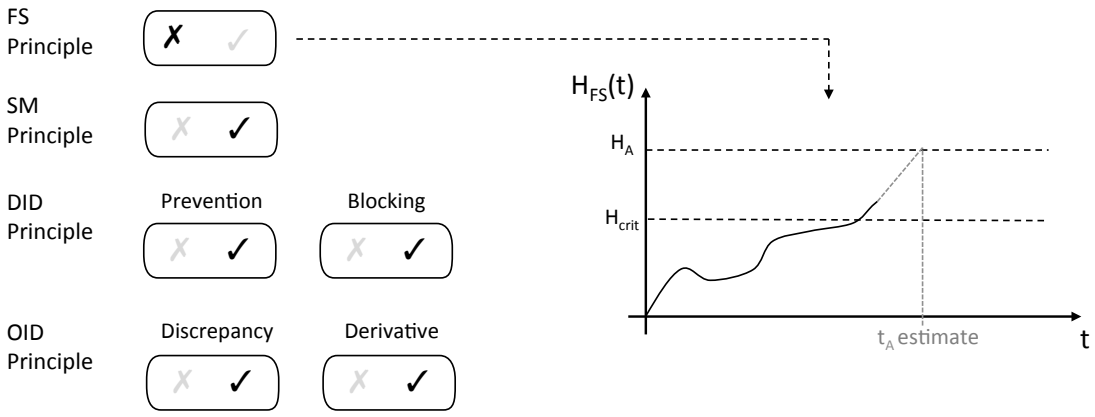

Figure 7 – Schematic view of the control panel to monitor the violation of the TL formulae and the corresponding hazard level. Here the symbol ✓ stands for property verified, while ✗ stands for property violated.

V. CONCLUSIONS

This work is part of a larger effort whose objective is to explore the use of TL for risk analysis and system safety applications. Specifically, in this work, we proposed the application of temporal logic for the definition of TL system safety properties that act as constraints on the system behavior. We reviewed a set of four system safety principles, and formalized them using temporal logic. The analysis here presented served multiple purposes, and several considerations were highlighted. The main ideas are summarized next:

- The approach presented can be applied off-line and on-line. Off-line analysis helps during the system development stages to better understand the need for re-design and re-engineering of safety features, system instrumentation, layout, and operating procedures. It supports and assists management and designers to ensure that safety requirements are met by the current system design. On-line analysis, done during system operations, deals with real-time hazard level monitoring and the recurrent checks of the TL properties. It serves the operator in improving situational awareness and provides a useful feedback to recognize the need for, rank, and trigger safety interventions in a timely manner.
- Through the hazard level monitoring and the verification of the TL safety constraints, the approach helps assess the effectiveness of measures taken to address various risks, and it supports the identification of measures that are not yet implemented in the system design and vulnerabilities in the system, towards improved accident prevention and risk mitigation strategies.
- The proposed approach can assists in the development of a "dynamic real-time accident prevention and risk management", where hazards are prioritized and ranked based on their closeness to being released. This dimension of temporal contingency directly stems from the verification of TL formulae and the real-time evaluation of the hazard level, and it constitutes an important advantage of introducing temporal logic for risk analysis. The real-time evaluation of the hazard level provides information not only on the satisfaction/violation of the TL safety specifications, but also on which requirements are about to be violated and require immediate attention/intervention. The notion of temporal contingency complements the notion of risk provided by traditional PRA by adding a temporal dimension to it.
- The verification of the safety principles is independent of the particular path followed by the system (as opposed to the event-driven and path-dependent computation of the conditional probabilities necessary for PRA). Multiple scenarios can be easily handled by considering multiple hazard level indices in the same analysis.
- Finally, the introduction of temporal logic for safety purposes provides common ground between the risk community and the software community. Providing common semantics across these two communities is an important step to ensure the integration of safety in the early steps of design of system, especially software-intensive systems.

This preliminary work showed that the adoption of the TL language, a non-traditional choice for the risk analysis and system safety domain, offers novel capabilities, complementary to PRA and rich possibilities for further contributions toward accident prevention and improved risk management.

REFERENCES:

1. Kress-Gazit, H., Fainekos, G. E., & Pappas, G. J. (2009). *Temporal-logic-based reactive mission and motion planning.* IEEE Transactions on Robotics, 25(6), 1370-1381.
2. Zhang, J., & Cheng, B. H. (2006). *Using temporal logic to specify adaptive program semantics*. Journal of Systems and Software, 79(10), 1361-1369.
3. Fainekos, G. E., Girard, A., Kress-Gazit, H., & Pappas, G. J. (2009). *Temporal logic motion planning for dynamic robots*. Automatica, *45*(2), 343-352.
4. Baier, C., & Katoen, J. P. (2008). *Principles of model checking* (Vol. 26202649). Cambridge: MIT press.
5. Mosleh, A. (2014). *PRA: a perspective on strengths, current limitations, and possible improvements*. Nuclear Engineering and Technology, 46(1), 1-10.
6. Favarò, F.M., Saleh, J.H. (2016). *Towards Risk Assessment 2.0: Safety Supervisory Control and Model-Based Hazard Monitoring for Risk-Informed Safety Interventions,* Reliability Engineering and System Safety, Vol. 152, pp. 316-330.
7. Favarò, F.M., Saleh, J.H. (2016). *Applications of Temporal Logic Safety Supervisory Control and Model-Based Hazard Monitoring*, Safety Science, under review.
8. Galton, A. (Ed.). (1987). *Temporal logics and their applications* (Vol. 10). London: Academic Press.
9. Fisher, M. (2011). *An Introduction to Practical Formal Methods Using Temporal Logic*. John Wiley & Sons.
10. Zio, E. (2014). *Integrated deterministic and probabilistic safety assessment: Concepts, challenges, research directions*. Nuclear Engineering and Design, 280, 413-419.
11. Rescher, N., & Urquhart, A. (1971). *Temporal Logic, Vol. 3* of Library of Exact Philosophy. Springer–Verlag, Heidelberg, Germany, 42, 140.
12. Manna, Z., Pnueli, A. (1992). *Temporal logic of reactive and concurrent systems* (Vol. 1). Springer.
13. Leveson, N. (2004). *A new accident model for engineering safer systems*. Safety science, 42(4), 237-270.
14. Hansen, K. M., Ravn, A. P., & Stavridou, V. (1998). *From safety analysis to software requirements*. Software Engineering, IEEE Transactions on, 24(7), 573-584.
15. Saleh, J. H., Haga, R. A., Favarò, F. M., & Bakolas, E. (2014). *Texas City refinery accident: Case study in breakdown of defense-in-depth and violation of the safety–diagnosability principle in design*. Engineering Failure Analysis, 36, 121-133.
16. Papadopoulos, Y., McDermid, J., Sasse, R., & Heiner, G. (2001). *Analysis and synthesis of the behaviour of complex programmable electronic systems in conditions of failure*. Reliability Engineering & System Safety, 71(3), 229-247.
17. Bozzano, M., Villafiorita, A., Åkerlund, O., Bieber, P., Bougnol, C., Böde, E., ... & Zacco, G. (2003). *ESACS: an integrated methodology for design and safety analysis of complex systems*. In Proc. ESREL (pp. 237-245).
18. Saleh, J. H., Marais, K.B., Favarò, F.M. (2014b). *System safety principles: A multidisciplinary engineering perspective.* Journal of Loss Prevention in the Process Industries, 29, 283-294.