SAFETY SUPERVISORY CONTROL, MODEL-BASED HAZARD MONITORING, AND TEMPORAL LOGIC: DYNAMIC RISK-INFORMED SAFETY INTERVENTIONS AND ACCIDENT PREVENTION

A Thesis
Presented to
The Academic Faculty

by

Francesca M. Favaro'

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
May 2016

SAFETY SUPERVISORY CONTROL, MODEL-BASED HAZARD MONITORING, AND TEMPORAL LOGIC: DYNAMIC RISK-INFORMED SAFETY INTERVENTIONS AND ACCIDENT PREVENTION

Approved by:

Dr. J. H. Saleh, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. B. German
School of Aerospace Engineering
Georgia Institute of Technology

Dr. D. Mavris
School of Aerospace Engineering
Georgia Institute of Technology

Dr. K. Marais
School of Aeronautics and Astronautics
Purdue University

Dr. E. Feron
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: March 4th, 2016

To my amazing family

ACKNOWLEDGEMENTS

I consider this thesis as my biggest academic accomplishment. However, this would not have been possible without the constant help, advice, and encouragement of my amazing advisor Dr. J. H. Saleh, to whom I express my most sincere gratitude. The PhD is characterized by a multitude of ups and downs, and he was capable of always shining a light in the darkest hours, and eager and ready to celebrate every milestone achieved.

I also wish to acknowledge the members of the committee for their time and the insights they provided me with through their academic guidance and class teachings, so "thank you" Dr. Mavris, Dr. Feron, Dr. German, and Dr. Marais.

Un grazie di cuore e speciale alla mia famiglia, in primis Matti compagno di avventura/sventura in questo dottorato, Sofia e genitori, fratelli, nonni e bisnonni. Senza il vostro sostegno non sarei mai arrivata a questo traguardo. Matti, sai che in tanti momenti sei stato la mia roccia ("sono roccia...") su cui sbollire rabbie, depressioni, e desiderio di mollare tutto, sempre pronto a darmi coraggio e farmi forza per arrivare fino in fondo. Non vedo l'ora di camminare su quel palco insieme a te!

Last but not least, I wish to thank all my dearest friends and officemates. You made this journey much more fun and pleasant. I will miss (and already do) the occasional break for coffee/ice creams/pastries/yoga/mani-pedi (Rachel I am talking to you), the nights out for painting classes, trivia, the endless shopping sprees, the Thanksgiving/Easter dinners and, simply put, you.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF SYMBOLS AND ABBREVIATIONS	xii
SUMMARY	xv
CHAPTER	
1 Introduction	1
1.1 Motivations and Objectives	1
1.2 A Novel Framework for Dynamic Risk-Informed Safety Interventions	5
1.3 Presentation Plan	7
2 Literature Review and State-of-the-Art	9
2.1 Seminal Works and Historical Development	9
2.2 PRA as State-of-the-Art: Current Challenges and Limitations	14
2.2.1 The PRA Workflow	14
2.2.2 Open Challenges and Limitations	16
2.3 DPRA: an Answer to the Time-Dependency Limitations of PRA	18
2.4 Current Approaches Involving the Use of Temporal Properties	20
3 Safety Supervisory Control Framework and Model-Based Hazard Monitoring	24
3.1The Safety Supervisory Control Framework	24
3.2 Model Development	29
3.2.1 Model-Based Safety Analysis	30
3.2.2 State-Space Representation	31

3.3 Safety Supervisory Monitoring 35
3.3.1 Hazard Level Identification and State Mapping 35
3.3.2 Hazard Level Monitoring 42
3.3.3 Example Application of H(t) Monitoring 45
3.4 Hazard Temporal Contingency Analysis 50
4 Temporal Logic Syntax and Properties Formulation 55
4.1 The Adoption of Temporal Logic 55
4.2 TL Syntax and its Use for Verification Purposes 57
4.2.1 TL Temporal Operators 58
4.2.2 Verification of Properties Expressed in TL 60
4.3 System Safety Principles 62
4.3.1 The Fail-Safe Principle 63
4.3.2 The Safety Margins Principle 65
4.3.3 The Defense-in-Depth Principle 66
4.3.4 The Observability-in-Depth Principle 69
4.4 TL Formulation of the Safety Principle 71
4.4.1 The Fail-Safe TL Property 72
4.4.2 The Safety Margins TL Property 73
4.4.3 The Defense-in-Depth TL Property 74
4.4.4 The Observability-in-Depth TL Property 75
5 Integrating TL and Model-Based Hazard Monitoring 78
5.1 TL in Support of the Safety Supervisory Control Framework 78
5.2 Application of the Integrated Framework and Case Study 82
5.2.1 Accident Narrative 83
5.2.2 State-Space Representation and Simulink Model 85

5.2.3 Hazard Monitoring and TL Safety Properties Verification 90

5.3 Insights and Advantages Enabled by the Approach 106

6 Conclusions and Future Work 110

6.1 Summary of Contributions 110

6.2 Future Work 113

6.2.1 Monitoring vs. Model Checking: Towards Automated Safety Verification 113

6.2.2 Displays and Visual Aides Development 114

APPENDIX A: Notes on Human Supervisory Control 117

APPENDIX B: A Comparison with PRA Techniques 124

APPENDIX C: Primitives of Causality and the Notion of Agonist, Antagonist, and Inverse Agonist 132

APPENDIX D: Operational Guidelines and Notes on the Framework Applicability 144

REFERENCES 151

LIST OF TABLES

Table 1.1: Financial losses and casualties for the accidents of Figure 1.1 2

Table 4.1: Temporal operators 58

Table 4.2: Logical connectives of classical logic 59

Table B.1: Comparison of the approaches 128

LIST OF FIGURES

Page

Figure 1.1: The Costa Concordia capsizing, the explosion of the Deepwater Horizon oil rig, and the crash of the Air France flight 447 2

Figure 1.2: Overview of the proposed framework 5

Figure 1.3: Schematic representation of "core chapters" structure 8

Figure 2.1: Evolution in the development of system safety approaches 12

Figure 3.1: Overview of model-based safety supervisory control, and dynamic hazard monitoring for safety interventions 25

Figure 3.2: Schematic representation of an oil tank 33

Figure 3.3: Illustration of hazard level dynamics 37

Figure 3.4: Schematic representation of an accident sequence 41

Figure 3.5: Hazard level dynamics for the oil tank example and comparison with criticality thresholds 43

Figure 3.6: Contours of H(t) plotted as a function of the initial conditions for the RTO 47

Figure 3.7: Contours of H(t) and comparison with V_1 limit 48

Figure 3.8: Comparison of "danger areas" and typical aircraft takeoff trajectory for best-case scenario 49

Figure 3.9: Illustrative hazard temporal contingency map 51

Figure 3.10: Illustrative estimation of time-to-accident for two hazard indices 53

Figure 4.1: Representation of "A∧O(B)" 59

Figure 4.2: Schematic representation of the verification process 60

Figure 4.3: Illustrative comparison of system behavior over time following a local failure with and without FS principle implementation 64

Figure 4.4: Illustration of the SM principle with sample accident trajectory 65

Figure 4.5: Illustration of the DID principle with sample accident trajectory 67

Figure 4.6: Hazard escalation over time and violation of the OID principle 70

Figure 5.1: Learjet 60 runway overrun STEP diagram 83

Figure 5.2: Screenshot of the Simulink model for the aircraft, the FADEC, and the TL property verification at takeoff 86

Figure 5.3: Dynamical system model 89

Figure 5.4: FS property – associated H(t) and violation detection 94

Figure 5.5: Implementation of the FS property in Simulink 95

Figure 5.6: Warning detection example 95

Figure 5.7: Contours for H(t) as a function of RTO initial condition. Best-case 97

Figure 5.8: Contours for H(t) as a function of RTO initial condition. Worst-case 98

Figure 5.9: Learjet trajectory during the accident sequence, superimposed to 5.8 99

Figure 5.10: Position, acceleration, and airspeed as per FADEC output and as per cockpit input provided by the pilots 102

Figure 5.11: Position discrepancy, OID violation detection, and time of barrier breaching 104

Figure 5.12: Acceleration discrepancy, OID violation detection, and time of barrier breaching 104

Figure 5.13: Implementation of the OID TL property in Simulink 106

Figure 6.1: Illustrative “control panel” for monitoring the verification of TL properties against multiple hazard levels 115

Figure 6.2: Illustrative “radar plot” for concurrent monitoring of multiple hazard levels 116

Figure A.1: Flowchart of the five supervisory functions 118

Figure A.2: Adaptation of the supervisory flowchart to the proposed framework 119

Figure A.3: Computer-based observer as an aid to the supervisor 122

Figure B.1: Integrated process for aircraft safety design 124

Figure B.2: Example of typical fault tree 126

Figure C.1: Bath-tub curve 132

Figure C.2: Hazard dynamics due to Agonist action 134
Figure C.3: Hazard dynamics due to Antagonist action 135
Figure C.4: Hazard dynamics due to Inverse Agonist action 135
Figure C.5: Agonist and Antagonist interactions in matrix form 137
Figure C.6: Direct Causation primitive of causality 138
Figure C.7: Blocking primitive of causality 138
Figure C.8: Despite primitive of causality 139
Figure C.9: Prevention primitive of causality 139
Figure C.10: Fragilizing primitive of causality 140
Figure C.11: Letting primitive of causality 141
Figure C.12: Primitives of causality – Agonist and Antagonist interactions 141
Figure C.13: De-escalation primitive of causality 142
Figure C.14: Primitives of causality – Agonist and Inverse Agonist interactions 143
Figure D.1: Operational steps for the application of the proposed framework 150

LIST OF SYMBOLS AND ABBREVIATIONS

$\wedge$	AND (Boolean Logic)
$\vee$	OR (Boolean Logic)
$\neg$	Negation operator
$\rightarrow$	Implication operator
$\exists$	Existence operator
$\Box$	Always operator
$\Diamond$	Eventually operator
O	Next operator
ρ	Density [kg/m^3]
Ψ	Control Matrix – Hazard Equation
U	Until operator
R	Release operator
a	Agonist action
$\bar{a}$	Antagonist action
A	Accident occurrence
C_L	Lift coefficient
C_D	Drag coefficient
e_f	Local failure event
e_{bi}	Breaching of barrier b_i
ia	Inverse Agonist action
L	Lift [N]
D	Drag [N]

h(t)	Height [m]
H(t)	Hazard Level
H_{crit}	Critical threshold for the hazard level
m	Mass [kg]
$\dot{m}$	Mass flow [kg/s]
p(t)	Pressure [Pa]
t	time [s]
T(t)	Temperature [C^o]
u(t)	Input vector
V(t)	Volume [m^3]
V_1	Decision speed
V_r	Rotation speed
W	Weight [N]
x(t)	State vector
y(t)	Output vector
AC	Advisory Circular
ARP	Aerospace Recommended Practice
CFR	Code of Federal Regulations
DID	Defense-in-Depth
DPRA	Dynamic Probabilistic Risk Assessment
ECU	Engine Control Unit
ET	Event Tree
FAA	Federal Aviation Administration
FADEC	Full Authority Digital Engine Control
FMEA	Failure Modes and Effects Analysis

FS	Fail-Safe
FT	Fault Tree
HRA	Human Reliability Analysis
HRO	High Reliability Organizations
IDPSA	Integrated Deterministic and Probabilistic Safety Analysis
IE	Initiating Event
LoC	Loss of Coolant
MLG	Main Landing Gear
MMD	Man Made Disasters
NA	Normal Accidents
NTSB	National Transportation Safety Board
OID	Observability-in-Depth
OUL	Operational Upper Limit
PAC	Potential Adverse Consequences
PoC	Primitives of Causality
PRA	Probabilistic Risk Assessment
RBD	Reliability Block Diagram
RTO	Rejected Take-Off
SM	Safety Margin
STAMP	System Theoretic Accident Model and Processes
STEP	Sequential Timed Events Plotting
TL	Temporal Logic
TR	Thrust Reversers
UAV	Unmanned Air Vehicle

SUMMARY

Accident prevention and system safety are important considerations for many industries, especially large-scale hazardous ones such as the nuclear, the chemical, and the aerospace industries. Limitations in the current tools and approaches to risk assessment and accident prevention are broadly recognized in the risk research community. Furthermore, as new technologies and systems are developed, new failure modes can emerge and new patterns by which accidents unfold. A safety gap is growing between the software-intensive technological capabilities of present systems and the still "too much hardware oriented" current approaches for handling risk assessment and safety issues.

To overcome these limitations, a novel framework and analytical tools for model-based system safety, or safety supervisory control, is developed to guide safety interventions and support a dynamic approach to risk assessment and accident prevention. This integrated approach rests on two basic pillars: (i) the use of state-space models and state variables (from Control Theory) to capture the dynamics of hazard escalation, and to both model and monitor "danger indices" in a system; and (ii) the adoption of Temporal Logic (TL, from Software Engineering) to model and verify system safety properties (or their violations, hence identify vulnerabilities in a system). The verification of whether the system satisfies or violates the TL safety properties along with the monitoring of emerging hazards provide important feedback for designers and operators to recognize the need for, rank, and trigger safety interventions. In so doing, the proposed approach augments the current perspective of traditional risk assessment with its reliance on

probabilities as the basic modeling ingredient with the notion of temporal contingency, a novel dimension here proposed by which hazards are dynamically prioritized and ranked based on the temporal vicinity of their associated accident(s) to being released. Additionally, the online application of the proposed tools and the ensuing insights can support situational awareness and help inform decision-making during emerging hazardous situations.

The integrated framework is implemented in Simulink and is capable of combining hardware, software, and operators' control actions and responses within a single analysis tool, as examined through its detailed application to runway overrun scenarios during rejected takeoffs (RTO). New insights are enabled by the use of temporal logic in conjunction with model-based system safety. For example, new metrics and diagnostic tools to support pilots' go/no-go decisions and to inform safety guidelines are derived. Limitations exists in the current recommended practice that advises pilots to initiate RTOs only before the decision speed V_1 is reached, as suggested by current statistics regarding RTOs accidents and as recognized by aircraft manufacturers. The new proposed metrics are capable of accounting for both situations in which RTOs are initiated below the traditional decision speed V_1 and still result in an accident, and situations for which RTOs are initiated above V_1 that do not. Moreover, within the context of a detailed case study, a new TL safety constraint is proposed to overcome an identified latent error in the logic of the Full Authority Digital Engine Control (FADEC) at takeoff, which in this case escalated a hazardous condition into a fatal crash. In short, by leveraging tools that are not traditionally employed in risk assessment, the framework and tools proposed offer novel capabilities, complementary to the traditional approaches

for risk assessment, and rich possibilities for informing safety interventions (by design and in real-time during operations) and towards improved accident prevention.

CHAPTER 1

INTRODUCTION

Accident prevention and system safety are important considerations for many industries, especially large scale hazardous ones such as the nuclear, the chemical, and the airline industries. Broadly speaking, system safety refers to the state of sustainably ensuring accident prevention through coordinated actions, strategic and tactical, on multiple safety levers, technical, organizational, or regulatory.

The interest in accident causation and system safety is self-evident, but it is worth articulating in order to provide a general background and motivation for the present work.

This chapter is structured as follows. Section 1.1 introduces the motivations and the objectives of this work. Section 1.2 presents a high-level overview of the novel framework and analytical tools proposed in the thesis. Section 1.3 provides the presentation plan of the thesis.

1.1 Motivations and Objectives

High-visibility accidents such as the crash of the Air France flight 447, the capsizing of the Costa Concordia, the explosion of the Deepwater Horizon drilling rig, or tragedies like the Bhopal and the Chernobyl disasters are often invoked to motivate an interest in accident prevention and system safety (Figure 1.1). Such accidents have a high impact on the media as they generally result in dramatic casualty tolls, significant financial losses, and environmental damages (Table 1.1). Unfortunately, industrial accidents, also known under the broader designation of *organizational or system accidents*, happen much more frequently than what may be conveyed by the "high-visibility" above-the-media-radar-screen accidents [Singer and Endreny, 1993].

Examples of such accidents abound in many industries, such as the chemical, oil and gas, mining, and transportation industries to name a few.

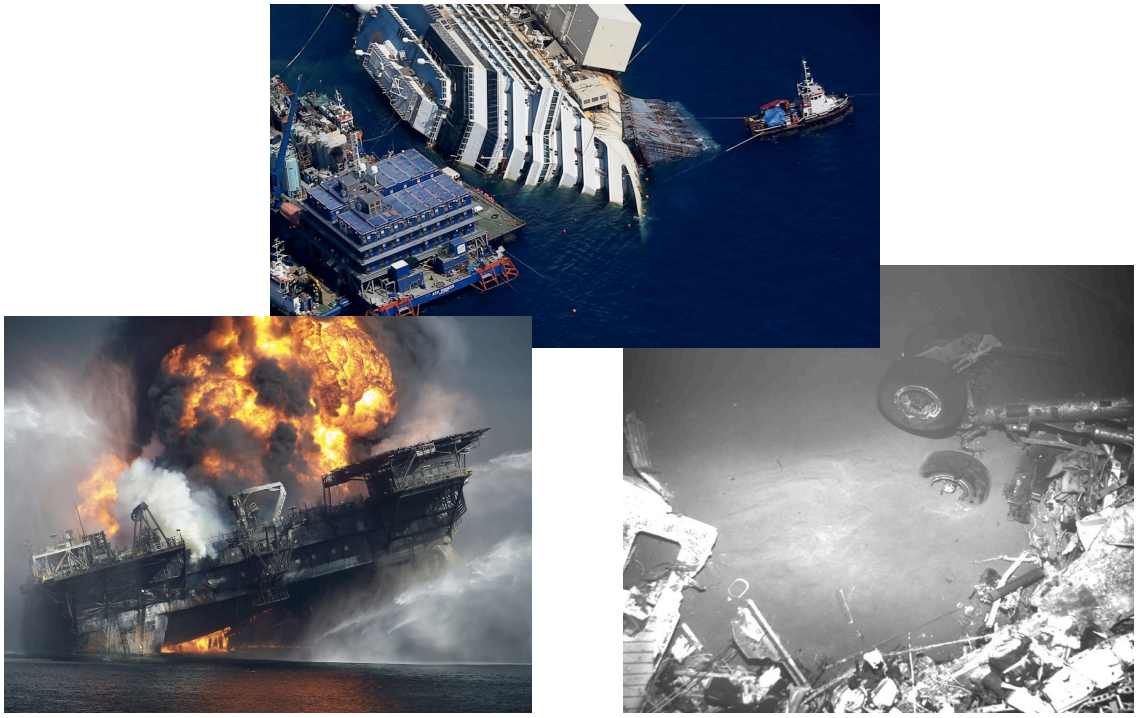

Figure 1.1 The Costa Concordia capsizing, the explosion of the Deepwater Horizon oil rig, the crash of the Air France flight 447 (credits: depositphotos.org)

Table 1.1 Financial losses and casualties for the accidents of Figure 1.1 (source: www.reuters.com)

Accident	Financial Loss	Casualties
Costa Concordia	> 600 $M	31
Air France 447	> 300 $M	228
Deepwater Horizon	> 4 $B	11

When carefully analyzed, many system accidents share a conceptual sameness in the way they occur, through a combination of system design and technical flaws, operational or workforce failings, compromised organizational behaviors and

management shortcomings, and/or deficient regulatory oversight [Saleh et al., 2010]. This observation of a conceptual sameness in the way system accidents occur along with the propensity for this class of adverse events suggests that some limitations may exist in the current way of thinking about and handling of these issues, and are indicative of theoretical deficiencies in the understanding of system accident causation and prevention.

Such limitations and deficiencies are becoming more evident with the increasing reliance on software-intensive systems in our daily lives and for process control. As new technologies and systems are developed, new failure modes emerge and new patterns by which accidents unfold. It is important to adopt a proactive safety attitude in understanding what these new failure modes might be and pre-empt them. A safety gap is growing between the software-intensive technological capabilities of present systems and our understanding of the ways they can fail, thus hindering the ability to prevent accidents. Other authors [Zio 2014; Mosleh, 2014] have expressed concerns regarding the "too much hardware oriented" approaches of the traditional tools of Probabilistic Risk Assessment (PRA), and advocated new and improved approaches to system safety and accident prevention in these regards.

Following to the increasing spread of cyber-physical systems, where interactions between technologically-advanced hardware, software, and human operators are necessary, there is a demand for novel approaches that can combine all these aspects within the same analytical framework. Moreover, while different analytical tools are available for risk analysis (many of which are included under the heading of PRA or variations on it), formal frameworks and analytical approaches also able to tackle system safety issues are conspicuously missing from the safety literature.

System safety and risk analysis, while complementary to each other, differ in one important way. Risk analysis, at its core, is the imagination of failure. Whether in a safety or security context – depending on the absence/presence of active volition – risk

analysis is anticipatory rationality examining the possibility of adverse events and failure mechanisms. The tools of risk analysis support this imaginative effort; they help identify and prioritize risks, inform risk management, and support risk communication. They do not provide however design or operational guidelines and principles for eliminating or mitigating the identified risks. The tools subsumed under risk analysis can help assess the effectiveness of measures taken to address various risks, but they offer no support in identifying or conceiving what these measures ought to be. Such considerations fall instead within the purview of system safety.

In this work it is proposed that the application of formal analytical tools to system safety issues as well as to traditional risk assessment procedures can not only work towards the identification and prioritization of emerging hazards in a system, but also towards guiding safety interventions on the system in both on-line and off-line contexts.

At a macro level, this work addresses the limitations previously highlighted by setting forward three main objectives:

1. The exploration of novel approaches and analytical tools to bear on risk assessment and system safety issues, inspired by important technical disciplines that still struggle to make a stand in the risk/safety community, such as of Control Theory and Computer Science/Software Engineering;
2. The development of an integrated framework able to handle hardware, software, and the effects of operators' control actions and responses to emerging hazards within the same analysis;
3. The development of an integrated framework that leverages formal approaches and novel techniques for *both* the identification/ranking/prioritization of interventions for emerging hazards (risk assessment purview), *and* the support and guidance for better informed decision-making regarding both on-line and off-line safety interventions (system safety purview).

The introduction of new concepts and complementary perspectives beyond the current probability-based toolset of risk analysis provides a useful addition for many safety practitioners. Additionally, the set up of novel bases, formulated at a high-level of abstraction, for system safety and accident causation deserve a careful attention, given their potential for "export" and broad application and adaptation to several different engineering domains.

1.2 A Novel Framework for Dynamic Risk-Informed Safety Intervention

In order to accomplish the objectives set forth for this work, a novel framework and formal tools for *model-based system safety*, which I also term *safety supervisory control framework*[1], is developed and presented in the thesis.

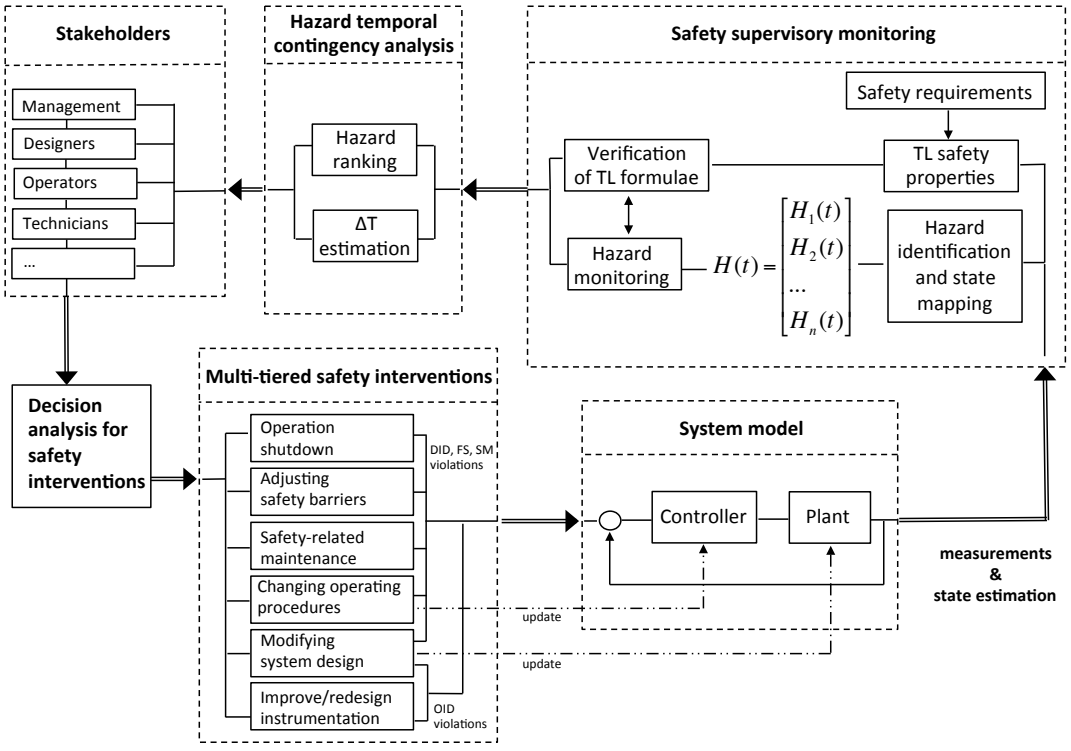

Figure 1.2 Overview of the proposed framework

[1] Appendix A provides a comparison of the proposed approach to classical human supervisory control.

The proposed approach has two fundamental ingredients: (1) the use of state-space models and state variables (from Control Theory) to capture the dynamics of hazard escalation, and to both model and monitor "danger indices" in a system; and (2) the adoption of Temporal Logic (from Computer Science and Software Engineering) to model and verify system safety properties (or their violations, hence identify vulnerabilities in a system). The integrated framework is shown in Figure 1.2 and its ingredients are analyzed in detail in chapters 3 and 4.

The framework and analytical tools here developed are grounded in Control Theory and make use of the state-space representation in modeling dynamical systems. The use of state variables allows the definition of metrics for accident escalation, termed hazard levels or danger indices (used interchangeably hereafter), which measure the "proximity" of the system to adverse events. Furthermore, the adoption of state-space formalism, as will be shown in detail in chapter 3, allows the estimation of the times at which critical thresholds for the hazard level are (b)reached. This estimation process provides important prognostic information and produces a proxy for a time-to-accident metric or advance notice for an impending adverse event. The hazard levels and the time-to-accident metrics create a portfolio of hazard coordinates that can then be displayed dynamically in a "hazard temporal contingency map" to support operators' situational awareness and help them prioritize attention and defensive resources for accident prevention. The idea and capability of measuring the proximity to a performance goal is essential for proper control of a system—this is a fundamental concept in Control Theory. By extension, the ability to measure the proximity of a system to adverse events, proximity in the form of hazard levels or danger indices, is crucial for accident prevention and sustainment of system safety. It also makes for improved dynamic risk assessment and management.

The monitoring of hazard levels and the estimation of the time window available for safety interventions provide important feedback for various stakeholders

and decision-makers 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).

These capabilities are furthermore extended by the adoption of Temporal Logic (TL) in support of the hazard level monitoring effort. The use of TL in risk assessment and safety issues offers many possibilities for overcoming some of the limitations associated with traditional Probabilistic Risk Assessment (PRA), for example in accounting for time-related considerations in accident scenarios and in handling software issues (details are presented in Chapter 2). Temporal Logic is here employed (among other things) for the specification of safety properties that act as constraint on the system behavior. The verification of whether the system satisfies a given property or not provides an important feedback in regards to missing/inadequate safety features embedded in the system and has a fundamental role for informing system design in the early development stages.

The integrated approach of Figure 1.2 augments the current perspective in traditional risk assessment and its reliance on probabilities as the fundamental modeling ingredient with the notion of temporal contingency, a novel dimension here proposed by which hazards are dynamically prioritized and ranked based on the temporal vicinity of their associated accident(s) to being released. It is hoped that this work helps to expand the basis of risk assessment beyond its reliance on probabilistic tools, and that it serves to enrich the intellectual toolkit of risk researchers and safety professionals.

1.3 Presentation Plan

The remainder of this thesis is structured in the following way.

Chapter 2 presents a literature review of the state-of-the-art for risk assessment, and details the limitations mentioned in Section 1.1 together with the workarounds

proposed by the risk and safety community. Those serve both as motivation and as basis for the work presented here.

The following Chapters 3, 4, and 5 constitute the core of the thesis and tackle in detail the development of the framework of Figure 1.2. Specifically: Chapter 3 presents the first ingredient, i.e., the model-based hazard monitoring process enabled by the state-space formalism and the prognostic dimension of temporal contingency for prioritizing safety interventions; Chapter 4 presents the second ingredient, i.e., the use of TL for the expression of safety properties that act as constraints on the system behavior and in support of the hazard monitoring process.

These two ingredients should not be considered in isolation. In fact each of them informs the other, and their integration is tackled in Chapter 5, which also provides a detailed application used as "proof-of-concept" for the proposed framework. Figure 1.3 provides a schematic representation of how these three "core chapters" fit together.

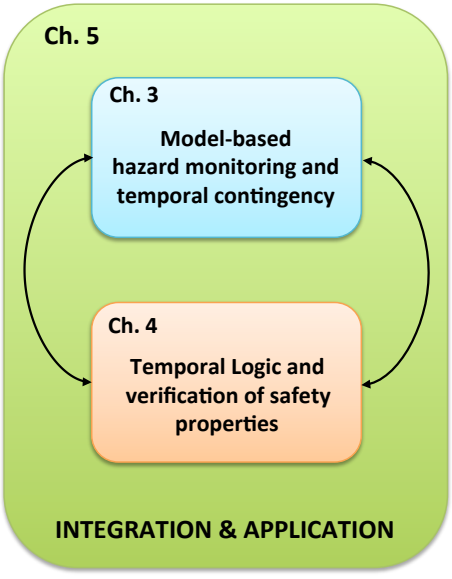

Figure 1.3 Schematic representation of "core chapters" structure

Finally, Chapter 6 concludes this work with a summary of the contributions and the presentation of future work opportunities to expand the presented framework.

CHAPTER 2

LITERATURE REVIEW AND STATE-OF-THE-ART

This chapter presents an organized overview of the historical development and the state-of-the-art techniques for risk assessment. The seminal works that have shaped the thinking about risk in hazardous industries are presented in Section 2.1. Among those, Probabilistic Risk Assessment (PRA) stands out as the first analytical and quantitative approach, which is nowadays widely adopted in industry. Section 2.2 thus tackles the presentation of this important approach together with the challenges and limitations that continue to occupy researchers. The workarounds and novel approaches that were born in response to such shortcomings are presented in Sections 2.3 and 2.4.

2.1 Seminal Works and Historical Development

Different academics and professionals communities have grappled with the multi-disciplinary issues of system safety and accident causation, including psychologists, sociologists, engineers, and management/organizational scientists. The literature on accident causation and system safety is extensive but fragmented, and it is strongly intertwined with the concepts and tools of risk analysis on the one hand, and accident models on the other hand.

Four seminal works shape the current thinking about accident causation and system safety, as well as the analytical handling of risk analysis [Saleh et al., 2010]:

- *Turner's Man-Made Disasters* [Turner 1978]: this work is one of the first scholarly accounts of industrial accidents not as fatalistic "sudden Acts of God" but as events that can be carefully analyzed. Turner's three seminal

contributions were: (1) man-made disasters are a particular class of events that have common patterns in their making, and they can be analyzed to improve future systems' safety; (2) accidents are in the making over long incubation periods, their causes extend deep into the past as a "chain of discrepant events [often] develop and accumulate unnoticed" until an accident is released; and (3) most importantly, accidents cannot be ascribed to purely technical problems, and management and organizational matters are key contributors to accidents. While the specifics of Turner's work may be in part obsolete today, its key theoretical insights have an enduring value and provide much of the conceptual foundations for an extensive literature that followed in its wake on accident causation and system safety.

- *Perrow's Normal Accidents* [Perrow, 1984]: the premise of this work is that in some systems characterized by "interactive complexity and tight coupling" among its components, accidents cannot be foreseen or prevented; they are "normal" and unavoidable. For example, Perrow's study of the Three Mile Island accident led him to consider it "unexpected, incomprehensible, uncontrollable and unavoidable; such accidents had occurred before in nuclear plants, and would occur again, regardless of how well they were run." Hopkins [2001] described this pessimistic conclusion as "an unashamedly technological determinism." Perrow's work remains influential to date. Despite its importance, Normal Accident has been criticized for its oversimplification or lack of understanding of technical and operational choices (Perrow being a sociologist). Normal Accident has a distinct sociological focus. One fundamental objection to Normal Accident is that it does not help make better risk-informed design or operational choices; it only advances the argument that in some systems, accidents are inevitable.

- *The work on High Reliability Organizations (HRO)* [Roberts, 1990a,b]: this collective work, originally started by Karlene Roberts at Berkeley, empirically examines what successful organizations do – how they organize and manage hazardous systems and processes – to promote and ensure system safety. This is a different (and, in a sense, opposite) mindset from what has been described as Perrow's pessimistic contribution to the safety community. HRO studies analyzed operations of US aircraft carriers, air traffic control, electric power distributions, and firefighting. These "organizations" constituted the initial basis of what came to be called High Reliability Organizations, and their practices became the benchmarks and best practices for handling risk and supporting safety in hazardous industries [Saleh et al., 2010]. While some objections exist on the definition of what a "high reliability" organization is, the contributions of this work should not be underestimated, as important advancements were made in identifying managerial and organizational issues affecting system safety.
- *Probabilistic Risk Assessment (PRA)* [Rasmussen, 1975]: The ideas and contributions to accident causation and system safety discussed previously lacked an analytical dimension and connections with the technicalities of the system under consideration, its configuration and operational characteristics. These were provided by the development of Probabilistic Risk Assessment. In a parallel track to the history of ideas pertaining to accidents and system safety discussed previously, a major study was published in 1975 in which the formal PRA technique was introduced and applied to nuclear reactors. The study developed a framework and a set of analytical tools under the broad heading of PRA for assessing accident scenarios and risks in complex systems. PRA is an event-driven framework and technique; it is based on the idea of a stochastic chain of events leading to an accident and starting with an undesirable initiating

event then progressing through various "risk scenarios" until the negative final outcome is reached. Most of the qualitative and quantitative tools that are well-established and used nowadays gained success from the preliminary work on PRA, so that in many occasions the PRA approach has become a synonym of the entire risk analysis field. These tools include the use of risk matrices, failure modes and effects analysis (FMEA) tables, event trees (ET) and fault trees (FT) analysis, and reliability block diagrams (RBD) to mention a few. Given the predominance of PRA in the current toolkit of safety practitioners and its traditional importance, some of the capabilities of the framework here presented are compared and considered as a complementary perspective to this approach.

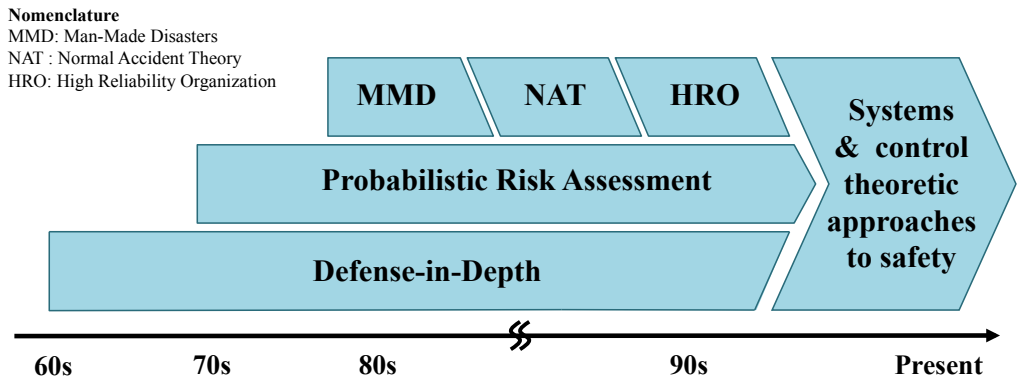

Figure 2.1 Evolution in the development of system safety approaches. Adapted from [Saleh et al., 2010]

The evolution of ideas and approaches in the past fifty years has revolved around three major tracks, highlighted in Figure 2.1. In addition to the seminal works previously mentioned, Figure 2.1 shows in a separate track the basis for risk-informed decisions originally conceptualized by the U.S. Nuclear Regulatory Commission, namely defense-in-depth. In its bare essence, defense-in-depth consists in the design and implementation of multiple safety barriers, technical, procedural, and organizational, and whose objective is first to prevent accident initiating events from occurring, second to block accident sequences from escalating, and third to mitigate

adverse consequences should the previous barriers fail. Accidents typically result from the absence, inadequacy, or breach of such defenses [Rasmussen, 1997; Svedung & Rasmussen, 2002].

Recent contributions are emerging as foundational for the thinking about system safety and accident causation in the framework of System and Control Theory, including the works by Reason [1997], Hollnagel [2004], Rasmussen [1997], and the distinct contributions by Leveson [1995] and [2004].

The control perspective on system safety mirrors the fact that accidents typically result from the absence or breach of defenses, technical and organizational safety barriers, or in this case from a violation of safety constraints [Leveson, 2004]. Conversely, system safety in this perspective is conceived to result from the establishment of safety barriers and enforcement of safety constraints (again of technical and organizational nature). Within this perspective, Leveson proposed the following three major "control factors" in accident causation: (1) inadequate enforcement of safety constraints, (2) inadequate execution of control actions, and (3) inadequate feedback. The explicit articulation of the control perspective identifies decision-makers as the "controllers" and discusses the attributes of controllers for properly handling of hazardous processes, including their competence, or "formal knowledge, heuristic, and practical skills […] to determine whether [they] can make the appropriate risk management decisions [for] a coherent safety control function" as well as their incentives and commitment to safety [Rasmussen, 1997]. Rasmussen's work, although informal in his treatment of a broad range of safety-related topics, remains highly influential to date and appears to be the intellectual foundation on which Leveson [2004] expanded and built the "Systems-Theoretic Accident Model and Processes" or STAMP model for accident causation and system safety. Leveson emphasizes safety *constraints*, rather than *events*, as the most basic concept in accident

analysis, and she highlights ways in which inadequate control of these constraints lead to their violation and subsequently to accidents.

The work carried out for this thesis falls under the last heading in Figure 2.1, namely "Controls and systems theoretic approaches to safety." It follows in the same spirit as some of the works mentioned previously, especially those by Rasmussen and Leveson. It expands on these works in some ways and departs from them in others, particularly in its clear departure from the analysis tools of PRA (e.g., fault trees) and in the adoption of tools from the actual discipline of Control Theory and of concepts derived from Computer Science.

2.2 PRA as State-of-the-Art: Current Challenges and Limitations

As previously mentioned, Probabilistic Risk Assessment (PRA) is a staple in the engineering risk community, and it has become to some extent synonymous with the entire quantitative risk assessment undertaking[2]. Since the 1970s PRA has gained popularity and it is widely adopted beyond the nuclear industry context, in many hazardous industries like the chemical, the oil and gas, and the aerospace ones.

2.2.1 The PRA Workflow

PRA tackles three important questions related to the entire risk analysis field, namely [Apostolakis, 2004; Kaplan & Garrick, 1981]:

1) What can go wrong?
2) How likely is it?

---

[2] Given its importance and its widespread use in industry, the framework presented in this work is compared in its capabilities and limitations to the more traditional approaches of PRA. A dedicated Appendix is provided to show a direct comparison of benefits and limitations of the proposed approach with more traditional tools. This comparison is presented in the form of a summarizing table, which lists important capabilities associated to risk assessment.

3) What would be the consequences?

Traditional (or static) probabilistic risk assessment (PRA) is aimed at providing an answer to these three questions roughly by means of the following workflow (adapted from [Apostolakis, 2004]):

- First, possible risks are identified through any hazard evaluation procedure (generally in the form of a structured brainstorming effort based on the system design and on past experience with similar plants). Risk analysis is at its core the imagination of failure [Saleh et al., 2014b], and this first step provides the analyst with a set of undesirable end states (the consequences of the third question) that are then traced back to the initiating events that bring the system from its nominal state of operations to off-nominal states.
- Secondly, Event Trees and Fault Trees are employed to generate accident scenarios. These two techniques, which employ standard logic (inductive and deductive logics with the use of the Boolean operators AND "$\wedge$" and OR "$\vee$"), identify the sequences of events that lead from possible initiating events to the undesired end state. These logical diagrams are necessary for both the qualitative and the quantitative evaluation of the risk associated to each accident scenario.
- Finally, the probability associated to each accident scenario is computed and the scenarios are ranked based on their expected frequency of occurrence [Apostolakis, 2004]. The probabilities associated which each event transition that can lead from the initiating event to the end state are computed by means of empirical data on the process and of expert opinion. Because of the sparsity of

relevant empirical data [Mosleh et al., 1988], some degree of expert opinion is often a fundamental pre-requisite of any PRA effort.

The results obtained from the application of PRA support and provide insight for any risk-informed decision and regulation. PRA modeling can be a massive task, and it is generally done by abstracting and clustering the events under consideration into classes and categories [Mosleh, 2014]. A certain familiarity with PRA tools is required for anyone interested in risk analysis and safety issues, and a vast literature is available together with specialized software tools for its application.

2.2.2 Open Challenges and Limitations

Despite its appeal, PRA is not without its flaws, and in recent years researchers have highlighted some of its limitations and proposed several improvements [Aldemir, 2013; Mosleh, 2014; Zio, 2014]. Those can be summarized as follows:

- *Timing and ordering considerations*: the static logic models used in traditional PRA are insensitive to dynamic process failures. For instance, when multiple top events are considered in a fault tree, "the actual final state of a [truly] dynamic scenario depends on the order, timing, and magnitude of the component failure events" [Zio, 2014], which traditional fault trees cannot capture. Similar arguments can be made with Event Trees. Given the scenario postulated and tested by the analyst, the order of occurrence of the failure events is pre-set resulting in potential vulnerable sequences that remain untested [Aldemir, 2013; Zio, 2014]. Additionally, recovery and other time-dependent performances cannot be combined into the static traditional PRA tools.
- *The inclusion of software failures*: the issue is the understanding of how software failures will affect the overall system, and how to include these

considerations in risk assessment. Arguably, PRA has been "very much hardware oriented" [Mosleh, 2014]. A gap exists in current methods to account for software failures or contributions to accidents [Leveson, 1995; Favarò et al., 2013] and model them with tools that are compatible with traditional PRA [Apostolakis, 2004; Kirschenbaum et al., 2009]. The need to leverage new tools and perspectives for software safety analysis has been argued by several authors [DOD, 2012; Zio, 2014; Mosleh, 2014].

- *The issue of human response*: Human Reliability Analysis (HRA) is used to estimate the quantitative and qualitative contributions of human performance to the overall system reliability. Current approaches to HRA interface well with traditional PRA tools and are included in many risk assessment efforts [Swain, 1990]. However, HRA does not account for modeling human response *during* the unfolding of an accident scenario [Mosleh, 2014]. Real-time analysis and/or simulation of the human performance are of paramount importance to estimate how operators' actions affect the estimated frequencies of failure events (and hence the risk estimation). Additionally, the study of human response has close ties with the analysis of instrumentation design and layout, which provide the operator a feedback on the system status. New tools and approaches can aim at relating potential operator performance degradation to ineffective instrumentation layouts, with missing information or misleading interpretations of the signals coming from the plant during an accident unfolding.
- *The inclusion of physical models*: current PRA tools have limited capacity for the integration of physical phenomena models (e.g., physics of failure, environment physics description) [Mosleh, 2014]. Whenever the physics behind failure mechanisms can be included, many restrictions are required in terms of simplification. This issue is related to the lack of a mathematical framework that can integrate and handle different domains of analysis at the same time.

- *The issue of completeness*: PRA is executed by means of abstracting events into classes and categories. The level of abstraction is dependent on the type of decisions that PRA is meant to support, the resources allocated to the PRA effort, and the state of the knowledge regarding the system [Mosleh, 2014]. Related to the level of clustering of the events and to the abstraction effort is the issue of how complete the PRA process is in terms of breath of coverage of the risk and accident scenarios, their depth of causality, and the fidelity in the definition of the basic events of the fault and event trees and their associated probabilities [Mosleh, 2014]. Additionally, the scenarios that are tested are postulated, hence developed a priori, by the analyst, so that traditional PRA cannot by itself discover new accident scenarios.

New techniques and approaches are currently under investigation to address some of these limitations. A brief overview of some of the most notable approaches follows. I divide them in two categories: those that fall under the heading of Dynamic PRA, and those that rely on the addition of time-properties and time-related considerations. The proposed approach falls at the intersection of the two as it combines aspects that are common to both categories, as will be presented in the next chapter.

2.3 DPRA: an Answer to the Time-Dependency Limitations of PRA

Dynamic PRA comprises a set of simulation-based methods that combine deterministic and probabilistic approaches to account for the time-dependency of the events they try to model. For this reason, DPRA tools also go under the name of Integrated Deterministic and Probabilistic Safety Analysis (IDPSA) [Zio, 2014]. DPRA can handle both continuous and discrete time, as well as hybrid systems, depending on the system model of choice [Aldemir, 2013]. Regardless of the method of choice, three basic inputs are needed for DPRA:

1) A time-dependent physical model of the system dynamics;
2) A list of identified normal and abnormal system configurations;
3) The transition probabilities among the normal and abnormal configurations (or a more complex model of the stochastic rules that govern each transition).

Kirschenbaum et al. [2009], Aldemir [2013], and Zio [2014] provide a survey of different DPRA methodologies including dynamic flowgraph methods, Markov/cell-to-cell mapping techniques, and Petri nets. These techniques have the potential to uncover and identify plant vulnerabilities that were a-priori unknown, and that could not be considered with traditional PRA tools. DPRA enlarges the exploration of the possible accident scenarios space, by including ordering and timing of events [Zio, 2014]. Moreover, simulation-based approaches can provide insight into an accident phenomenology and its causal basis for different accident scenarios [Mosleh, 2014]. This is due to the fact that the sequencing of events is no longer pre-determined by the analyst, but derives from the stochastic simulation itself.

DPRA is not an alternative to the traditional PRA, but rather complementary. Traditional PRA is still used in conjunction with the more sophisticated, but more complex, simulations carried out in DPRA. On one hand, DPRA provides additional insight for complex systems; on the other hand, PRA provides a technique popular for its simplicity and clarity in communicating the results of risk assessments [Aldemir, 2013]. Limitations and drawbacks of DPRA include:

- Substantial efforts are needed to generate the data for the transition probabilities among the different configurations. Although DPRA seeks to reduce the need for expert judgment, expert opinions are still required. Additionally, the model input data is not always readily available, so that experimental testing and/or

components simulation may be required for the computation of the stochastic rules that regulate the system transitions.

- The development of the system models can be computationally intensive. Many of the available modeling tools suffer from the number of states explosion problem, and the size of the system under consideration is limited by the current computational capabilities (see [Zio, 2014] for a discussion on possible solutions).
- DPRA is a simulation-based methodology. This implies that the verification effort is never completely exhaustive (i.e., not all possible existing scenarios are tested). Generally, dominant-risk scenarios are given higher priority, but completeness of the testing effort is not guaranteed.
- There are difficulties with the output post-processing and with the classification of the various accident scenarios generated by the tools (e.g., problems with clustering of scenarios by similarity of the event sequences and/or the end state of the system). The classification of the scenario is related to the capability of recognizing unanticipated scenarios [Zio, 2014]. Additional concerns regard what kind of output to generate for risk communication and how to organize and communicate the data produced by the simulation in a clear manner.

DPRA is still far from being broadly adopted as an industrial practice, and considerable research remains underway in this field. Benchmark examples continue to be developed for the consistent comparison of the different risk assessment methodologies [Kirschenbaum et al., 2009; Aldemir et al., 2010].

2.4 Current Approaches Involving the Use of Temporal Properties

Dynamic PRA has not been the only answer provided by the risk and safety community to the previously highlighted limitations of its static counterpart. With the

increasing importance of digital systems, frameworks that leverage approaches derived from and inspired by those used in computer science have also surfaced in the academic literature. These approaches introduce formal languages for the definition of temporal properties to be used in conjunction with the tools of static PRA. Some notable works in this area is briefly reviewed here.

- *Fault Trees (FT) temporal extensions*: I denote under this heading works aimed at extending the classical fault tree analysis within traditional PRA. Notable contributions in this regard have been made by [Hansen et al., 1998], [Palshikar, 2002], and [Magott and Skrobanek, 2012]. The FT temporal extensions were achieved in several ways:
  - By adding temporal gates to the standard FT notation: in this approach new gates were added to the standard pool of static logical gates (e.g., AND, OR, XOR gates). The new gates activation is dependent on the particular sequence or duration of the events that are fed into them. For instance, the "Priority AND" gate requires the ordered occurrence of the events fed into it from left to right; the "For all t instants" gate requires that the events fed into it hold for t instants of time, basically translating into an AND gate of an event holding at t_1, AND again holding at t_2, AND so on up to time t. Additional examples of temporal gates can be found in [Hansen et al., 1998; Palshikar, 2002].
  - By adding time dependency in the events definition: rather than adding a temporal dependency inside the gate logic (as it was done for the temporal gates of the previous examples), this approach includes time-related considerations inside the definition of the events that are then connected by static logical gates. A common way of doing this is by adding to each event description a duration interval. The duration

interval in turn affects the applicability of the logical gate. For instance, it can prevent an AND gate from being activated unless a minimal duration time for the event is achieved.

Although not always explicitly stated, these extensions make use of temporal logic (or some rudimentary form of it). In general, they do not require the user to be familiar with the formalities of TL and the techniques for the verification of TL properties. This makes their use simple and approachable, but it hinders the user from tapping into and benefitting from the full potential of these techniques.

- *Formal logics for the analysis of time-critical systems*: these approaches introduce timed logics for the explicit expression of time-dependent considerations. Timed logics add temporal operators to the pool of classical operators from propositional and predicate logics. Different logics can be used for this purpose such as probabilistic computational tree logic [Johnson, 1995] or real-time logic [Jahanian and Mok, 1986, 1994]. The use of timed logics allows to reason about the ordering and timing of events and to specify the desired dynamical behavior of the system. Specifically, timed logics are used to express time-dependent system requirements and performance constraints. These approaches no longer make use of traditional PRA. They require a model for the system under consideration, such as in DPRA. Contributions that resort to timed logics for system properties specification are somewhat more infrequent in the literature when compared to the above-mentioned approaches that extend traditional PRA tools. A separate mention should be given to applications of timed logics to problems that are not strictly related to risk assessment, but still span safety applications. This is the case for instance of Johnson's work on the use of formal methods for accident investigations [Johnson, 2000], or the application of temporal logic to support human factors engineering [Johnson and Harrison, 1992].

The problem of including time considerations in the risk assessment process is tackled in the proposed approach by two complementary features: on the one side, the hazard levels or danger indices model dynamical quantities that enable, through the estimation of the time-to-accident metric, to account for time-dependent considerations for safety interventions; on the other side, the use of Temporal Logic (TL) allows the explicit inclusion of temporal ordering (through the use of TL operators) within the definition of safety properties that act as constraints for the dynamic behavior of the system [Favarò and Saleh, 2016a,b]. These two ingredients are separately introduced in the next chapters. Specifically, Chapter 3 tackles model-based hazard monitoring and presents an overview of the safety supervisory control framework; Chapter 4 tackles the adoption of TL to bear on risk assessment and system safety, and presents the formulation of illustrative TL safety properties. Chapter 5 will tackle their integration in detail.

CHAPTER 3

SAFETY SUPERVISORY CONTROL FRAMEWORK AND MODEL-BASED HAZARD MONITORING

This chapter focuses on the ingredients of model-based system safety, the associated modeling of danger indices and hazard equations, and the creation of a hazard temporal contingency map. The integration of all these elements with the ingredient of Temporal Logic and safety properties verification (which is presented in detail in chapter 4) constitutes the novel safety supervisory control framework presented in Figure 1.2. The chapter is organized as follows. Section 3.1 presents in more detail the framework proposed, and analyzes the workflow for its adoption. Section 3.2 tackles the first step of the approach, i.e., the model development based on the state-space representation formalism. Section 3.3 introduces the notion of hazard level (or danger index) with detailed analytical examples related to its monitoring process. Section 3.4 concludes the chapter with the presentation of the hazard-temporal contingency mapping.

3.1 The Safety Supervisory Control Framework

The framework proposed in this work adopts a model-based approach and state variables to capture the dynamics of hazard escalation and to monitor "danger indices" in the system. The identification and quantification of indices of proximity to adverse events supports the development of a safety supervisory control approach (shown in Figure 1.2 and reported here in Figure 3.1 for convenience), and it is particularly helpful for triggering pre-emptive safety interventions and improving accident prevention, as argued shortly. The continuous monitoring of the hazard level and the

estimation of the time-to-accident metric provide important feedback for various stakeholders, from management and designers, to front-line operators and technicians, to guide safety interventions over different time scales. The monitoring of the distinctive macro-state variables "hazard levels" during system operation (i.e., on-line) provides important feedback for operators to recognize a developing adverse situation, prioritize attention, and allocate defensive resources for safety interventions and hazard de-escalation. Additionally, the off-line application of the safety supervisory process can assist in checking the presence/adequacy of safety features implemented in the system, providing an important feedback during the design stages.

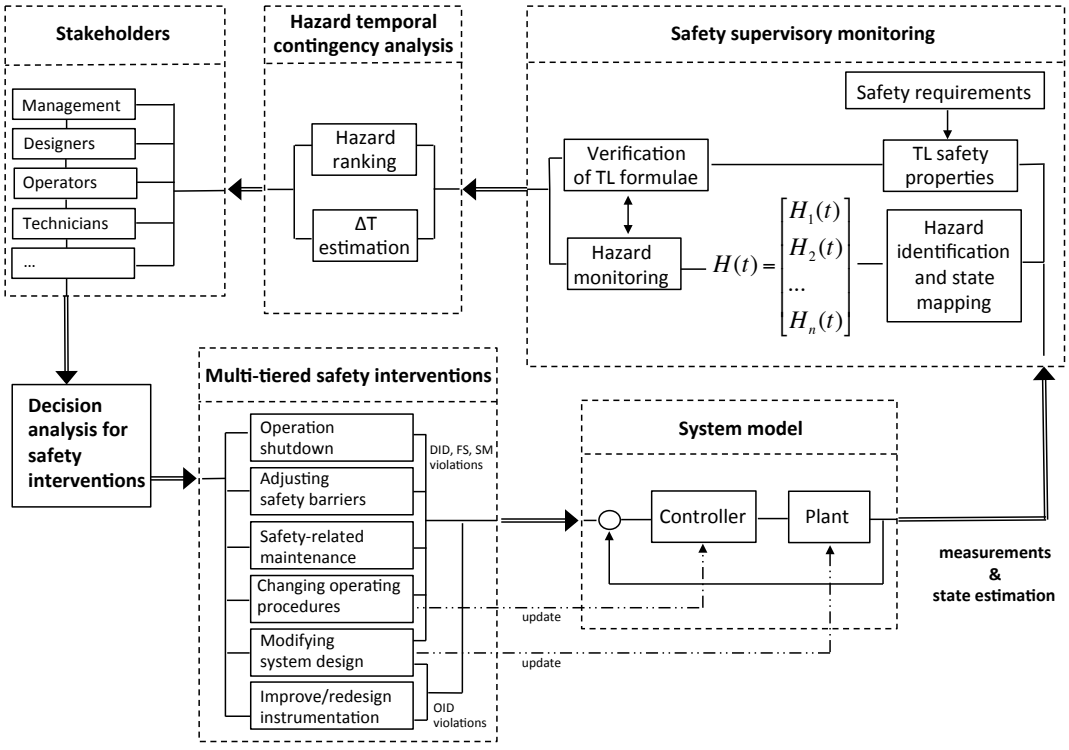

Figure 3.1 Overview of model-based safety supervisory control, and dynamic hazard monitoring for safety interventions (not meant to be exhaustive; several loops and blocks are not shown to avoid clutter)

The elements of Figure 3.1 are described in detail next, together with the explanation of the "workflow" behind the application of the approach.

The *system model* block in the bottom right corner includes the state-space model of the system under consideration (the "plant" block) and a "system controller" within an inner loop (e.g., a digital controller, a human controller, or a combination of both for varying degrees of automation). The controller provides inputs to the plant, seeking to ensure that the system fulfills its performance requirements (to behave "as expected") and to steer it away from off-nominal hazardous conditions. The reference inputs to the system model block (the input to the comparator upstream the controller box in Figure 3.1) depend on performance and production requirements, as well as the safety requirements and constraints.

Output measurements and system state estimations are undertaken downstream of the "*system model*" (not shown in a separate "*observer*" block so as not to further clutter the figure). These measurements and state estimation are fed into the macro-block entitled *safety supervisory monitoring* where several functions are performed:

i. *Hazard identification and state mapping*: The hazard level or danger index is an analytic metric for capturing accident escalation, and it reflects the proximity of the system to a particular adverse event (details in 3.3). This block identifies the hazards of interest and maps them into (a subset of) the state variables of the system. The hazard level depends on the system state variables, and this block provides the model and connection between these two analytical concepts. Examples of the mathematical equations that represent the mapping between system states and hazard levels are presented in Section 3.3. Multiple hazard levels (for various risks) are considered at a time, and they are reflected in the vector output *H(t)* of the Hazard Identification block in Figure 3.1.

ii. *Hazard level monitoring*: Once the hazard level metrics are defined, they are to be continuously monitored (either by the operator or through an automated

process). Monitoring the values and trends of the hazard levels is an important step for the prioritization and triggering of safety interventions.

iii. *TL expression of safety properties* (chapter 4): This step accounts for the translation of system safety requirements into TL formulae. These formulae in turn act as constraints on the system behavior. Each safety property is predicated on a particular hazard level function, making the monitoring of the hazard level a fundamental step for the definition and verification of each TL formula.

iv. *Verification of the TL safety properties* (chapters 4 and 5): The TL safety properties are continuously checked for compliance/violation. The violation of a safety constraint provides diagnostic information for re-engineering the system design, the safety barriers layout, and the system instrumentation, to mention a few possible types of safety interventions that can be triggered by this verification.

The end-objective of the safety supervisory block is to support decision-making, especially in relation to safety interventions on the system, and to improve accident prevention. The functions and blocks discussed in (i–iv) are some of the means for contributing to this end-objective. One particularly important tool in support of this end is shown in Figure 3.1 downstream the safety supervisory block and is entitled *Hazard temporal contingency analysis (and map)*. This block is both an analysis and visualization tool: it dynamically assesses and displays the "coordinates" of hazards in a system to support operators' sensemaking and help them prioritize attention and defensive resources for accident prevention. The coordinates of hazards include the hazard level or danger index (how hazardous a particular situation is) and an estimated time-to-accident metric (how much time is left before the accident associated with a particular hazard is released if no changes are made to the system operation). The hazard temporal contingency map provides prognostic information regarding the time-

window available for operators to intervene before a hazardous situation becomes unrecoverable, and in so doing it helps prioritize risks and hazards based on their temporal contingency, not based on probability, or some combination of probability and consequence, as is traditionally done in PRA.

This safety supervisory and hazard temporal contingency blocks are not only helpful for system operators, they also affect various *stakeholders* involved in the safety value chain[3] of the system. The outer loop in Figure 3.1 closes back on the system by providing the hazard information (dynamics/trends) to different stakeholders, and prompts them to assess the need for and trigger *multi-tiered safety interventions*. These interventions can range from immediate actions (e.g., emergency shutdown, adjustment of safety barriers, or safety-related maintenance) to off-line re-engineering of safety features in the system design, the system instrumentation, or the operating procedures for example. These changes affect the *system model* block both in terms of the plant description (state space model) and of the controller definition and operations, thus closing the outer feedback loop in Figure 3.1. The model-based safety supervisory control and the hazard temporal contingency map support safety interventions over different time-scales and by different agents in the safety value chain. Finally as noted in the caption, Figure 3.1 is not meant to be exhaustive; several blocks and additional feedback loops are not displayed to avoid visual clutter (for example the "observer and state estimation block, and the feedback loops for monitoring the effectiveness of the safety interventions).

These considerations are revisited in detail in the next subsections. Three key steps can be highlighted in the process here described and are analyzed next:

---

[3] The safety value chain consists of individuals or groups who contribute to accident prevention and sustainment of system safety. It includes operators, technicians, engineers, system designers, managers and executives, regulators, safety inspectors, and accident investigators, individuals who affect and contribute to system safety over different time-scales [Saleh et al., 2010].

1. **System model development**: development of a mathematical model for the dynamical system under consideration (or for a subset of the system with the safety implications of interest), with identification of state variables and state-space representation (Section 3.2);
2. **Safety supervisory monitoring**: identification of the hazard levels or danger indices of interest and state mapping, along with the continuous monitoring of these indices (Section 3.2);
3. **Hazard temporal contingency analysis (and map)** to guide safety interventions: estimation of the time-to-accident metric and development of the hazard temporal contingency map, for ranking and prioritization of safety interventions (Section 3.4).

3.2 Model Development

The creation of a model for the dynamical system under consideration constitutes the first step of the approach. Similar to Hansen's [1998] work on the extension of temporal fault trees, the framework here proposed makes use of state variables (either continuous or discrete) denoting functions of time, as in Modern Control Theory. As highlighted in [Cowlagi and Saleh, 2013], although well-established safety strategies such as defense-in-depth "reflect an implicit recognition of accident prevention as a control problem" and several authors have articulated and developed this recognition more explicitly [Rasmussen, 1997; Leveson, 2004], actual control theory has been to a large extent absent from the discussion of safety as a control problem. In the following, I make use of some tools from the actual Control discipline, in particular with references to state-space representations of dynamical systems and state estimation. Given the need of a system model, the proposed framework falls under the category of model-based safety analysis, which is briefly

reviewed next. Afterwards, the state-space formalisms is introduced together with an example model.

3.2.1 Model-Based Safety Analysis

A dynamical system is one whose properties (or a subset of them) change with time. A model for such system is defined as a set of equations that represent its behavior in time. Once an analytical model is developed, it can be translated/imported into a simulation environment for various types of analyses.

In simple terms, whenever a mathematical model is developed and employed for the analysis of the system under consideration (instead of carrying out experiments on the actual system), the approach is referred to as a model-based analysis. Specifically, model-based *safety* analysis has gained popularity over the past decade. It was first introduced to provide a more formal approach for analysis techniques that had traditionally been performed manually, with a low likelihood of being complete, consistent and error-free [Joshi and Heimdahl, 2005].

The main benefit of model-based analysis is the possibility of interfacing the system model(s) with automated analysis tools that can analyze the system behavior, allowing the verification of different aspects of fault tolerance and potentially the auto-generation of different outputs (e.g., fault trees) [Joshi and Heimdahl, 2005], and the repeatability of the analyses.

Many of the early efforts in model-based safety analysis were aimed at the auto-generation of PRA-types of analysis (see for instance [Papadopoulos et al., 2001]). The interest then expanded towards automated fault-detection and diagnosis [Isermann, 2005], and to the introduction of formal verification of the models to improve system reliability [Bozzano and Villafiorita, 2003; Bozzano et al., 2003]. The need for novel model-based techniques was justified by the increasing complexity of the systems under consideration, and by the need of safety engineers to assess the system behavior in

degraded situations without the need to manually develop for example an extensive set of fault trees [Bozzano and Villafiorita, 2003].

In the following, I build on ideas from model-based safety analysis, with the distinction that they are not employed in conjunction with PRA tools and techniques, but for the proposed safety supervisory control approach, providing operators and other stakeholders with the means to continuously monitor the system for hazardous conditions and scan for potential unfolding adverse events. The following approach leverages the state-space representation formalism, which is briefly reviewed next.

3.2.2 State-Space Representation

The state-space representation is a mathematical formalism widely used in Modern Control Theory. It is concerned with three types of variables (all functions of time): *input* variables (denoted by the vector $\mathbf{u}(t)$), *output* variables (denoted by the vector $\mathbf{y}(t)$), and *state* variables (denoted by the vector $\mathbf{x}(t)$). Inputs and outputs are the means by which an external agent can interact with the system: the appropriate control actions are applied through the inputs to ensure the desired system behavior, which in turn is monitored through the output recording and state estimation [Bakolas and Saleh, 2011]. State variables (or simply the "state" of a system) are formally defined as the minimum set of variables that contain all the necessary information of the internal conditions of a system at some time t_0, such that the knowledge of the system state at time t_0 along with the knowledge of the input vector $\mathbf{u}(t)$ for $t \geq t_0$ is sufficient to determine all the system future outputs (for $t \geq t_0$) [Chen, 1995].

A dynamical system can then be represented in terms of its state-space representation through a system of first order differential equations, such as those of Eq. (3.1).

$$\begin{cases} \dot{\mathbf{x}}(t) = F(\mathbf{x}(t), \mathbf{u}(t)) & \text{state equation} \\ \mathbf{y}(t) = G(\mathbf{x}(t), \mathbf{u}(t)) & \text{output equation} \end{cases} \tag{3.1a}$$

F and G are generic functions (linear or non-linear) that relate on the one hand the rate of change of the state to the state itself and the input vector (state equation), and on the other hand the output vector to state vector and input vector (output equation). Equation (3.1a) holds for continuous systems, and can be easily generalized for discrete cases. For the case of linear systems, Eq. (3.1a) assumes the well-known form:

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) \end{cases} \tag{3.1b}$$

where the matrices A, B, C, and D may also be dependent on time.

The role of state variables is central to the discussion, and it will enable the definition of a quantifiable metric for accident escalation, the hazard level function or danger index. The hazards of interest for the system are mapped into (a subset of) the system state variables. A simple example is provided next to clarify some of these concepts and the application of the process described in Figure 3.1.

Figure 3.2 shows a schematic of a cylindrical oil tank, with an incoming mass flow $\dot{m}_{in}(t)$, and mass outflow $\dot{m}_{out}(t)$. Valves in the feeding and in the outflow line regulate the two mass flows. From the perspective of safety supervisory control, the role of the operator is to monitor the condition of the oil tank, and to apply control actions to steer the system away from dangerous situations should they develop. For instance, for a system such as that of Figure 3.2 we may want to ensure that: (i) a certain threshold height of oil inside the tower is never (b)reached or simply that the tower does not overflow; and (ii) that correct instrumentation and alarms are set up to inform the operator of potential problems or escalating hazard level in a timely manner.

Violation of both these considerations let to the explosion at the Texas City refinery in 2005 in which 15 people were killed and 180 injured [Saleh et al., 2014a].

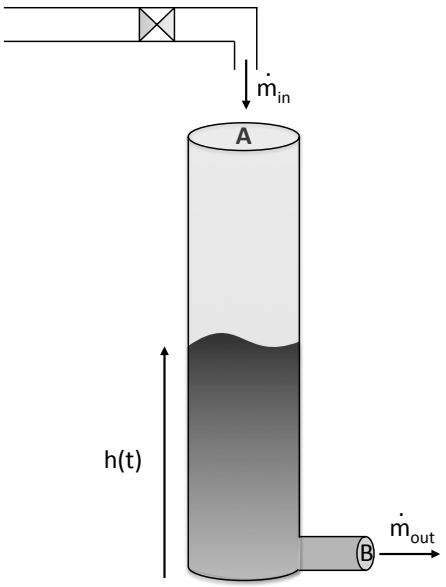

Figure 3.2 Schematic representation of an oil tank

The distinction between the two considerations, (i) and (ii), is subtle but important: the first requires the operator to monitor the height of the oil in the tower (*h(t)* in Figure 3.2). In the proposed framework, it is possible to set up the oil height to be one of the system state variables, and then map it into a hazard level function. Monitoring the current value of the oil height against specific thresholds, and allowing the operators the use of appropriate control actions (e.g., regulating the incoming mass flow, or closing/opening the outflow line) provides the operator with the information needed to satisfy the first property. The second property is instead related to the notion of observability-in-depth and the ability to correctly diagnose the hazard level associated with the system, and is presented in detail in chapter 4.

The model of the system can be set up in the following way. For simplicity, a one-dimensional problem is here considered, with the height of oil inside the tower picked as system state. The model considered two control inputs, given by the incoming

mass flow $\dot{m}_{in}(t)$, and the possibility to open up or close out the outflow line (hence zeroing out the outflow cross-section area, B in Figure 3.2). The output is given by the mass outflow $\dot{m}_{out}(t)$. We have:

$$\left.\begin{cases} x(t) \rightarrow h(t) \\ y(t) \rightarrow \dot{m}_{out}(t) \\ u_1(t) \rightarrow \dot{m}_{in}(t) \\ u_2(t) \rightarrow B(t) \end{cases}\right\} \rightarrow \mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \tag{3.2}$$

To set up a state-space model, the first step is to consider the mathematical model of the physics governing the system under consideration. The mass balance for the tank gives:

$$\frac{dV(t)}{dt} = \frac{1}{\rho}\left[\dot{m}_{in}(t) - \dot{m}_{out}(t)\right] \tag{3.3}$$

with V being the volume of oil filling the tank, and where for simplicity the density of the oil is considered to be a constant ρ. Given a constant cross-sectional area A for the tank, we obtain:

$$\frac{dh(t)}{dt} = \frac{1}{A\rho}\left[\dot{m}_{in}(t) - \dot{m}_{out}(t)\right] \tag{3.4}$$

The outflow can be expressed as

$$\dot{m}_{out}(t) = B(t)\rho\sqrt{2gh(t)} \tag{3.5}$$

where $\sqrt{2gh(t)}$ represents the velocity of the fluid in the outflow pipe assuming a constant acceleration g for the incoming mass flow. The differential equation governing the process is thus obtained:

$$\frac{dh(t)}{dt} = \frac{1}{A\rho}\left[\dot{m}_{in}(t) - B(t)\rho\sqrt{2gh(t)}\right] \tag{3.6}$$

It is now possible to obtain the non-linear state-space representation of the dynamical system of Figure 3.2 by considering the choice of states, outputs, and inputs provided in Eq. (3.2):

$$\begin{cases} \dot{x}(t) = \frac{1}{A\rho}\left[u_1(t) - u_2(t)\rho\sqrt{2g\,x(t)}\right] \\ y(t) = u_2(t)\rho\sqrt{2g\,x(t)} \end{cases} \tag{3.7}$$

The model of Eq. (3.7) is the basis for the application of the safety supervisory monitoring analysis that follows according to Figure 3.1. The state-space representation is a powerful tool for modeling a significantly broad range of dynamical systems. The choice of which variables to select as states of the system is not unique, and in this case is dependent on and informed by the particular hazards to be monitored and safety constraints under consideration. In model-based approaches, the analytical expression of the model (such as that of Eq. (3.7)) is then translated/imported in a simulation environment, and it enables a broad range of uses such as controller design, (hazard) monitoring, and diagnostic.

3.3 Safety Supervisory Monitoring

This section analyzes in detail the safety supervisory monitoring block of Figure 3.1, focusing on the following two steps: (i) the hazard level(s) identification and its mapping into the system state (section 3.3.1), and (ii) the execution of the monitoring process (section 3.3.2). Section 3.3.3 presents the application of the hazard monitoring process to a rejected takeoff scenario, to exemplify its use and capabilities.

3.3.1 Hazard Level Identification and State Mapping

The hazard level, denoted by *H(t)*, can be intuitively conceived of as the closeness of an accident to being released [Saleh et al., 2014a]. Its definition provides an index to quantify "how dangerous" the current system state is, in terms of its proximity to an accident occurrence. In the following the terms hazard level and danger index are used interchangeably.

In order to define the function *H(t)*, the first thing is to specify what accident to monitor against. For instance, in the oil tank example presented in Figure 3.2, monitoring against the accident "loss of containment (LoC) through tower overflow" suggests that a suitable danger index maps the state of the system "oil height *h(t)*" against the maximum height picked as threshold. This is captured by

$$H_{LoC}(t) = \frac{h(t)}{h_{max}} \tag{3.8}$$

where the height of 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".

More generally, a series of adverse events that bring a system from its nominal operational conditions to off-nominal ones and finally to an accident occurrence can be reflected by the dynamics of the hazard level over time (an illustrative example is shown in Figure 3.3). The dynamics of the hazard level is not necessarily monotonic, and it can consist in a sequence of escalation, de-escalation, and constancy phases[4].

---

[4] 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.

Safety interventions are meant to block or de-escalate a hazardous situation (or its hazard level)[5].

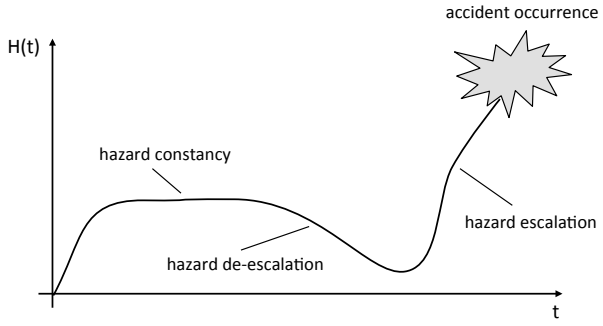

Figure 3.3 Illustration of hazard level dynamics

More complex danger indices can also be devised for the tower example in Figure 3.2, for instance by accounting for the velocity at which the tower is filling up, or by considering multiple states such as pressure (*p(t)*) and temperature (*T(t)*) of the oil. For example, one can set up a limit for maximum temperature inside the tower (T_{max}), where also the temperature change due to changing height and pressure of the oil inside the tower is taken into account[6]:

$$H_T(t) = \frac{T(t)}{T_{max}}\left(1 + \frac{\alpha}{\rho c_p}\dot{p}(t)\,\frac{h(t)}{\dot{h}(t)}\right) \tag{3.9}$$

The idea of introducing a quantitative index for capturing the hazardousness of a situation is not novel. It is well established and particularly useful in the field of human-robot interaction [Ikuta et al., 2003; Kulić and Croft, 2005] where "danger indices" are

[5] Appendix C examines the notion of Agonist, Antagonist and Inverse Agonist that are in a sense responsible for the three dynamic behavior of the hazard level examined in Figure 3.3.

[6] Equation (3.9) assumes an isentropic process. α is the volumetric thermal expansion coefficient, c_p is the constant pressure specific heat capacity.

devised based on the distance between the agents involved (e.g., patient and robot for human care) and the relative velocity to identify situations in which safety is compromised. For instance, in [Kulić and Croft, 2005] expressions of the following kind appear for the definition of danger levels:

"If (Distance = LOW) and (Velocity = HIGH) -> (DANGER = HIGH)"

Similarly to what was done in Eq. (3.8), Ikuta et al. [2003] proposed a danger index α based on the force of a potential impact, compared to a critical impact force, where the force is dependent on the velocity, the distance, the shape, and the mass of the agents involved:

$$\alpha = \frac{\mathrm{F\ (v,d,s,m)}}{\mathrm{F_c}} \tag{3.10}$$

These danger evaluation methods are aimed at establishing quantitative metrics to measure and control the hazardousness of a situation during system operation, and to minimize the danger involved in robot tasks. I propose here their extension beyond the specific field of human-robot interactions. These methods are an important tool in support of accident prevention and for sustaining system safety. Regardless of the specifics of their definition, the notion of quantifiable danger indices is a powerful one, and it adds a real-time dimension to the problem of risk assessment and hazard monitoring; it is also an important piece in the view of safety as a control problem (since accident prevention requires maintaining danger indices within safe bounds).

In the proposed approach the definition of the hazard level is dependent on (a subset of) the state of the system. Equations such as (3.8), (3.9), and (3.10) can be generalized in the case of a N-dimensional state vector by the functional definition:

$$H(t) = f(x_1, x_2, \dots, x_N, t) \tag{3.11}$$

The estimation of the system state enables to measure the proximity to particular adverse events, an important step for accident prevention. Other authors have in the past advocated the need to include state variables dependencies in the notion of risk. For example, according to Haimes [2009] the reason why a universally agreed-upon definition of risk, a complex multidimensional concept, is still lacking is to be found in the missing understanding of some requisite ingredients, such as the state variables of the system. As the "performance capabilities of a system are a function of its state vector" [Haimes, 2009], then by the same token so is the safety or lack thereof and the hazardous condition of the system at any point in time. The notion of a danger index enables one to make explicit this (dynamic) risk dependence on the state vector of the system, and it becomes important to ensure the proper control of the system.

Based on the previous considerations, the proposed model-based approach can augment the system model shown in Eq. (3.1) with an additional *hazard equation*, which captures the dependency of the hazard level on (a subset of) the state vector **x**(t), and of the dynamics of the hazard level on the control variables **u**(t). For the case of linear systems we obtain:

$$\begin{cases} \dot{\mathrm{x}}(\mathrm{t}) = \mathrm{Ax}(\mathrm{t}) + \mathrm{Bu}(\mathrm{t}) & \text{state equation} \\ \mathrm{y}(\mathrm{t}) = \mathrm{Cx}(\mathrm{t}) + \mathrm{Du}(\mathrm{t}) & \text{output equation} \\ \text{----------} & \\ \dot{\mathrm{H}}(\mathrm{t}) = \Phi\mathrm{x}(\mathrm{t}) + \Psi\mathrm{u}(\mathrm{t}) & \text{hazard equation} \end{cases} \tag{3.12}$$

where the matrix Φ derives from the mapping of the hazard level into the state vector and the matrix Ψ embodies the dependence of the hazard level dynamics on the inputs vector. Adjusting the values of the matrix Ψ (whether done "manually" by the operator,

or through an automated controller) results in different control actions on the hazard level. This process, called input shaping, is generally carried out in modern control theory to achieve specific performance goals. In this case, input shaping for the hazard equation allows to control the system and steer it away from dangerous conditions (e.g., de-escalate hazardous levels).

For many years, the guiding principle behind the control synthesis problem was that of output feedback (i.e., the observation of the system output). After the seminal works of Kalman [1960] and Bellman [1957], it became evident that the selection of control inputs is more efficient when based on the knowledge of the actual internal state of the system, rather than on its output [Bakolas and Saleh, 2011]. This consideration is reflected in the proposed approach in the mapping of the state vector into the hazard level, and thus in the fundamental role of state estimation to capture the dynamics of the danger indices. The process of hazard monitoring is thus a form of state estimation, and it provides the proper feedback upon which to base control actions for safety interventions.

A final remark is worth noting. The hazard level provides an index of accident escalation, *regardless of the sequence of events that leads to that particular accident*. In other words, the hazard level spans every sequence and scenario of escalation that will lead to such an accident occurring. The choice of setting up a metric based on the system state (i.e., a proxy of its internal condition) allows to eliminate the path-dependency implicit in traditional PRA, where the computation of the conditional probabilities that lead to an accident occurrence has to account for the specific path followed by the system.

In general, 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 3.4 the string that starts with the initiating event IE_1 and terminated in the accident state A_k is written as:

$$s_{1,k} = IE_1 e_2 e_3 \ldots e_n A_k \tag{3.13}$$

where each event (e) in the sequence provided by Eq. (3.13) presents one subscript that identifies its position inside the string s. As indicated in Figure 3.4, 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

$$p(e_{i+1}|e_i) \quad \text{or generally} \quad p(e_k|e_i) \quad \text{for } k > i \tag{3.14}$$

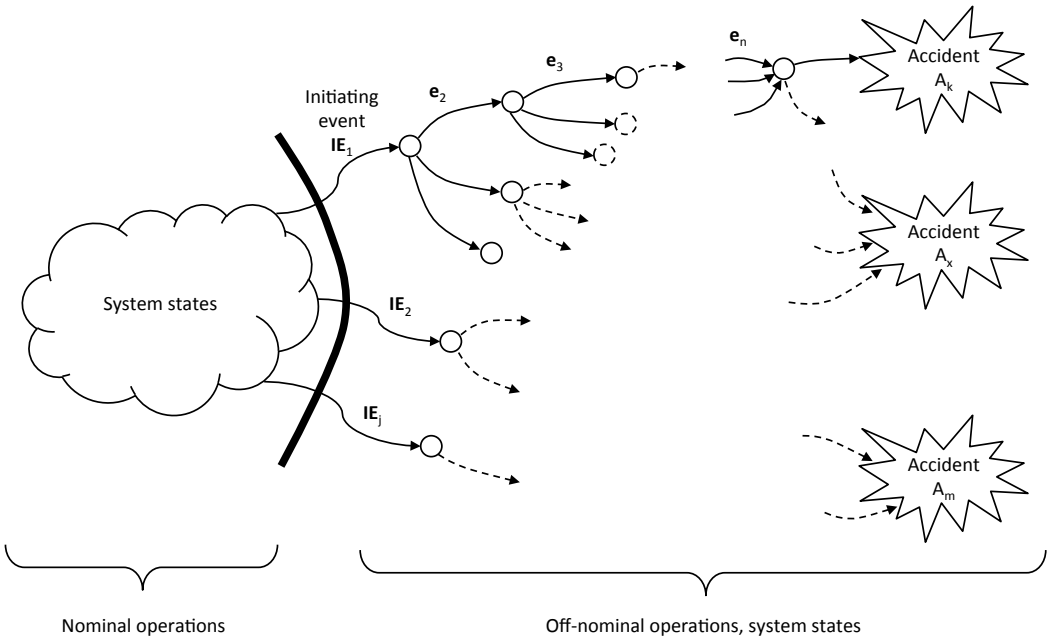

Figure 3.4 Schematic representation of an accident sequence

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. (3.15), based on the scenario expressed by Eq. (3.13), needs to be computed.

$$p(s_{1,k}) = \; p(IE_1) \cdot p(e_2|IE_1) \cdot p(e_3|e_2) \dots p(A_k|e_n) \tag{3.15}$$

The approach proposed strikes then for its simplicity in handling the computation of the proximity to accident A (and the ensuing time-to-accident metric that is analyzed in section 3.4) regardless of the particular sequence of events followed by the system. In short, danger indices are agnostic to the series of events that led to their particular value at any given instant of time, and as such they are independent on the specific accident trajectory followed by the system. The set up of danger indices for the system hence shifts the reliance of the risk assessment process from the identification of all possible accident trajectories and their associated probabilities to the identification of suitable hazard levels, whose choice is informed by the particular safety requirements imposed for the system.

3.3.2 Hazard Level Monitoring

The last step in the Safety Supervisory block in Figure 3.1 is the hazard level monitoring. To illustrate its role, consider the first requirement that was set up for the oil tank example, i.e., ensuring that the tower does not overflow. Intuitively, the implementation of this requirement in a quantifiable form implies the verification of the following constraint for the hazard level:

$$H(t) < H_A \tag{3.16}$$

where H_A represents the hazard level associated with the onset of the accident "loss of containment", thus $H_A = 1$ in the example of $H(t)$ provided in Eq. (3.8). Properties such as that expressed in Eq. (3.16) allow the set up of safety bounds (or safety envelopes for higher dimensions than 1D) and criticality thresholds for the hazard level. Safety margins can also be accounted for in the definition of the threshold values, so that in general it is required that $H(t) < H_{crit}$, for a pre-defined H_{crit} criticality threshold.

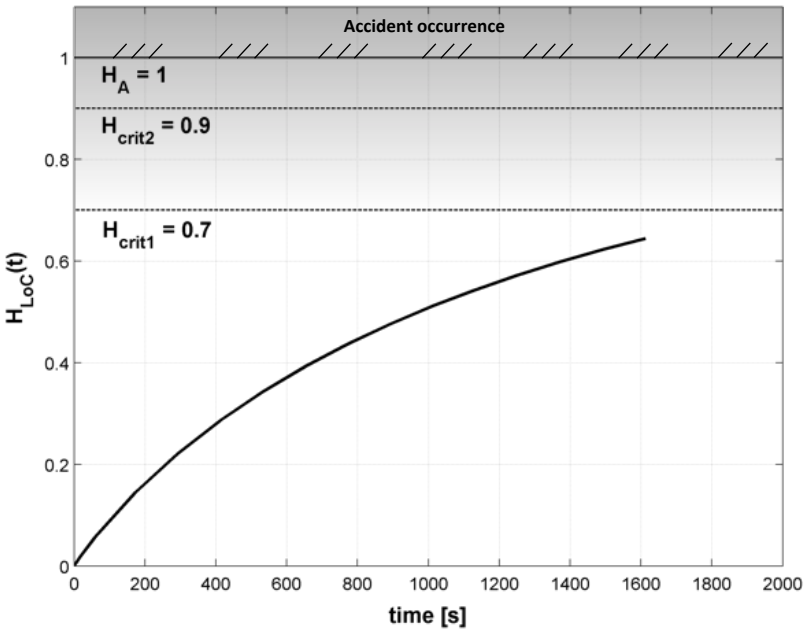

Figure 3.5 Hazard level dynamics for the oil tank example and comparison with criticality thresholds

Continuous monitoring of the hazard level informs the operators of developing dangerous situations, and thus supports their situational awareness by capturing the specific hazard dynamics and escalation (and the particular accident the system is approaching). As an illustration, Figure 3.5 provides an example of hazard level

dynamics for the oil tank example, compared in this case to three criticality thresholds: one corresponding to 70% of the tower filled up (H_{crit1}), one corresponding to 90% of the tower filled up (H_{crit2}), and one corresponding to actual overflow conditions (H_A). In the case of Figure 4, the hazard level value is obtained by computing the height of oil inside the tower through direct integration of Eq. (3.7), using as input values a constant incoming mass flow of 35 kg/s and considering a partially closed outflow line.

Plots such as the one of Figure 3.5 can serve as a diagnostic tool to inform *on-line* safety interventions. For instance, in this case a value of *H(t)* too close to a critical threshold H_{crit}, and a sustained positive slope for *H(t)*, suggests to the operator that a safety intervention is warranted—at a minimum to block the dynamics of hazard escalation through emergency shutdown for example, or fully open the outflow line to de-escalate the hazardousness of the situation and decrease the height of the oil in the tower away from the critical thresholds. This corresponds (from a control/mathematical perspective) to adjusting the values of the control matrix Ψ in the hazard equation. By comparing the current value of *H(t)* to the criticality thresholds, the operator is also enabled to get a real-time estimate of the time when the thresholds will be (b)reached, as examined in the next section.

When the hazard level monitoring is executed *off-line*, a detailed analysis of the history of hazard dynamics can help answer important questions regarding on the one hand, the occurrence and ranking of near misses (frequency and severity or hazardousness—how close the situation got to critical thresholds), and on the other hand, the identification of missing or ineffective safety features, that allowed the increase in the hazard level, including inadequate operator training. Although the following topic is tangential to the purposes of this work, I believe the connection between the proposed safety supervisory control and model-based hazard monitoring on the one hand, and near miss management systems on the other hand [Gnoni and Lettera,

2012; Gnoni et al., 2013] offer many possibilities for meaningful contributions and is a rich area for further research and investigation.

To further illustrate the capabilities and insight that can be derived from the hazard monitoring process, the next subsection provides an application of the presented tools in support of the "go/no-go" decision-making in rejected takeoff situations (RTO), which will be instrumental for the analysis of chapter 5.

3.3.3 Example Application of H(t) Monitoring

Traditionally the thinking about the problem of setting regulations and policies for rejecting a takeoff has revolved around the notion of the decision speed V_1. Pilots are advised against rejecting a takeoff after the decision speed V_1 is achieved unless they have reason to believe "the aircraft cannot be safely airborne" [ECAST, 2016].

Statistics show that there is more to the "go/no-go" decision than the simple "stop before V_1" and "go after V_1" strategy [TSTA, 2016]. The fact that the V_1 limit is not sufficient in of itself is recognized by both air manufacturers and regulators, who advocate new metrics to expand on the current thinking about these issues. For instance [Airbus, 2005] shows that about 54% of runway excursions occur when RTOs are initiated at speed above V_1, but also highlight that about 26% of them occur for RTOs initiated below V_1.

The set up of hazard levels and criticality thresholds can support pilots in their decision to reject the takeoff versus "take the problem into the air" strategies. Consider for instance the hazard level defined in Eq. (3.17).

$$H(t) = \frac{d_{STOP}(t)}{l_{run}+d_{RESA}-d(t)} \tag{3.17}$$

This hazard level quantifies and relates the distance required for the aircraft to come to a stop (once a RTO is initiated) to the total length available to the aircraft

before encountering an obstacle on its path. This length is computed as the runway length still available (given by the runway length l_{run} minus the distance already traveled d(t)) plus the runway end safety area (d_{RESA})[7]. Rather than defining the accident as a simple runway overrun, this danger index identifies the accident as that condition for which the stopping distance required would bring the aircraft beyond the limit of the RESA. In other words, the situation *H(t)* = 1 would thus identify either a collision with an obstacle and/or the encounter of highly uneven terrain.

The calculation of the stopping distance $d_{STOP}(t)$ depends on several factors, such as the velocity at which the RTO is initiated, the position of the aircraft along the runway, the conditions of the runway (e.g., wet, dry,...), and the availability of the brakes and thrust reversers among other things. In order to compute such distance, it is necessary to set up a model for the aircraft dynamics during the RTO. For simplicity, only the longitudinal motion of the aircraft along the runway is considered, and some simplifications for the aerodynamic coefficients of interest are made. The governing equation is provided by

$$m\frac{d^2x}{dt^2} = T - D - \mu_r(W - L) \tag{3.18}$$

m is the vehicle mass; T the thrust provided by the engine(s); D the drag, and it is dependent on the aircraft configuration (e.g., with flaps and slats deployed) and the velocity $\frac{dx}{dt}$; μ_r is the rolling friction coefficient (and for the RTO case its increase models the brakes application); W is the aircraft weight; L the lift. Equation (3.18) can be translated in the state-space representation formalisms as follows:

---

[7] The runway end safety area (RESA) accounts for an additional region beyond the end of the runway before sudden changes in the terrain gradient and/or obstacles are encountered.

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{T(t) - D(x_2(t)) - \mu_r(t)(W - L(x_2(t)))}{m} \end{cases} \tag{3.19}$$

where the first state x_1 represents the distance traveled along the runway (x-axis), and the second state x_2 the instantaneous velocity of the aircraft. The following plots are obtained applying the model to the data of a Learjet 60. The following assumptions are considered: full braking power and thrust reversers are available, and the runway is dry.

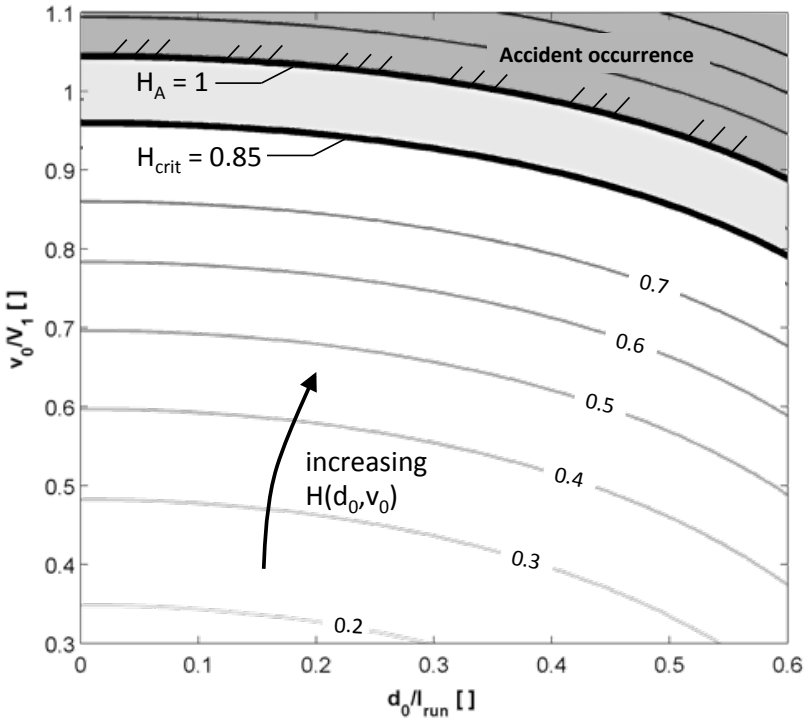

Figure 3.6 Contours of the hazard level of Eq. (3.17) plotted as a function of the initial conditions for the RTO

The model of Eq. (3.19) is integrated to compute distance, velocity, and acceleration of the aircraft at any point in time. Specifically for the RTO, when brakes and thrust reversers are applied, the stopping distance is computed as the distance corresponding to a zero velocity. This procedure can be repeated for a range of different

initial conditions, i.e., for a range of different velocities v_0 and positions along the runway d_0 at which the RTO is initiated. Plotting the hazard level as a function of these initial conditions (which is normalized for convenience with respect to V_1 and to the runway length) yields plots such as the one of Figure 3.6, where two criticality thresholds are highlighted. The first threshold represents situations in which the aircraft comes to a stop within a 15% safety margin from the end of the RESA, while the second threshold corresponds to the accident unfolding.

The accident threshold $H_A = 1$ can be compared to the traditional limit imposed on the decision speed V_1 (Figure 3.7).

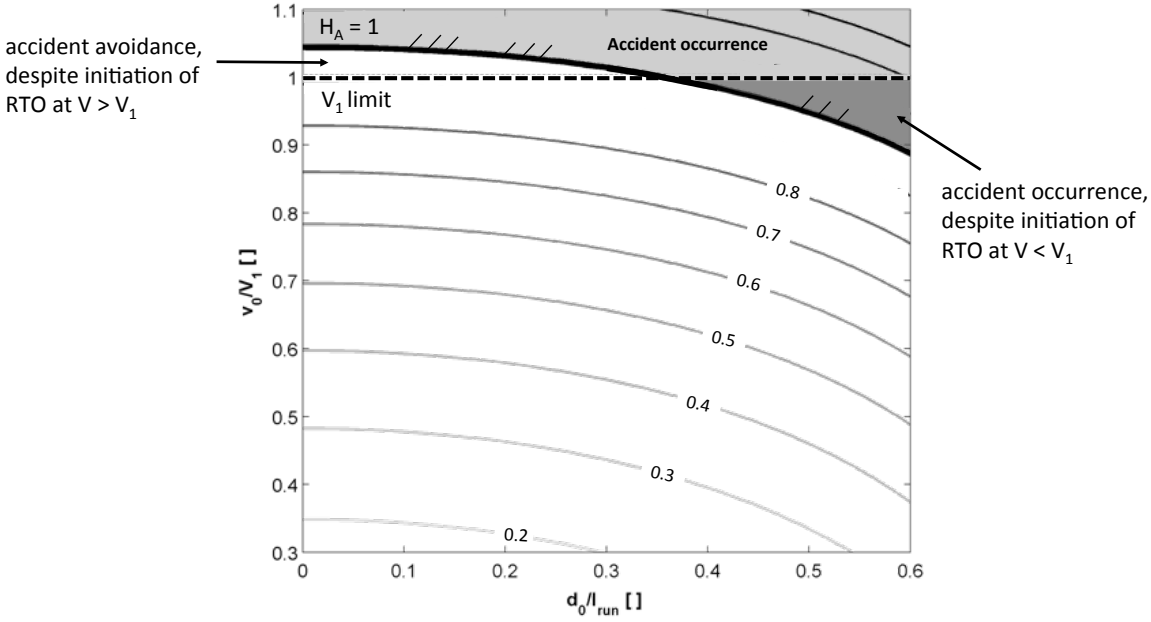

Figure 3.7 Contours of the hazard level of Eq. (3.17) and comparison with V_1 limit

Figure 3.7 provides a clear visualization of how the metric established by Eq. (3.17) informs traditional approaches for the RTO decision-making problem. Specifically, by accounting for the stopping distance dependence on the state of the system when the RTO is initiated (in terms of velocity and position), the selected hazard level can account for both situations in which RTOs are initiated below V_1 and

still result in an accident, and situations for which RTOs are initiated above V_1 that do not.

Finally, Figure 3.8 superimposes a typical aircraft trajectory during takeoff to the mapping of Figure 3.7. It can be seen that for this particular scenario (dry runway and thrust reversers deployed), the trajectory briefly enters the new "danger area" highlighted in Figure 3.7. More so will be in the case when full braking power is not available, or the runway conditions are less than ideal. As the possibility of an RTO should always be considered by the pilots before the initiation of takeoff procedures, a situation such as the one of Figure 3.8 can advise the pilots to reconsider the suitability of that particular runway and/or make sure that the entire available length of the runway is exploited (e.g., not starting the takeoff from an intersection with a taxiway).

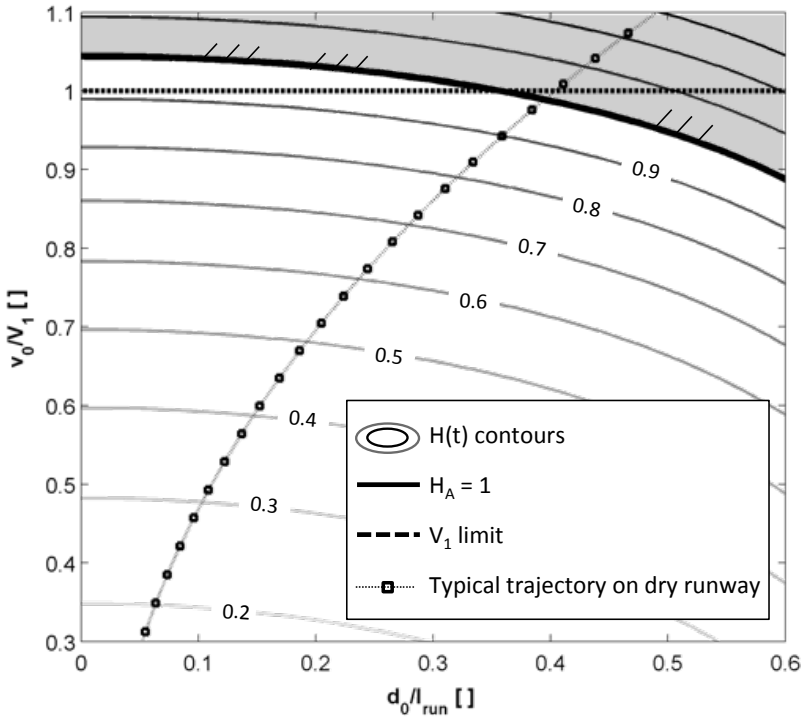

Figure 3.8 Comparison of "danger areas" and typical aircraft takeoff trajectory for best-case scenario

Metrics and diagnostic tools such as the one here considered can also be employed by regulators and policy makers to inform safety guidelines, and at the same time they can be applied on-line to support real-time decision-making in critical situations. Research is required for their adaptation to different contexts and several interesting research opportunities arise (for instance, in the proposed case, in relation to the devise of avionics development and user interface/displays in support of the proposed metrics). The safety supervisory monitoring process presented in this section offers many advantages that complement the traditional approaches to risk assessment. Other than the diagnostic information presented in this section, the continuous monitoring of the hazard level within a model-based approach supports a *prognostic dimension* as well, which is introduced next.

3.4 Hazard Temporal Contingency Analysis

This step is shown downstream of the safety supervisory monitoring block in Figure 3.1. It is shown as a separate entity to highlight its importance. The development of a hazard equation (Eq. 3.12), which is enabled by the adoption of a model-based approach, allows one to estimate the time at which critical thresholds for the hazard level are (b)reached. This estimation process provides prognostic information and produces a proxy for a time-to-accident metric or advance notice for an impending adverse event. This temporal metric[8] can also be construed as providing an estimate for the time-window available for safety interventions, assuming no changes are made to the system operation/inputs. This helps with the identification of the temporal criticality

[8] The time-to-accident metric can be described as a random variable, or more appropriately a stochastic process. One objective of a dynamic risk assessment and accident prevention is to monitor and control the set of such metrics in a system, and keep them at a safe temporal distance away from 0.

of different hazards on the one hand, and the prioritization of attention and defensive resources for hazards that warrant more timely intervention on the other hand.

To illustrate this estimation process, consider one more time the oil tank example. Given the current value of the hazard level at time t_e (the time at which the estimation will take place), the remaining time before the LoC accident occurs, assuming no change of inputs, can be derived using various estimators, the simplest one is expressed as follows:

$$\widehat{\Delta T_{LoC}}(t_e) = \frac{h_{max} - h(t_e)}{\dot{h}(t_e)} = \frac{1 - H_{LoC}(t_e)}{\dot{H}_{LoC}(t_e)} \tag{3.20}$$

The knowledge of these two "coordinates" of a hazard, $H_{LoC}(t_e)$ and $\Delta T_{LoC}(t_e)$, provides an important feedback for operators and decision-makers to dynamically monitor and actively manage the hazard of loss of containment in real time.

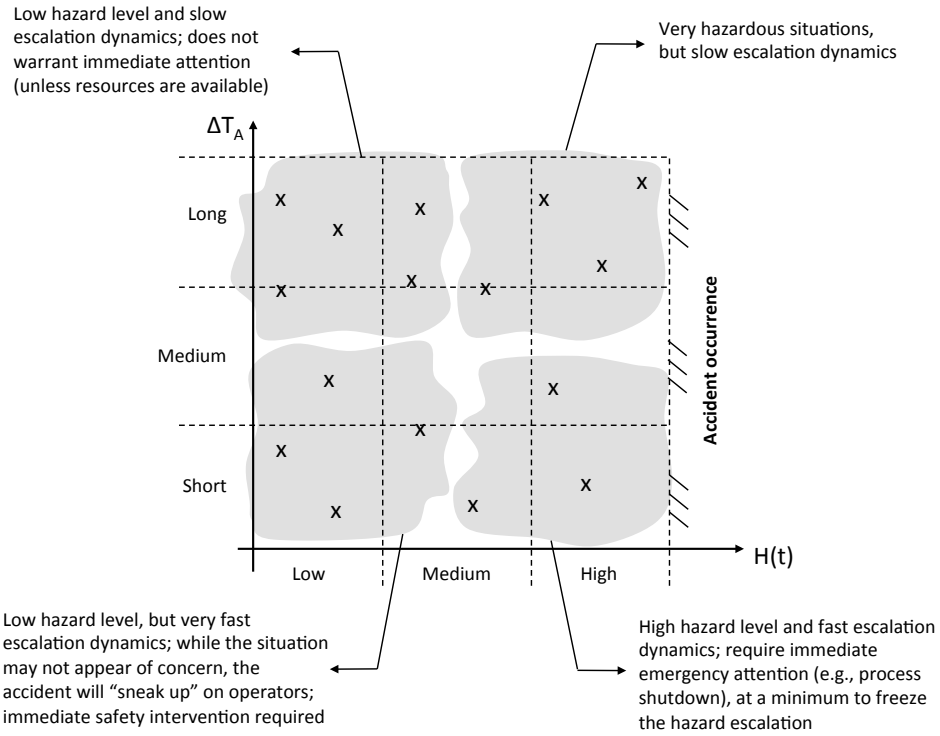

Figure 3.9 Illustrative hazard temporal contingency map

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". This collective information can then be displayed dynamically in a *hazard temporal contingency map* (Fig. 3.9) to support operators' sensemaking and help them prioritize attention and defensive resources for safety interventions and accident prevention[9].

Figure 3.10 provides a graphical illustration of how the estimate for the time-to-accident ΔT_{Ai} can be achieved in practice (shown here for two accidents A_1 and A_2). The plots represent the evolution in time of the quantity $H_{Ai} - H(t)$, which reflects how far the current hazard level is from the level associated with each accident (normalized at 1 for simplicity and consistency with the similar feature discussed in the previous examples). The two panels in Figure 3.10 show the situation at two instants of time. The top and the bottom plots relate to two different hazard indices $H_1(t)$ and $H_2(t)$. At the beginning of the monitoring period (left panel), both indices indicate no hazardous condition developing ($\Delta T_{Ai} \rightarrow \infty$). At time t_2, both hazard levels $H_1(t)$ and $H_2(t)$ escalate, the former faster than the latter (right panel). In this situation a simple estimation of the time to accidents for both indices informs the operators which sequence deserves more timely attention or immediate intervention ($H_1(t)$ in this case). The time-window available for safety interventions can be simply estimated according to Eq. (3.21):

$$\widehat{\Delta T_A}(t_2) = t_A - t_2 = \frac{H_A - H(t_2)}{\dot{H}(t_2)} \qquad (3.21)$$

[9] Trends over time and uncertainty bars in the estimates of both hazard coordinates can also be assessed and displayed.

More elaborate estimators can be devised to account for the persistency of increase in H(t) as well as its slope and other dynamic features. Furthermore when the estimate is conducted repeatedly over time, a probability density function of ΔT_{Ai} can be obtained, thus reflecting the true nature of this time-to-accident metric as a random variable. Several uses can be made of this random variable and its features to inform safety-related decision-making, for example the shrinking of its standard deviation would reflect an increasing certainty of an impending accident (should business-as-usual in the operation of the system be maintained, or no safety intervention triggered). These issues are left as fruitful venues for future work.

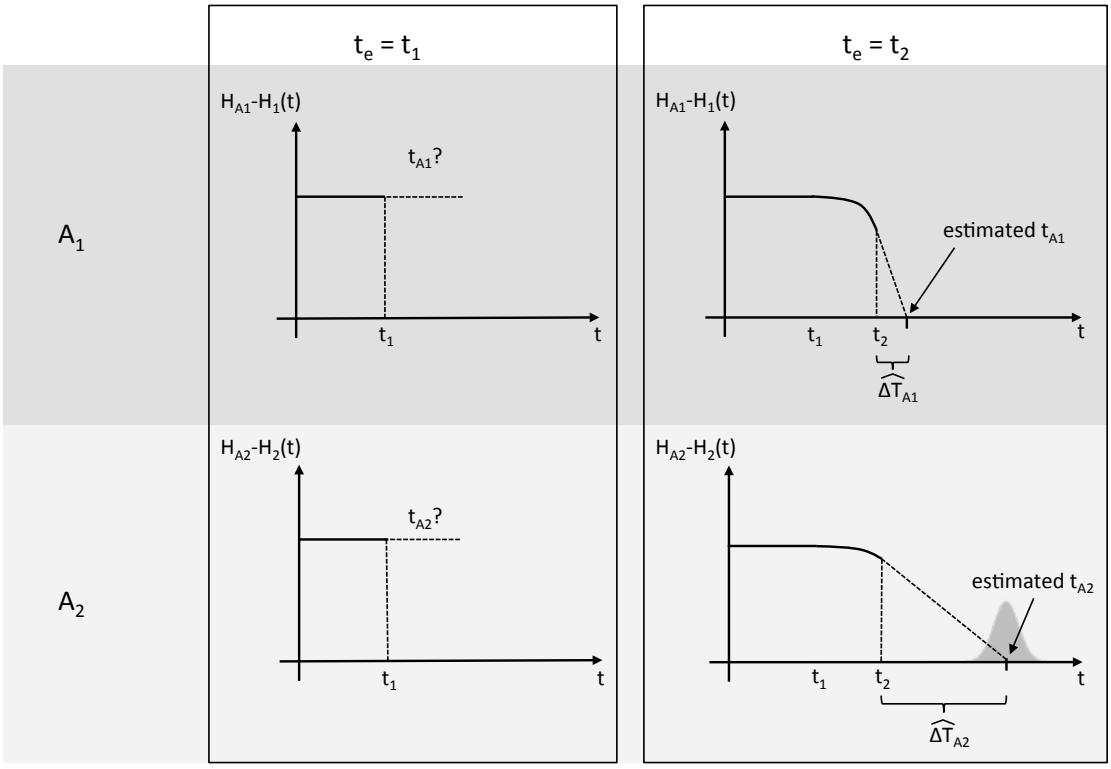

Figure 3.10 Illustrative estimation of the time-to-accident for two hazard indices

The considerations presented in this section also allow for the identification of areas were additional response capacity is required in order to improve the response

time to emerging hazards and/or other mitigating actions, as advocated in [Mosleh, 2014]. The effect of safety interventions directly translates into decreases in the hazard level, and hence new estimations of the time-to-accident metric (i.e., extension of ΔT_A). Scenario-based testing can ensure that the safety features included in the system provide the operator with enough time to either trigger a safety intervention and abate the hazard level (i.e., block an accident from unfolding), or to mitigate its consequences should it occur (e.g., with a timely evacuation).

CHAPTER 4

TEMPORAL LOGIC SYNTAX AND PROPERTIES FORMULATION

This chapter introduces the Temporal Logic (TL) syntax and motivates the use of this formal language to bear on risk assessment and system safety issues. Furthermore, it presents in detail the formulation of TL safety properties, to be used in conjunction with the model-based hazard monitoring approach presented in chapter 3. The chapter is structured in the following way. Section 4.1 presents the high-level motivations for the adoption of TL. Section 4.2 provides an overview of the TL syntax and of the process for verifying safety properties. Section 4.3 introduces the safety properties that will be used for the case-study of chapter 5 and will serve as constraint for the system behavior. Section 4.4 presents their TL formulation.

4.1 The Adoption of Temporal Logic

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 [Kress-Gazit et al., 2009], or the specifications of software program semantics capable of dynamically adapting to changing external conditions [Zhang and Cheng, 2006]. 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 [Fainekos et al., 2009]. 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 [Baier and Katoen, 2008]. 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 proposed offers novel capabilities, complementary to PRA, and rich possibilities for further contributions toward accident prevention and improved risk management.

There are several reasons that motivate the introduction of TL to bear on risk assessment and system safety issues. First, 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 enables the explicit expression of these notions, translating the event-based path dependency (implicit in risk analysis) into time-based considerations. Second, 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 [Mosleh, 2014]. Finally, the adoption of temporal logic also allows sharing the benefits of formal verification techniques (standard in computer science) for risk assessment. The potential for formal verification techniques, adapted to risk and safety applications, remains largely unexplored. 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 assessment and system safety can serve a useful purpose and a first step towards this aim.

In the proposed framework, TL is employed in conjunction with the tools presented in chapter 3, to augment the definition of the hazard level H(t) with the use of temporal operators, and to express constraint on the overall behavior of the system. In this chapter tackles this second ingredient in detail, while the integration with model-based hazard monitoring is thoroughly examined in chapter 5.

4.2 TL Syntax 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 [Galton, 1987]). TL adds operators that are related to time to the pool of operators from classical logic [Fisher, 2011]. 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 a good candidate to overcome some of the time-related limitations of traditional PRA highlighted by several authors [Zio, 2014; Mosleh, 2014; Favarò and Saleh, 2016a] and analyzed in chapter 2, and for the specification of key properties of systems whose behavior is time dependent, including software systems.

4.2.1 TL Temporal Operators

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 [Rescher and Urquhart, 1971]. 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 that involved timing considerations such as ordering of events in a sequence and repetitiveness of events. The basic temporal operators of TL are presented in Table 4.1. Additional details on their definitions can be found in [Manna and Pnueli, 1992; Baier and Katoen, 2008; Fisher, 2011].

Table 4.1 Temporal operators, based on [Fisher, 2011]

Operator	Description
$\Box$ (f)	f is true in all future instants of time
$\Diamond$ (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

These basic operators can be extended with annotations allowing the expression of real-time constraints [Fisher, 2011]. 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., "$\blacksquare$(f)" for always true in the past). Other operators or logical connectives used hereafter are described in Table 4.2.

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 [Baier and Katoen, 2008].

Table 4.2 Logical connectives of classical logic

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

In this work linear temporal logic is employed, and it 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 4.1. The real-time constraints previously mentioned can assist in specifying particular time intervals of interest.

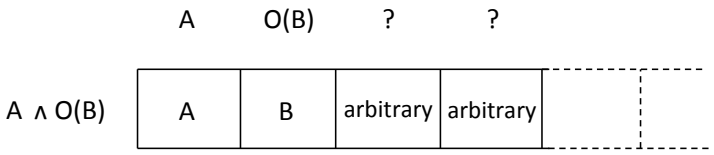

Figure 4.1 Representation of "A⋀O(B)"

The 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.

4.2.2 Verification of Properties Expressed in TL

TL provides an intuitive and mathematically precise notation for expressing properties that relate different system states at different times [Baier and Katoen, 2008]. 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) [Fisher, 2011].

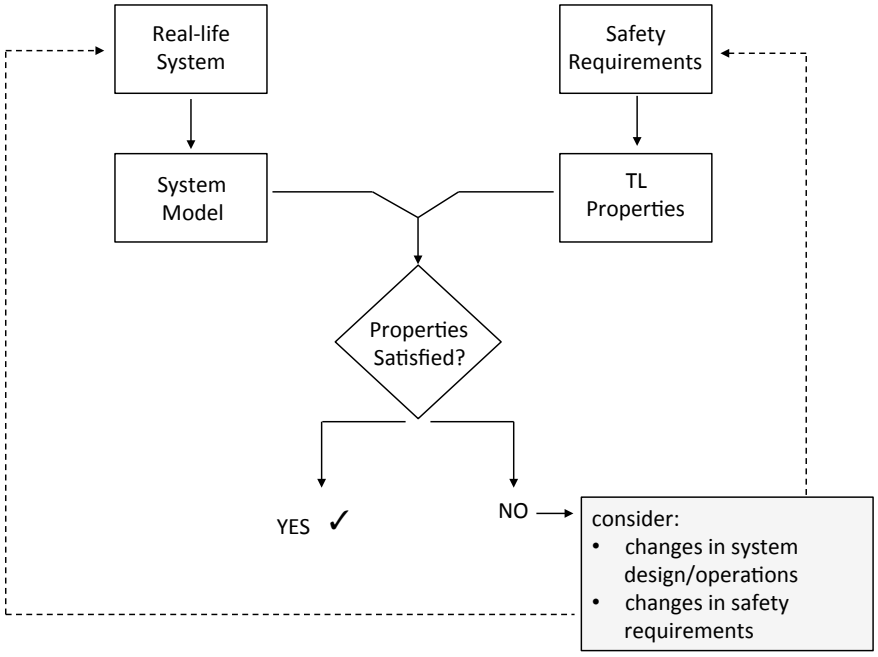

Figure 4.2 Schematic representation of the verification process

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 (more details are included in chapter 6). The verification process is schematically represented in Figure 4.2.

Figure 4.2 shows that if the compliance check is not satisfied, changes in the system design, in the system operating procedures, and/or 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, 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. As it is analyzed in chapter 5, it is important to ensure that violations are discovered during the design stages on a system, to avoid serious consequences associated to the violation of the properties during operations. Should this be the case, the online verification can still provide a useful feedback to the operators to guide safety interventions. Detailed examples are provided in chapter 5.

The overall verification of TL properties in general helps assessing 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.

In this work, I 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 that was introduced in chapter 3. 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. The analytical definition of these properties is presented next, and afterwards their TL formulation is provided.

4.3 System Safety Principles

The introduction of system safety principles formulated at a high-level of abstraction can serve a useful role in safety engineering, in addition to the current tools of risk analysis and management. As presented next, system safety principles tackle safety issues from a perspective complementary to the one of risk analysis, and help overcome some limitations of current and well-established tools. This capability is then reflected in the role TL safety properties assume in the proposed framework, where they act as constraints on the system behavior. Their violation is indicative that the principle they stand for is not correctly implemented in the system, and provides and important feedback toward the re-engineering of safer systems.

Risk analysis has been described as addressing three main questions [Apostolakis, 2004; Kaplan and Garrick, 1981]:

(1) What can go wrong?

(2) How likely it is?

(3) What would be the consequences?

The end-objective of risk analysis is to help identify and prioritize risks, inform risk management, and support risk communication. These tools however do not provide design or operational guidelines or principles for eliminating or mitigating risks. Such considerations fall within the purview of system safety. To this end, a set of four safety principles is here proposed: the *fail-safe principle*; the *safety margins principle*; the *defense-in-depth principle*; the *observability-in-depth principle*. These principles are domain-independent, technologically agnostic, and broadly applicable across industries. They are presented in relation to both the classical notion of conditional probability, and the presented model of an accident sequence and the notion of hazard-level.

The safety principles here examined provide guidelines and conceptual support during system design and operation for addressing the most important follow-up question, namely:

(4) What are you going to do about it [what can go wrong]? Or how are you going to defend against it?

For each property, a brief explanation is presented together with its analytical definition as published in [Saleh et al., 2014b; Favarò and Saleh, 2013; Favarò and Saleh, 2014; Favarò and Saleh, 2016b,c].

4.3.1 The Fail-Safe Principle

The fail-safe (FS) principle imposes, or is defined by, one particular solution to the problem of how a local failure affects the system level hazard. The local failure of a system component (or disruption/termination of its function) can propagate and affect the system in different ways. For example it can lead to a cascading failure (domino effect), which would result in a complete system failure or accident (e.g., nodes in an electric power grids operating at maximum capacity). It can also remain confined to the neighborhood of the failed item and have a limited impact at the system level.

e_f : failure of the item/function of interest at time t_{e_f}

$$\begin{cases} \dfrac{\partial H}{\partial t} = 0 \quad \text{for } t > t_{e_f} \\ and \\ p\left(e_{f+k} \mid e_f\right) = 0 \qquad e_{f+k} \in \boldsymbol{s} \ \text{ following } e_f \end{cases} \tag{4.1}$$

Specifically, the fail–safe principle requires that the failure of an item in a system or disruption/termination of its function should result in operational conditions that (i) block an accident sequence from further advancing, and/or (ii) freeze the dynamics of hazard escalation in the system, thus preventing potential harm or damage. The effects of the fail-safe principle can be expressed as indicated in Eq. (4.1).

Equation (4.1) expresses the fact that the dynamics of hazard escalation are frozen after the failure of the item/function, and the accident sequence is blocked (see Figure 4.3). Conversely, if the fail-safe principle is not implemented, the item's failure, or termination of the function it performs, would aggravate a situation by further escalating its level of hazard, thus initiating an accident sequence or leading to an accident, as shown in Figure 4.3.

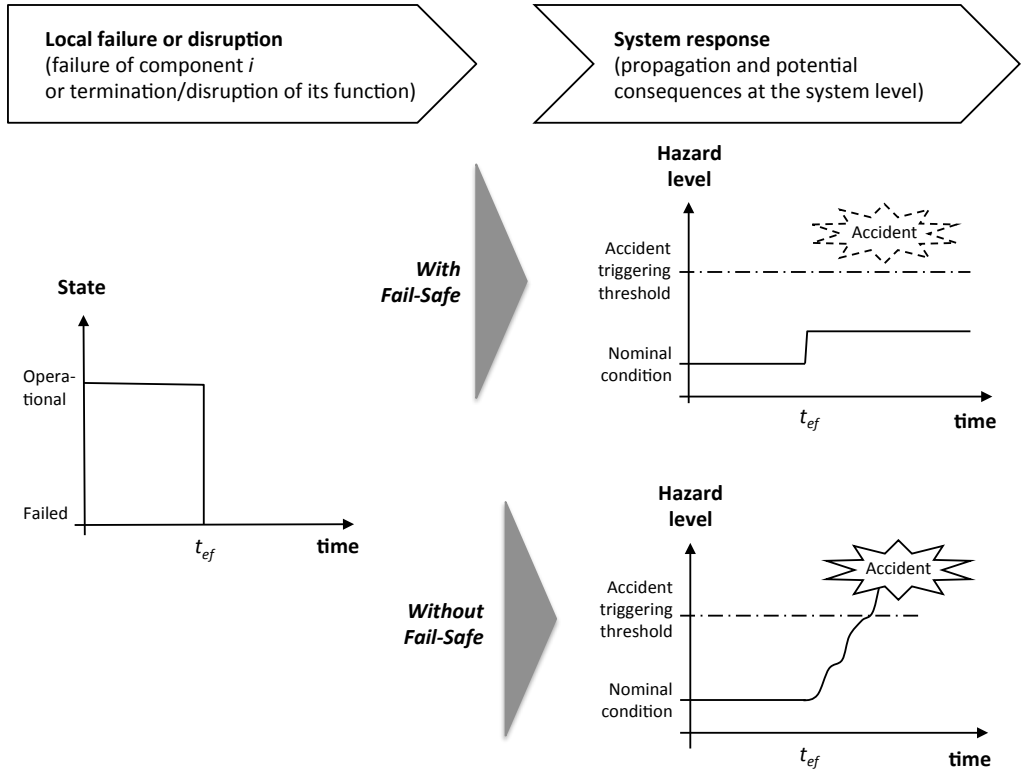

Figure 4.3 Illustrative comparison of system behavior over time following a local failure, both with the implementation of the FS principle and without it (t_{ef} is the time of occurrence of the failure of the component/function of interest)

4.3.2 The Safety Margins Principle

The adoption of safety margins is a common practice in civil engineering where structures are designed with a safety factor to account for larger loads than what they are expected to sustain, or weaker structural strength than usual due to various uncertainties. The idea of safety margins in civil engineering is an instantiation of a broader safety principle, which is here referred to by the same name.

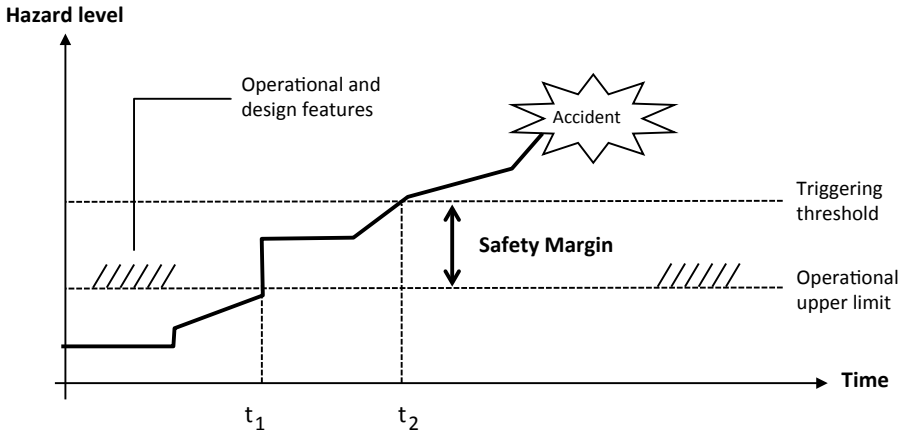

Figure 4.4 Illustration of the SM principle with a sample accident trajectory from a nominal operating condition to an accident. A larger margin makes it more likely that the system state will not reach the accident-triggering threshold, or that a longer time window is available to detect a system state that has crossed the operational upper limit (for nominal conditions) and abate the hazardous situation before an accident is triggered.

The safety margin principle has a simple form and is intuitively understood. It requires first an estimation of a critical hazard threshold for accident occurrence, H_{crit} (triggering threshold in Figure 4.4), and an understanding of the dynamics of hazard escalation in a particular situation. Secondly, the safety margin principle 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:

$$\|H_{crit} - H(t)\| \geq SM \tag{4.2a}$$

or in relative terms:

$$\left\|\frac{H_{crit}-H(t)}{H_{crit}}\right\| \geq SM_{\%} \quad or \quad \|H(t)\| \leq \frac{\|H_{crit}\|}{1+SM_{\%}} \tag{4.2b}$$

Equation (4.2) is satisfied as an equality for a particular value of H(t) termed the hazard level corresponding to the "Operational Upper Limit" (OUL).

Note that in general, the hazard level is best modeled as a random variable. There are uncertainties associated with both the estimation of its value and with the definition of critical thresholds in the first place. Safety margins are one way for coping with uncertainties in both the critical hazard threshold and in our ability to manage the operational conditions in a system such that their associated hazard level *H(t)* does not intersect with the real but unknown.

4.3.3 The Defense-in-Depth Principle

Defense-in-depth (DID) derives from a long tradition in warfare by virtue of which important positions were protected by multiple lines of defenses (e.g., moat, outer wall, inner wall). First conceptualized in the nuclear industry, defense-in-depth became the basis for risk-informed decisions by the U.S. Nuclear Regulatory Commission [NRC, 2000; Sørensen et al., 1999-2000], and it is adopted under various names in other industries. Defense-in-depth has several pillars:

i. Multiple lines of defenses or safety barriers should be placed along potential accident sequences;

ii. Safety should not rely on a single defensive element (hence the "depth" qualifier in defense-in-depth) and the successive barriers should be diverse in nature, and include technical, operational, and organizational safety barriers. In other words, defense-in-depth should not be conceived of as implemented only through physical defenses.

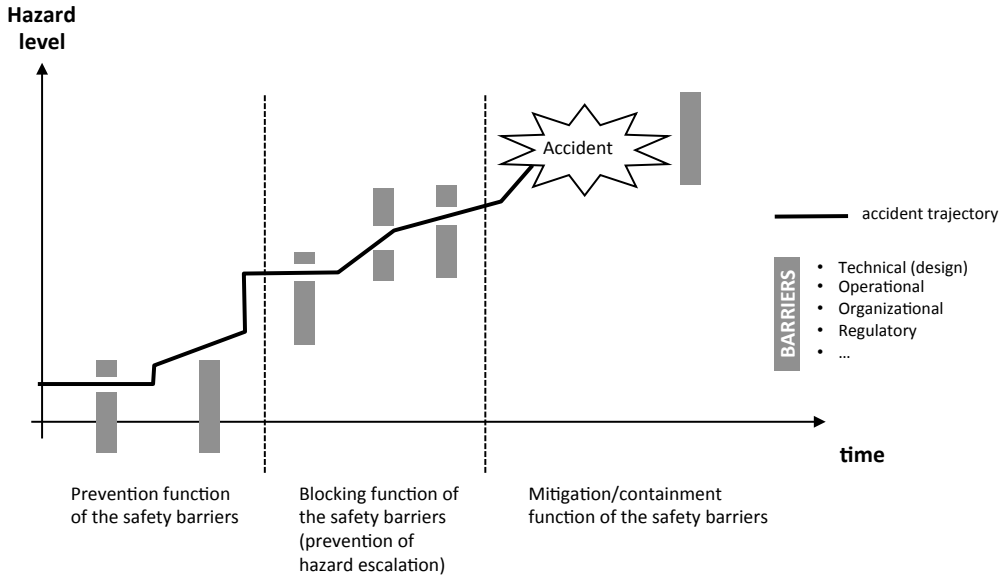

Figure 4.5 Illustration of the DID principle, along with a hypothetical accident sequence (its occurrence is the result of the absence, inadequacy, or breach of various safety barriers)

Figure 4.5 provides a schematic illustration of this safety principle, along with a particular accident sequence.

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. The first line of defense implies that safety features are devised and put in place such that the probability of an accident-initiating event (IE) is minimized:

$$\min\left[p\left(IE_i\right)\right] \tag{4.3}$$

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 from further escalating:

$$\min\left[p\left(e_{i+k} \mid e_i\right)\right] \qquad \forall i,k \quad for \quad e_i \in \boldsymbol{s} \quad \text{and} \quad e_{i+k} \in \boldsymbol{s} \quad \text{following } e_i \tag{4.4}$$

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 mitigate its consequences**. This third line of defense is designed and put in place based on the assumption that the accident will occur, but its potential adverse consequences (PAC) should be minimized[10]. The objective of the third line of defense can be expressed as follows:

$$\min\left(PAC \mid A\right) \tag{4.5}$$

These three lines of defenses constitute defense-in-depth and its three functions, namely (i) prevention, (ii) blocking further hazardous escalation, and (ii) containing the damage or mitigating the potential consequences. Notice that all else being equal, the hazard level scales with the extent of PAC. A minimization of PAC implies then hazard de-escalation.

---

[10] The potential adverse consequences are a function of both the amount of energy involved or being handled in a system, and the extent of vulnerable resources in its neighborhood (people and structures). For example, a chemical plant in the middle of a densely populated city has a higher potential for adverse consequences than if it were sited in a remote industrial zone.

4.3.4 The Observability-in-Depth Principle

Despite its general appeal, defense-in-depth is not without its drawbacks, which include its potential for concealing the occurrence of hazardous states in a system, and more generally rendering the latter more opaque for its operators and managers, thus resulting in safety blind spots. This in turn translates into a shrinking of the time window available for operators to identify an unfolding hazardous condition or situation and intervene to abate it. To prevent this drawback from materializing, I proposed in [Favarò and Saleh, 2013; Favarò and Saleh, 2014] a novel safety principle termed "observability-in-depth" (OID). It is characterized as the set of provisions technical, operational, and organizational designed to enable the monitoring and identification of emerging hazardous conditions and accident pathogens in real-time and over different time-scales. Observability-in-depth also requires the monitoring of conditions of all safety barriers that implement defense-in-depth; and in so doing it supports sensemaking of identified hazardous conditions, and the understanding of potential accident sequences that might follow (how they can propagate). Observability-in-depth is thus an information-centric principle, and its importance in accident prevention is in the value of the information it provides and actions or safety interventions it spurs[11].

To visually illustrate this argument, consider the situation represented in Figure 4.6. This is similar to the dynamics of hazard level and accident sequence represented

---

[11] While there are similarities with the notion of "observability" and "diagnosability" from Control Theory, observability-in-depth represents a broader concept that accounts also for the establishment of an observer. Additionally, observability-in-depth includes an important aspect of predicting the propagation of current states in the future to assess potential accident sequences that might follow from specific actions (and is hence prospective in nature, while Control Theory observability can be thought of being quasi-retrospective). Finally, observability-in-depth requires the direct scanning and monitoring of accident pathogens, which by definition have no visible effect on the system's output under nominal operating conditions, and as such they are not observable in a control theoretic sense [Favarò & Saleh, 2014].

previously in chapter 3, except for the distinction between the actual hazard level, *H(t)*, and the estimated or assumed hazard level, $\widehat{H}(t)$.

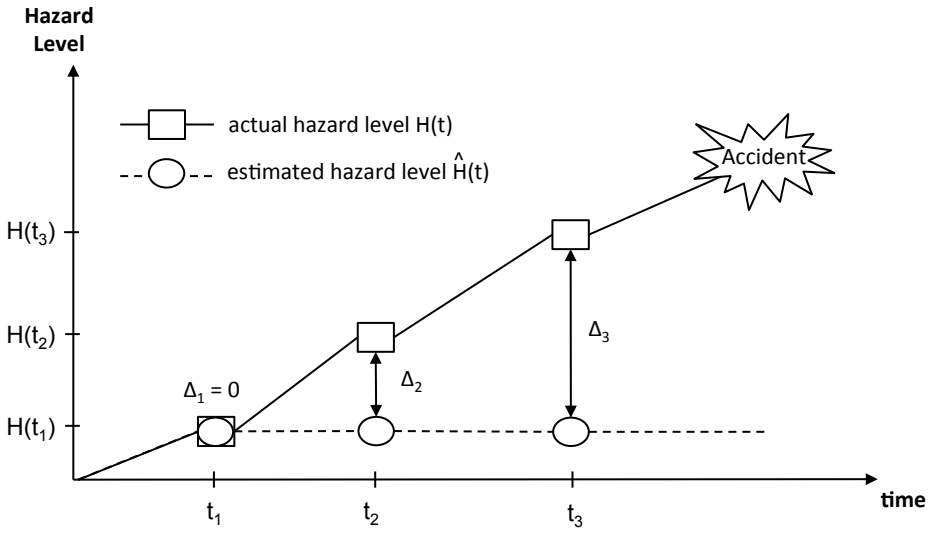

Figure 4.6 Hazard escalation over time and violation of the OID principle. The figure shows how underestimating the actual hazard level (ovals) can lead to an accident occurring seemingly without warning (rectangles).

Roughly speaking, operators make decisions during system operation, which are based on and affect the hazard level in a system. If the system conditions/states are not carefully monitored and reliably reported, there is a distinct possibility that the hazard level *estimated* by the operators will diverge from the *actual* hazard level reached by the system, as indicated in Eq. (4.6).

$$\Delta H \equiv \left\| \hat{H} - H(t) \right\| \tag{4.6}$$

The gap between these two quantities can result in the operators making flawed decisions, which in turn can compromise the safe operation of the system or fail to check the escalation of an accident sequence (e.g., no action when an intervention is warranted). In light of Figure 4.6 and Eq. (4.6), it can be said that observability-in-depth

seeks (i) to minimize the gap between the actual and the estimated hazard levels, and (ii) to ensure that at the hazard levels associated with the breaching of various safety barriers (e.g. triggering of alarms and warnings at t_1, t_2, and t_3 in Figure 4.6), the two values (actual and estimated) coincide. This concept can be expressed as follows:

$$e_{b_i}: \text{ breach of safety barrier } b_i$$

$$\begin{cases} \min \Delta H \;\Leftrightarrow\; \min \left\| \hat{H} - H(t) \right\| \\ and \\ \Delta H_{b_i} = 0 \qquad \forall i \end{cases} \tag{4.7}$$

4.4 TL Formulation of the Safety Properties

Figure 4.2 presented the high-level view of the steps that are needed for the execution of the verification process. Chapter 3 tackled the development of the system model, the left side of Figure 4.2. The previous section presented the safety property the system should satisfy in a descriptive way, together with an analytical definition based on the notion of the hazard level (and in some cases, its counter-part in terms of conditional probabilities). The next step of the process is thus the translation of the safety properties in the language of TL.

In this subsection, the logical operators presented in Tables 4.1 and 4.2 are employed for the translation of the elements defined in Eqs. (4.1-4.7) in the language of TL. Equations (4.8-4.11) provide the TL formulae describing the safety properties introduced in the previous section. For each TL formula a detailed explanation of how to read and interpret the syntax is provided. 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 4.2. 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). I will revisit this point in chapter 5 with detailed examples.

4.4.1 The Fail-Safe TL Property

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. (4.8)) does not induce a breaching of such threshold and that the hazard level dynamics is not an escalating one.

$$\text{FS} \triangleq \{ e_f(t = t_{ef}) \rightarrow [\Box(H(t) < H_{crit}) \wedge \Box_{t>tef} \neg \frac{dH}{dt} > 0] \tag{4.8}$$

Equation (4.8) reads: "*If the local (component) failure event e_f occurs at time t_{ef}, then the hazard level is always less than the critical level and for all instants of time following t_{ef} the hazard level does not escalate*". As previously noted, Eq. (4.8) 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 4.2. The violation of a safety principle like the one expressed in Eq. (4.8) 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 hazard level are monitored at each time). Mapping the specific hazard level of interest into a diagram such as the hazard-temporal contingency map of chapter 3 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. (4.8) 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. A detailed example of such violation and the re-design shrewdness needed to overcome it are presented in the case study of chapter 5.

4.4.2 The Safety Margins TL Property

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. The TL property related to this principle is defined in Eq. (4.9).

$$\text{SM} \triangleq \{ H_{OUL}(t = t_1) \rightarrow [\, \exists\, \text{T}: \square_{\text{t<t1+T}}(H(t) < H_{crit})]\} \tag{4.9}$$

Equation (4.9) reads: "If the operational upper limit is reached at time t_1 then there exists a time T such that for all instants of time before t_1+T the critical hazard level is not reached". To set up a proper safety margin it is necessary thus to ensure time T is greater or equal to a pre-specified time-window needed for safety interventions. A corollary of Eq. (4.9) is hence 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 property, 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. (4.9). Equation (4.9) is not directly implemented in the case study of chapter 5. This is because it can be subsumed under the following property of defense-in-depth. The definition of various criticality thresholds for the hazard level *H(t)* allows to account for safety margins inside their definition. This is the case, for instance, of the fact that the critical threshold H_{crit} never corresponds to the accident occurrence threshold H_A. Setting barriers and subsequent warnings in between these thresholds already accounts for the safety margins principle.

4.4.3 The Defense-in-Depth TL Property

Three lines of defenses embody the functions of defense-in-depth. In the proposed model-based framework their quantification is straightforward and relates to the breaching of critical thresholds of H(t) and to the prevention or blocking of hazard escalation dynamics. These functions are represented in Eq. (4.10).

$$\text{PR} \triangleq \{\Box(H(t) < H_{crit})\} \tag{4.10a}$$

$$\text{BL} \triangleq \{\blacklozenge\,(H(t) = H_{crit}) \rightarrow [\Diamond \frac{dH}{dt} \leq 0 \ \wedge \Box^{+}(H(t) < H_A)]\}^{12} \tag{4.10b}$$

$$\text{MIT} \triangleq \{\blacklozenge\,(H(t) = H_A) \rightarrow [\text{PAC}_{|A} < \max(\text{PAC})]\} \tag{4.10c}$$

Equation (4.10a) reads as follows: "*The hazard level is always less than the critical threshold*". This condition ensures that prevention barriers are put in place to maintain the system within its safe operating conditions. When this condition is violated, Equation (4.10b) 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 critical threshold is reached, then it follows that at some point in the future the hazard level dynamics is*

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

frozen or it de-escalates, and that for all future instants of time the accident hazard level is not reached". The same considerations apply to Eq. (4.10c) formalizing the last line of defenses, those that embody the mitigation function. Equation (4.10c) 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*". The final function of DID is not directly related to hazard level dynamics, and an extensive body of work is available in the literature on methodologies for the quantification of the potential consequences associated to an accident and their ranking. The case study of chapter 5 will only verify and examine the first two lines of defenses, with the focus of preventing the accident from unfolding.

4.4.4 The Observability-in-Depth TL Property

The OID property is meant to eliminate the potential for safety blind spots—the concealment of hazardous states or event occurrence—in system design and operation, in support of operators' situational awareness. The sensemaking of increasingly critical conditions is related in the proposed framework to the quantification of the hazard level. Among other things this principle requires then that a correct estimation of the hazard level is achieved, and that the breaching of subsequent barriers supports and informs such estimation. The "correctness" of the estimation process is expressed in terms of the discrepancy between two evaluations of the hazard level *H(t)*. In simple terms, one evaluation is considered to correspond to the actual conditions of the system, and the other to the operator's estimation of those conditions (more details in chapter 5). The TL constraints for observability-in-depth are expressed in Eq. (4.11a) and (4.11b).

$$\text{OID1} \triangleq \{ \square\neg\left(\left\|H(t) - \widehat{H}(t)\right\| > \varepsilon\right)\} \tag{4.11a}$$

$$\text{OID2} \triangleq \{ \square[e_{b_i}(t = t_i) \rightarrow \frac{d(H(t) - \hat{H}(t))}{dt} < 0]\} \quad (4.11b)$$

Equation (4.11a) reads: "*The actual and the estimated hazard level never 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 (4.7) 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. (4.11b), which reads as follows: "*If a defense barrier is breached at time* t_i*, it follows that the discrepancy between the actual and the estimated hazard level decreases*". Violations of the OID property will have a fundamental role in the escalation of the accident sequence of the Learjet overrun analyzed in chapter 5 and are therein analyzed in detail.

The formalization of the safety principles through the TL syntax supports the real-time monitoring of emerging risks and the identification of potential vulnerabilities and deficiencies in risk managements strategies. TL safety properties act as constraints on the system behavior and are continuously checked and verified for correctness in real-time. Violations of the safety properties indicate that the principles they stand for are not correctly implemented in the system, and provide an important feedback for both designers/analysts and operators/technicians to guide safety interventions. These capabilities are carefully explored in the next chapter, with the presentation of a real-life case study. Finally, the TL syntax provides tools for the formal specification of the safety principles in a design process and can support the automatization of the verification process. As previously mentioned, the introduction of temporal logic for safety purposes creates a bridge between the risk community and the language adopted

for the automated specification of requirements in the software community. Providing common semantics across the two is a fundamental step to ensure the integration of safety in the early steps of design. I will revisit this point in the concluding chapter 6.

CHAPTER 5

INTEGRATING TL AND MODEL-BASED HAZARD MONITORING

The objective of this chapter is to integrate TL and the material presented in chapter 4 with the safety supervisory framework of chapter 3 on the one hand, and to demonstrate the practical application of the integrated framework and the novel insights it can provide for improved risk assessment and accident prevention on the other hand. The chapter is structured in the following way. Section 5.1 presents a high-level introduction on the uses of TL in support of the safety supervisory control framework. Section 5.2 examines a real-life case study used as "proof-of-concept" for the integrated framework. Section 5.3 analyzes the use of the material from chapter 4 in support of the case study and summarizes the particular insights that it enabled to derive.

5.1 TL in Support of the Safety Supervisory Control Framework

As explained in chapter 4, TL has been traditionally employed as a specification language for systems whose behavior is time-dependent (e.g., to describe the sequence of states taken up by a traffic light: first red, then green, then orange). In this work, TL is employed in the following ways: 1) to model and include temporal considerations in the analytical definition of the hazard level *H(t)*; 2) to model the behavior of software and digital components in the simulation environment; 3) to model safety properties and constraint for the system. Details on the advantages deriving from each use follow next. The use of TL for the analytic definition of the hazard level provides two advantages. First, it allows more complex expressions of *H(t)* that include temporal operators (examples are presented in the case study of Section 5.2). Explicit considerations of ordering and timing of events, faults, or sequence of states can be

included in the definition of *H(t)* itself. This provides significant benefits for expressing complex conditions or situations in a compact form (and readily testable/operationalizable), as is frequently done in software code for example, and that is exceedingly difficult to render based on state variables alone. The direct inclusion of temporal operators in the definition of the hazard level can be viewed as a *state augmentation* operation, where the state vector of the system is expanded to also account for past (and or future) states and state transitions (or states within a slice of time when other conditions are present). The richness of this expressive capability cannot be overstated. The use of TL in the definition of *H(t)* can also alleviate problems when the entire state of the system is not available and or not modeled in the state equation. For example, the use of TL can readily capture state transitions when they are needed in the definition of hazard levels (e.g., past values of a subset of the state variables and their comparison with the current values) without resorting to the dynamics of the entire system, thus bypassing the use of the state equation. In other words, TL allows to re-introduce a richness in the problem analysis that may have been originally lost by accounting for a reduced set of state variables in the system model. For instance, in Section 5.2 I will define a hazard level based on the history of the squat switch state (a sensor used to indicate whether an aircraft is on the ground or in the air) without accounting or developing a state equation for this specific state.

TL is also employed in the proposed framework for the specification of software and digital components behavior (modeled in Simulink through State Charts). This use closely follows what is traditionally done in the robotics and computer science. By leveraging a language that is typical of software systems, the proposed approach allows the integration of both software and hardware components within the same framework. As such, temporal logic can serve as a bridge between the risk/safety community and the computer science community. As previously noted in chapter 4, 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. The adoption of TL can also help reduce miscommunications occurring at the interface between the two engineering disciplines of risk/safety and software engineering [Hansen et al. 1998], and it can provide a common semantic model for terms used in safety analysis and in software requirements. Such a common model is important whenever engineers from multiple disciplines need to work together, which is the case for all modern cyber-physical systems that heavily rely on the integration of software and hardware for system design and process control. This would also work towards satisfying the need of traceability between software requirements and system requirements, as discussed in [NUREG, 1995, 1996].

Finally, TL is employed for the expression of safety properties and constraints for the system, as presented in detail in chapter 4. The use of temporal logic enables to express more complex constraints than the one of Eq. (3.16), and allows to leverage formal verifications techniques (standard in computer science) for automatic safety requirements validation.

Before presenting the case study and the detailed application of the framework, I briefly expand on this final use.

In general, *safety properties* specified in temporal logic take up the following form [Baier and Katoen, 2008; Hansen et al., 1998]:

$$\Box\neg A \tag{5.1}$$

where A represents the occurrence of an accident or adverse event, and hence the expression reads: "*accident A never occurs*". In [Hansen et al., 1998], the accident A is set as the top event in a Fault Tree Analysis (FTA), and then it is decomposed into lower level events and assembled using logic gates augmented with temporal operators. Gates in FTA allow or prevent the fault logic to propagate up the fault tree, from basic

events (e.g., at a component level, typically not further decomposable) toward the system-level top event. One important synergy emerges at this point from the integration of the previous model-based hazard modeling/monitoring with TL, which allows combining the format of Eq. (5.1) with the simple constraint provided in Eq. (3.16) into the quantifiable form

$$\square(H(t) < H_A) \tag{5.2}$$

where H_A represents the hazard level associated with the accident occurrence and reads: "*The hazard level is always less than the threshold corresponding to the accident occurrence*". Equation (5.2) represents the general requirement for the system to always remain within safety bounds of operation (with respect to accident A). When comparing Eq. (5.2) with the TL properties presented in chapter 4, it is intuitive to understand how Eq. (5.2) serves as building block for the devising of more complex constraints for the behavior of the system. Each TL property is predicated on a particular hazard level *H(t)*, and at the same time can inform the analytical definition of the specific danger index (e.g., if the system is required to abide by the fail-safe principle, a choice of *H(t)* may be more suitable than another, in a process similar to the one by which the designer/analyst chooses which state variables to pick for the state-space representation of a system).

Note that TL safety properties' expressions are independent of the specific hazards functions *H(t)* of interest. That is, the same TL property can be used for a wide range of *H(t)*, which are developed for a specific accident and within a particular system. Said differently, the hazard functions are specialized and tailored to particular contexts, whereas the TL safety properties are general and agnostic to the underlying system. They can be conceived of as elements within a broad library of safety properties to be adapted and applied for the analysis of different dynamical systems.

Applications and insights derived from the specific TL safety properties analyzed in chapter 4 and the specific hazard functions are provided in Section 5.2.

In the proposed framework, TL safety properties are continuously checked/verified for compliance. By leveraging a formal language, this approach allows the automatic generation of warning signs (e.g., the display of error messages) whenever constraints are violated, or whenever critical thresholds for *H(t)* are about to be (b)reached. This is an important capability for their online use, to support the operator's situational awareness and sensemaking of the system conditions and the timely execution of safety interventions. Their violation is a clear indication that the principle they stand for is not correctly implemented in the system, and raise concerns on the effectiveness of the safety measures embedded in the system. When applied off-line, this diagnostic information provides important guidelines in support of the design of additional safety features and system re-engineering, as it is showcased by the case study presented next.

5.2 Application of the Integrated Framework and Case Study

This section presents an analysis of a recent aircraft accident, examined within the integrated framework previously discussed. The purpose of this section is to demonstrate the practical implementation of the integrated safety supervisory control framework, and to illustrate some of the insights that can be obtained from TL and model-based hazard modeling/monitoring. In addition, within the specific context of the case study, I identify one important flaw in the logic that allows the Full Authority Digital Engine Control (FADEC), not identified during the accident investigation, and recommend a solution for addressing it (which should be considered and carefully assessed by aircraft manufactures for safer takeoffs).

The section is structured as follows: 5.2.1 provides the accident narrative; 5.2.2 presents the analytical and numerical model development; finally, 5.2.3 delves into the

identification and monitoring of the hazard levels informed by the specific constraints and their verification of compliance.

5.2.1 Accident Narrative

The selected case study involves a runway overrun by a Bombardier Learjet 60. The overrun occurred during a rejected takeoff at Columbia Metropolitan Airport, South Carolina on September 19, 2008, and resulted in the death of the two pilots and two of the four passengers, as well as total loss of the aircraft and substantial damage to the airport property [NTSB, 2010]. This section provides the salient details that are necessary to understand the analysis of the hazard monitoring process and the verification of the TL safety principles.

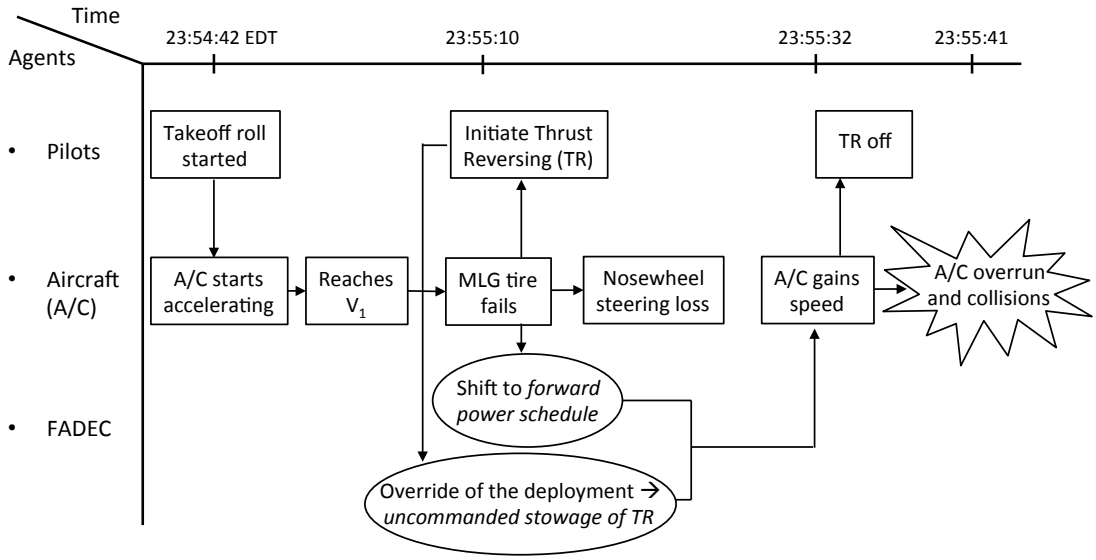

Figure 5.1 Learjet 60 runway overrun STEP diagram

Figure 5.1 shows a simplified visualization of the accident sequence through the use of a STEP (Sequential Timed Events Plotting) diagram [Favarò et al., 2013]. The diagram is structured as a matrix, with each row representing the actions or events ascribed to one agent [Embrey & Zaed, 2010], and leading up to the accident. Actions

executed by physical agents are indicated in rectangles, and ovals are used for software contributions to the sequence of events. The agents are indicated in the first column or y-axis, and the timeline is represented on the x-axis.

As shown in the STEP diagram, the accident sequence started when the pilots initiated the takeoff roll, around 23:54 EDT. The aircraft reached a speed of 136 NM/hr (V_1 speed) before the initiating event of the accident occurred. About 30 seconds into the takeoff, the tires on the main landing gear (MLG) disintegrated due to insufficient inflation, and the pilots decided to abort the takeoff. The thrust reversers (TR) were then activated using the cockpit TR lever to help slow down the aircraft. At this point an important role was played by a flawed logic in the Full Authority Digital Engine Control (FADEC), causing a hazardous situation to become unrecoverable and leading to the accident. In order to allow the deployment of the thrust reversers, the FADEC subsystem required the presence of signals coming from several sources, including the squat switches of the main landing gear. The squat switches are sensors that signal when an airplane is on the ground. The "GROUND mode" signal is received upon sensing that the MLG is appropriately compressed to support the plane's weight [NTSB, 2010]. The FADEC would not allow the deployment of the thrust reversers unless the squat switches on the MLG positively indicated that the landing gear was indeed on the ground[13]. The squat sensors of the Learjet had been damaged during the tires' explosion, and the absence of signal from the switches (no compression of the MLG could be sensed) was positively interpreted by the FADEC as the aircraft being in "AIR mode" (i.e., not on the ground). This flawed deduction was a critical factor in the occurrence of the accident since the distinction between "AIR Mode" and "GROUND

[13] This requirement was in place to prevent other dangerous situations, such as the unintentional deployment on TR while the aircraft was in flight.

Mode" enabled or disallowed many of the inputs available to the pilots to de-escalate the hazardous sequence of events, as presented next[14].

Not only did the FADEC override the pilot's request of the thrust reverser, it also shifted to a *forward power schedule* proper for the air mode. Subsequently it increased the thrust in accordance with the throttle value that the engines were set to by the pilot (though the pilot did so in backward schedule as the cockpit TR levers were engaged for TR deployment). This throttle value was increased in proportion to the maximum thrust reversing level called for by the pilots. As a result of the FADEC logic, the engines produced a high level thrust and the aircraft further accelerated. After noticing that the aircraft was still accelerating, the pilots eventually turned off the thrust reversers, seconds before overrunning the runway and striking a concrete highway marker post. The aircraft then went on crossing a five-lane road, and striking an embankment on the far side of the road, then exploding in a fireball.

5.2.2 State-Space Representation and Simulink Model

This section provides the model for the system during the accident sequence up to the first collision of the aircraft beyond the end of the runway (first 51 seconds of the accident sequence).

Figure 5.2 provides a screenshot of the Simulink model developed for the case study. The upper left portion and the central portion represent the dynamics of the system (which follows the one presented in chapter 3 and is briefly reviewed next),

---

[14] It is interesting to note that despite its importance to the accident, the FADEC logic was not examined in detail during the accident investigation and no recommendations were issued to improve on it. This is an unfortunate missed learning opportunity, and we have argued in [Foreman et al., 2015] for the need to involve software engineers in aircraft accident investigations and to dedicate a section to software contributions to the accident.

while the grey boxes around the edges contain the verification blocks for the TL safety constraints discussed in the next subsection.

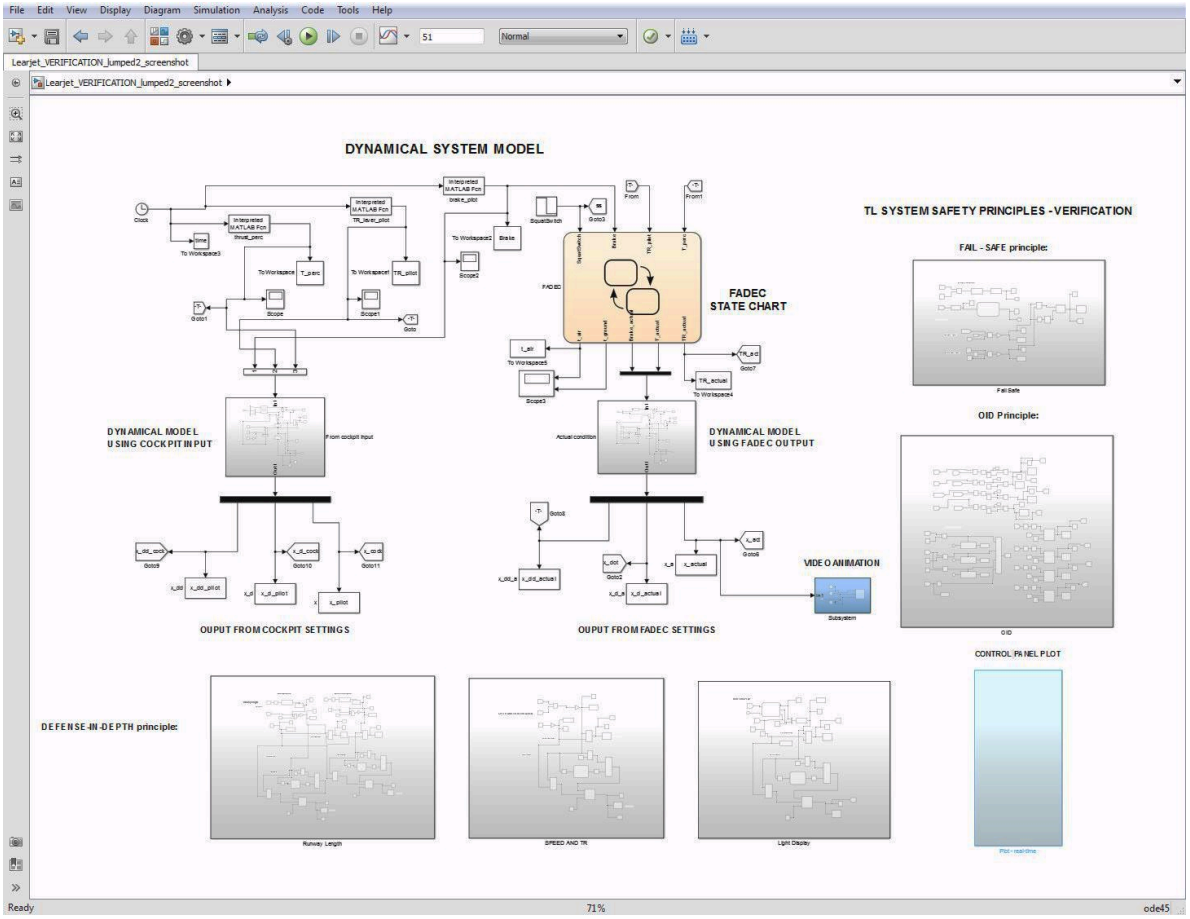

Figure 5.2 Screenshot of the Simulink model of the aircraft and FADEC at takeoff

The aircraft dynamics is treated along the x-axis (along the runway) as done in chapter 3. No lateral dynamics are examined since the NTSB accident investigation report showed that the aircraft had no relevant side movement up to the point of the first collision. The governing differential equation, which will be next translated into the state-space model is given by

$$m\frac{d^2x}{dt^2} = T - D - \mu_r(W - L) \tag{5.3}$$

m is the vehicle mass (estimated for that day at 10,800 kg); T is the thrust provided by the engines (this is an input to the model, as discussed in detail next); D is the drag, and it is dependent on the aircraft configuration and the velocity $\frac{dx}{dt}$; μ_r is the rolling friction coefficient, which depends on whether brakes are applied or not (much higher when brakes are applied), and among other factors on the tires condition (after the MLG damage its value is significantly reduced); W is the aircraft weight; L is the lift, and just like drag it is dependent on the aircraft configuration and the velocity[15].

As noted previously, the thrust value T is an input to the model. Other inputs include the position of the TR lever from the cockpit, and a binary choice for the tire brakes (applied or not applied by the pilot). The value of the thrust (as set by the thrust lever in the cockpit) is provided for the first 51 seconds of the sequence in the NTSB report as a percentage of the maximum available power (T_{max} = 20,400 N for each of the two engines).

The specification of the model's inputs makes this a scenario-based case study. This choice was made to render the present application of the framework and the results more understandable (without the added complexity of auto-generating test cases). The Simulink toolbox Design Verifier also allows the automated generation of test cases for different values of the inputs, carefully designed to ensure that all the possible combinations and settings of the model are adequately tested. Additionally, the framework is capable of handling multiple scenarios at a time, in case the user still opts

---

[15] Here are some of the basics assumptions for the computation of drag and lift: the maximum lift coefficient C_L is considered 1.25; the zero-lift drag coefficient C_{D0} is considered constant throughout the takeoff procedure with a value of 0.025; lift-induced drag is estimated using an Oswald efficiency factor of 0.71 typical of business jets [Gong and Chan, 2002] and a constant factor K obtained from the ground effect estimation: $K = \frac{\left(16\frac{h}{b}\right)^2}{1+\left(16\frac{h}{b}\right)^2}$ with b span of the aircraft and h the height of the wing from the ground.

for postulated scenarios simulation, rather than the automatic generation of the input values provided by Design Verifier. This point will be revisited in chapter 6.

Choosing the first state x_1 as the distance traveled along the runway, and the second state x_2 as the velocity, Eq. (5.3) can be translated into the state-space equations:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{T(t) - D(x_2(t)) - \mu_r(t)(W - L(x_2(t)))}{m} \end{cases} \tag{5.4}$$

where the dependency on time and on velocity have been explicitly shown for each factor.

Figure 5.3 shows a closer look of the dynamical system model. To better illustrate one of the points of the analysis that follows, I duplicate the model Eq. (5.4) and record the system behavior corresponding to two different input settings as explained next. The system structure of Figure 5.3 is divided into two parts and leading to two sets of outputs: a part that uses the pilot's cockpit input settings (on the left), and one part that uses the FADEC settings (on the right, which is modeled through the use of a State Chart). I will refer to these as the unfiltered (pilot) and filtered (FADEC) settings. This split allows to identify discrepancies, when they emerge, between the pilot's settings and how the FADEC executes or overrides them. Although this duplication is a minor detail in the present work, I believe it is rich in possibilities for testing software in cyber-physical systems and revealing deficiencies and automation flaws.

The inputs previously discussed are fed to both parts. However, the FADEC subsystem acts on the additional input provided by the state indication from the squat switches (i.e., a binary state: GROUND or AIR mode).

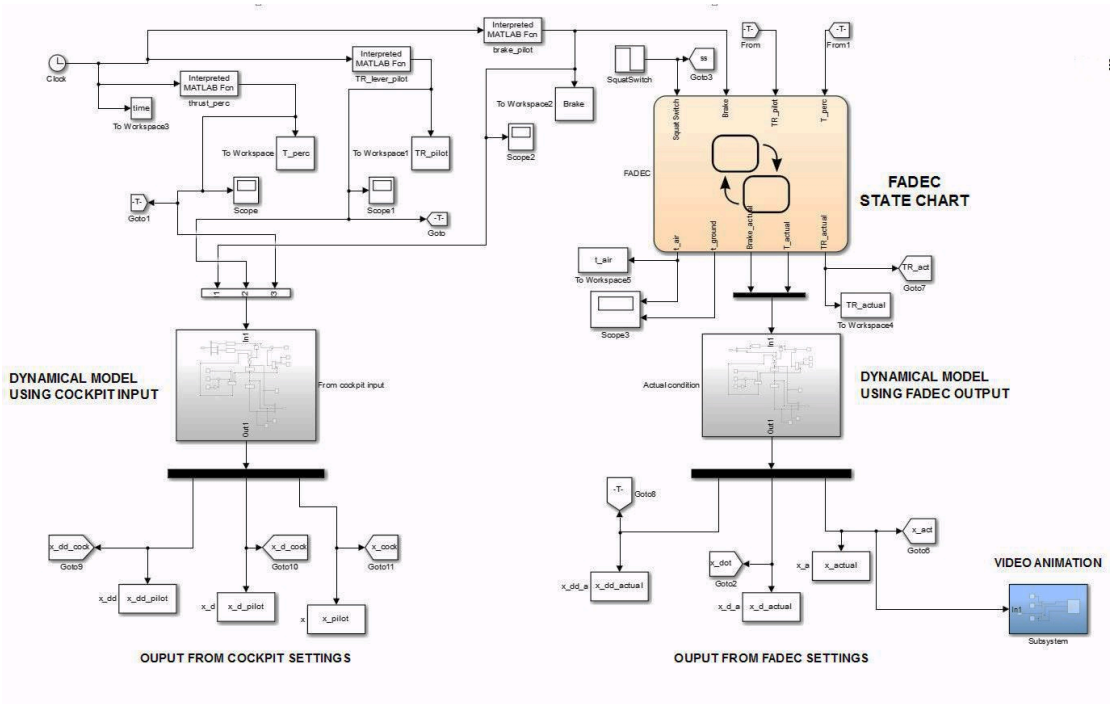

Figure 5.3 Dynamical system model

The choice to examine a "split model" with inputs from two different nodes in the system decision-making chain (in this case, the pilot followed by the FADEC) lends itself to a sort of dichotomy in the output, which is examined hereafter. On the one hand, the left side of Figure 5.3 results in the output that would be obtained if the system were indeed using the inputs provided by the pilots in the cockpit (reflecting the pilot's intent). On the other hand, the right side of Figure 5.3 results in the output that was actually obtained, with the aircraft executing inputs provided by the FADEC.

A potential discrepancy in the results coming from the two parts of the system raises concerns about a possible degraded authority and situational awareness of the pilots (and hence to correctly act on de-escalating or mitigating a hazardous situation). This point is examined in detail in conjunction with the analysis of the observability-in-depth TL property in subsection 5.2.3. Both parts use the same model for the system dynamics provided by Eq. (5.4), and whose integration structure is contained in the two "Dynamical Model" grey boxes of Figure 5.3.

The system outputs in this case are given by the distance, velocity, and acceleration of the aircraft. As noted earlier, the model for the present analysis provides two sets of outputs: those coming from the "unfiltered" cockpit inputs (i.e., exactly the settings set up by the pilots in the cockpit) and those coming from the "FADEC filtered" input settings. The model expressed in Eq. (5.4) and shown in Figure 5.3 was validated by comparing the reconstructed distance, speed and acceleration (actual values as executed by the FADEC subsystem) with the values provided in [NTSB, 2010].

5.2.3 Hazard Monitoring and TL Safety Properties Verification

Following the Learjet 60 accident in 2008, the national transportation safety board (NTSB) launched an official investigation. The ensuing report in 2010 highlighted and focused on the role that the explosion of the under-inflated tires had in initiating the accident sequence. Furthermore, it issued recommendations to the FAA and aircraft manufacturers for improving maintenance and inspection schedules.

The considerations included in the official report have an important role, which is here recognized, as they work towards the removal of the immediate cause of the accident, or in other words, in preventing the initiating event. At the same time though, it is equally important to assess the role of the factors that allowed the escalation of adverse conditions (with the dire consequences associated with them) and that failed to better inform the pilots at important decision-making nodes during the accident sequence. The upcoming analysis revolves around these considerations, and tackles them by examining three important questions, namely:

i. Was it possible to prevent the FADEC from wrongly estimating that the aircraft was in AIR mode (thus allowing the use of thrust reversers by the pilots)?

ii. Was it safe for the pilots to initiate a rejected takeoff (RTO), and if not, how could they have been made aware of the danger?
iii. Was it possible to warn the pilots that the inputs they selected in the cockpit were not being executed?

The careful examination of these questions in conjunction with the verification of the TL safety properties sheds light on blind spots in the analysis that guided both the investigation and the drafting of recommendations to prevent similar occurrences in the future. I present next the definition of several hazard indices informed by specific TL safety properties to provide an answer to each of these questions.

i. Was it possible to prevent the FADEC from wrongly estimating that the aircraft was in AIR mode (thus allowing the use of thrust reversers by the pilots)?

As noted previously, the explosion of the tires of the main landing gear (MLG) damaged the squat switches, whose indication provided an important input to the FADEC subsystem for discerning whether the aircraft was in GROUND or AIR mode. The no-signal from the damaged switches was interpreted as a signal of zero compression of the landing gear, which resulted in the FADEC estimating that the aircraft was in AIR mode.

The answer to the first question revolves then around the possibility of including a check that even for the condition of the case study (i.e., damaged switch treated as zero input) would ensure that the FADEC would not estimate the aircraft to be in AIR mode when it actually is still on the ground. In order to develop such check, which I will impose as a constraint on the system, the fail-safe principle is considered: the correct implementation of this principle ensures that a local failure does not lead to a

system-level failure. In this case I will consider the role played in the accident sequence by the failure of the squat switches on the MLG. Specifically, the local failure event is the tire explosion 30 seconds into the takeoff that led to the missing compression-signal from the switch. To set up the analytical definition of a hazard index, the first thing is to determine what condition would define a violation of the constraint of interest (in this case the TL expression of the fail-safe principle), and hence constitute a breach of property (4.8). This can be achieved in a number of ways. Temporal ordering plays a central role in the notion of "fail-safe", and hence the definition of *H(t)* can be informed by the direct inclusion of TL operators, as noted in Section 5.1. I devise the following statement of Eq. (5.5) as representing the accident for the violation of the fail-safe safety principle in relation to the squat switch (ss) operation, where V_r is the rotation speed and ss is the signal coming from the squat switches, with ss = 0 indicating AIR mode (no compression of the MLG) and a value different from zero indicating GROUND mode.

$$[(\bullet\ ss \neq 0) \wedge (ss = 0)] \quad \wedge \quad [\blacksquare\ (\dot{x} < V_r)] \tag{5.5}$$

Equation (5.5) consists of two statements, which read as follows: "*at the previous instant the squat switch was sensing aircraft in GROUND mode AND in the present instant aircraft in AIR mode"* (first bracket) *AND "the airspeed up to (and including) the present instant of time has always been less than the rotation airspeed*" (second bracket). Equation (11) thus identifies an accident as the situation in which the change from GROUND mode to AIR mode is sensed, but the airspeed is still less than the rotation speed. Since take off (and hence the switch to AIR mode) should not occur before V_r is reached, a violation of Eq. (5.5) reflects a serious safety concern with the system. Equation (5.5) sets a fundamental check that is missing from the design of the FADEC logic of the Learjet (and possibly of other aircraft as well) and needs to be

carefully considered by software developers and aircraft manufacturers. Its lack is a lurking accident pathogen waiting to contribute to further escalating an accident sequence, as was the case of the Learjet overrun. Proper checks of Eq. (5.5) should be executed before the FADEC overrides pilot's requests to engage the thrust reversers.

Once the hazardous condition to avoid is defined, it is possible to set up a quantifiable hazard level function to monitor against it. Equation (5.6) provides one choice for the hazard level, where "V_{check}" and "S_{check}" are binary functions defined in Eq. (5.7a,b), x is the aircraft position along the runway, and ℓ_{rw} the runway length.

$$H(t) = V_{check} \cdot S_{check}\left(\frac{x}{\ell_{rw}}\right) \tag{5.6}$$

$$V_{check} = \begin{cases} 1 & if \ \blacksquare \ \dot{x} < V_r \\ 0 & if \blacklozenge \ \dot{x} \geq V_r \end{cases} \tag{5.7a}$$

$$S_{check} = \begin{cases} 1 & if \ (\bullet \ ss \neq 0 \ \wedge ss = 0) = true \\ 0 & otherwise \end{cases} \tag{5.7b}$$

The hazard level is scaled in proportion to the aircraft position on the runway so that the closer the vehicle is to the end of the runway the more hazardous the situation is. A criticality threshold H_{crit} can be set up by defining a position of interest, after which the situation is considered critical. In this case, the critical threshold corresponds to values of H(t) greater than zero, as this indicated that Eq. (5.5) no longer holds true.

The hazard level defined in Eq. (5.6) is plotted in Figure 5.4 for the first 51 seconds of the accident sequence (considering the output handled by the FADEC). It is possible to see that in correspondence with the tire explosion at about 30 seconds into the takeoff the hazard level escalates, a condition that holds up until about 35 seconds, and that causes a violation of the FS property. This happens as Eq. (4.8) prohibits the

increase in *H(t)* once the critical threshold is breached. A few seconds after the violation is detected the hazard level goes back to zero. This happens due to the aircraft reaching the rotation speed, and hence V_{check} zeroing out. The detection of a constraint violation is indicative of a problem in the system. In the case of Figure 5.4, this problem corresponds to the indication of AIR mode before the aircraft has reached the rotation speed, and thus, before the aircraft has actually taken off.

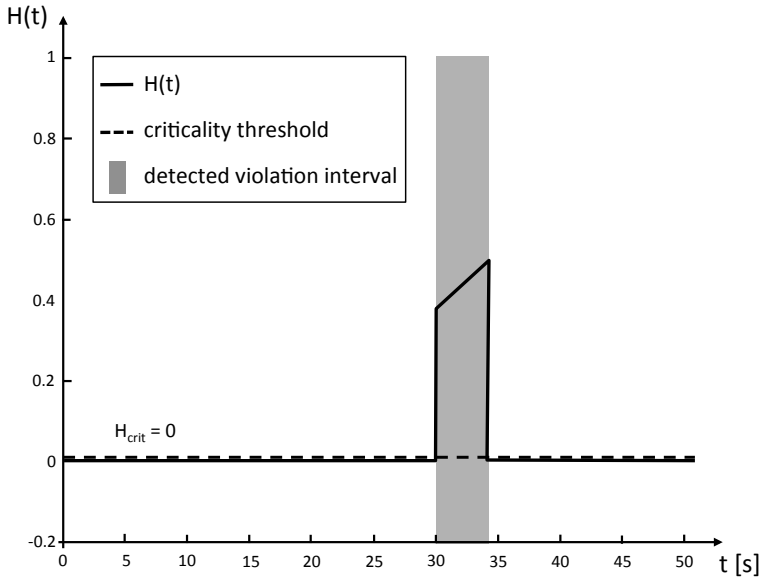

Figure 5.4 FS property – associated hazard level plot and violation detection

As noted in Section 5.1 the verification of compliance with the TL safety properties serves a useful role during the design/development stages of a system, to ensure that situations such as the one depicted in Figure 5.4 do not unfold during operations. Should this be the case, the online application of the proposed framework allows to set up warnings and proper feedback to the pilot to recognize the problem. A violation of the TL property detected offline during the design/development stages would indicate to the designers (software developers or testers) that the check of Eq. (5.5) was not implemented in the FADEC, and would hence advise towards re-coding parts of the system.

Figure 5.5 provides a screenshot of the implementation of the FS property of Eq. (4.8) in Simulink. The assertion block on the far right displays a warning when the property is violated, such as the one shown in Figure 5.6. To implement this principle I opted for the inclusion of a state chart to detect the change in the status of the squat switch. State charts offer a convenient tool for modeling discrete and event-driven subsystems.

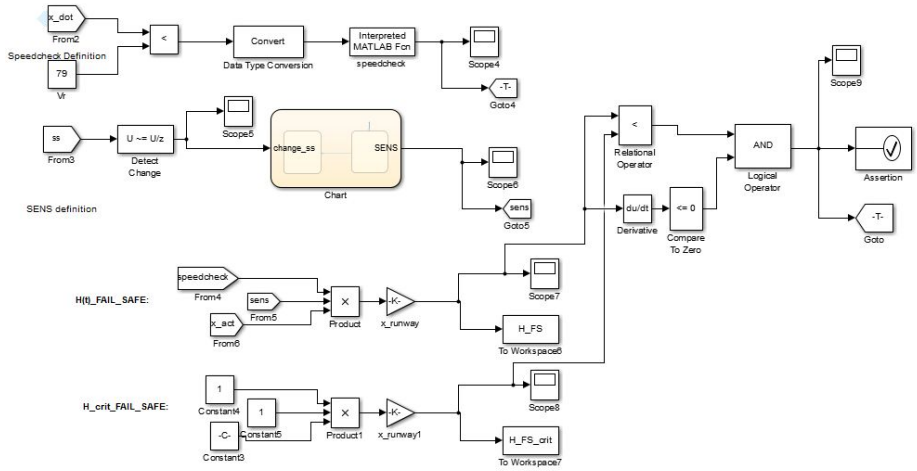

Figure 5.5 Implementation of the FS property in Simulink

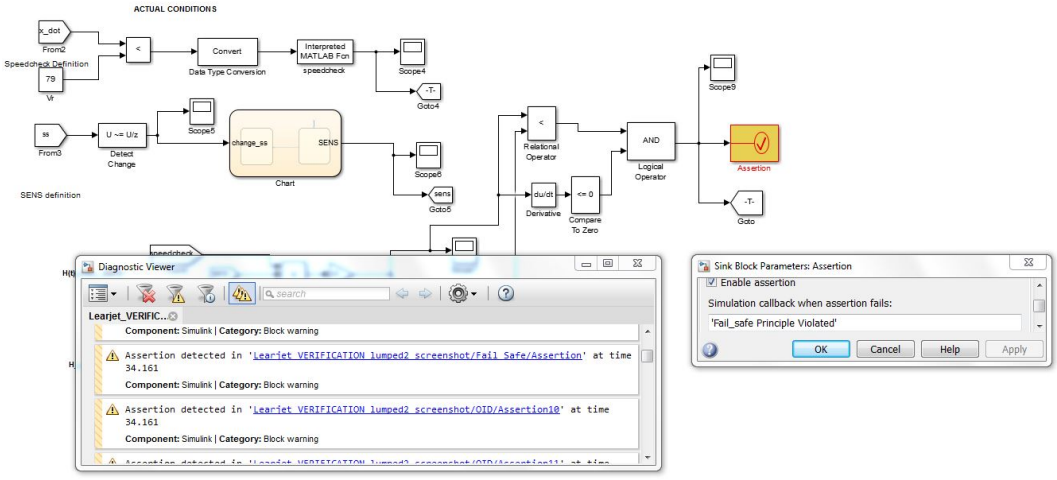

Figure 5.6 Warning detection example

The lack of a check such as that of Eq. (5.5) allowed the FADEC to override the pilot's request to engage the TR, and to automatically (and unknowingly to the pilot) shift to the forward thrust schedule. The remaining two questions are analyzed next, together with the inclusion of other possible barriers, warning signs, and indicators of the situation that could have helped the pilots and informed their decision-making even in the absence of the check of Eq. (5.5).

ii. Was it safe for the pilots to initiate a rejected takeoff (RTO), and if not, how could they have been made aware of the danger?

To date, the thinking about the problem of setting regulations and guidelines for rejected takeoffs (RTO) has revolved around the notion of the decision speed V_1. In chapter 3 this perspective was augmented through the introduction of a danger index based on the necessary stopping distance for the aircraft from the moment the RTO is initiated. This index considers the existence of an additional barrier (other than the regulatory condition on the decision speed V_1) provided by the runway end safety area. The DID principle can thus be used to inform this novel metric, that is presented again for convenience in Eq. (5.8).

$$H(t) = \frac{d_{STOP}(t)}{\ell_{\mathrm{rw}} + d_{RESA} - x(t)} \tag{5.8}$$

As explained in chapter 3, the hazard level of Eq. (5.8) quantifies and relates the distance required for the aircraft to come to a stop (once a RTO is initiated) to the total length available to the aircraft before encountering an obstacle on its path. This length is computed as the runway length still available (given by the runway length ℓ_{rw} minus the distance already traveled x(t)) plus the runway end safety area (d_{RESA}). Rather than

defining the accident as a simple runway overrun, this danger index identifies the accident as that condition for which the stopping distance required would bring the aircraft beyond the limit of the RESA. In other words, the situation $H(t) = 1$ would thus identify either a collision with an obstacle and/or an excursion into highly uneven terrain.

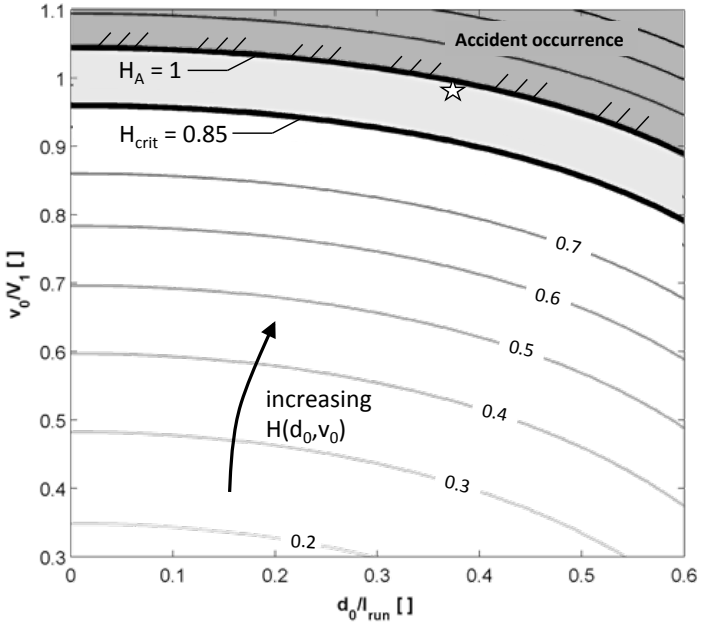

Figure 5.7 Contours for the hazard level as a function of the RTO initial conditions. Full brakes and TR (best-case)

The calculation of the stopping distance $d_{STOP}(t)$ depends on several factors, such as the speed at which the RTO is initiated, the position of the aircraft along the runway, the conditions of the runway (e.g., wet, dry,...), and the availability of the brakes and thrust reversers among other things. All of those conditions affect the analysis of the danger index of Eq. (5.8) when applied to the Learjet accident sequence. Consider once more, as was done in chapter 3, the best-case scenario (when full braking power and thrust reversers are available). By integrating Eq. (5.4) and computing the stopping distance for different initial conditions, the plot of Figure 5.7 is obatined (which was originally presented in Figure 3.6). A star on Figure 5.7 highlights the conditions at which the Learjet 60 RTO was initiated. Two thresholds are highlighted:

the first threshold represents situations in which the aircraft comes to a stop within a 15% safety margin from the end of the RESA, while the second threshold corresponds to the accident unfolding. As can be seen, the RTO of the Learjet 60 was initiated in conditions very close to the $H_A = 1$ threshold.

By comparison, it is interesting to analyze the worst-case scenario, when braking capabilities are severely compromised. In these conditions the contour levels are more skewed (as a longer stopping distance is required), and the situation becomes even more dire, as depicted in Figure 5.8. Once more, a star marker represents the airspeed/runway location coordinates of the Learjet when the pilots decided to opt for a RTO initiation.

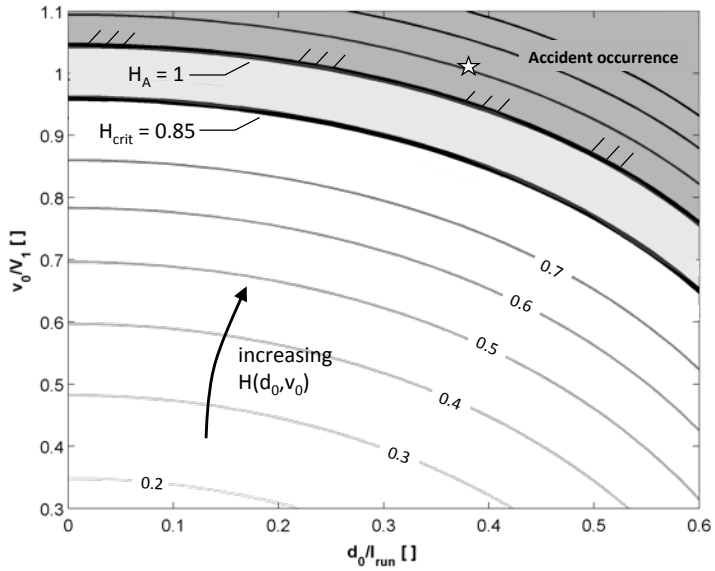

Figure 5.8 Contours for the hazard level as a function of the RTO initial conditions. Braking severely compromised (worst-case)

Plots such as the ones of Figure 5.7 and 5.8 would have advised the pilots against initiating the RTO, as the required stopping distance was (even in the best-case scenario of Figure 5.7) too close to the thresholds set up by the hazard level of Eq. (5.8). Moreover, note that these mappings can be constructed ahead of time (e.g., one

for best-case and one for worst-case scenario) and can then be used during takeoff to support in real-time the go/no-go decision-making.

To illustrate the evolving condition of the takeoff, it is possible to superimpose the actual trajectory followed by the aircraft to the mapping of Figure 5.8. This is represented in Figure 5.9. The trajectory followed by the aircraft in time corresponds to different values of *H(t)* at each instant of time, and its computation allows the verification of the defense-in-depth constraints of Eq. (4.10a,b).

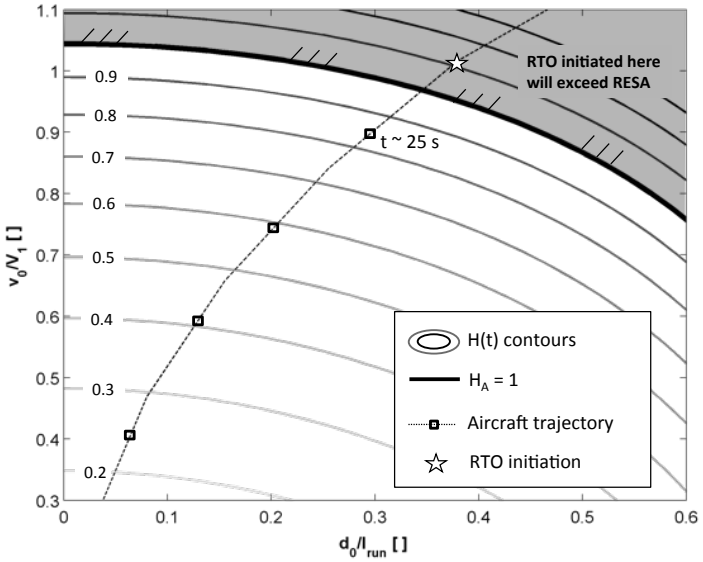

Figure 5.9 Learjet trajectory during the accident sequence, superimposed to the contour levels of Figure 5.8

The contour levels of *H(t)* can be assessed in real-time and dynamically displayed to the pilots to help them make better RTO decisions. In the Learjet case, the analysis shows that a RTO could have been initiated up to about 25 seconds into the takeoff, after which the defense-in-depth prevention constraint is violated, shortly followed by the blocking/de-escalation one. At the time when the tire exploded (~30 seconds), the proposed analysis and plot would have indicated to the pilots to "take the problem to the air" since a RTO would consume the entire runway length and most of

the RESA (the prevention constraint sets a threshold at 85% of the available RESA) before bringing the aircraft to a stop.

iii. Was it possible to warn the pilots that the inputs they selected in the cockpit were not being executed?

The Learjet accident sequence shows that the pilots initiated an RTO right after the decision speed V_1 had been reached. As noted in the previous analysis, this is a critical situation even when not further aggravated by the shift to forward thrust schedule executed by the FADEC in response to (and overriding) the pilot's request to engage the thrust reveres (TR). Nevertheless, this criticality could have been abated by checks and alarms implemented to warn the pilots of the situation and informing them of a discrepancy between the inputs selected in the cockpit and those actually executed by the FADEC. The need to check coordination and consistency of executed actions at different nodes in the line of subsystems that process a command is related to what was defined as the observability-in-depth (OID) safety principle, translated in the TL property of Eq. (4.11).

The OID principle supports the operators' situational awareness and sensemaking of escalating hazardous situations. This principle requires among other things that a "correct" estimation of the hazard level should be achieved [Favarò and Saleh, 2014]. The "correctness" of the estimation process is expressed in terms of the discrepancy between two (or more) values of the hazard level *H(t)*. In other words, the constraint of Eq. (4.11) requires the consistency among different "samples" of the hazard level. The Learjet case study displays significant violations of this principle, which is analyzed through the "split model" strategy presented in Section 5.2.2.

The split model lends itself to a dichotomy of the model's output, accounting for the "unfiltered" output obtained when pilot's inputs in the cockpit are directly executed (bypassing the FADEC) and for "filtered" ones when the FADEC re-processes them (as is the case of the actual system design, which assigns full-authority to the FADEC). For the verification of the OID constraints, three hazard indices are considered: one for position, one for velocity, and one for acceleration. This way, the following two evaluations for the hazard level vectors are accounted for

$$H(t) = \begin{bmatrix} \mathrm{x}_{\mathrm{act}} \\ \dot{\mathrm{x}}_{\mathrm{act}} \\ \ddot{\mathrm{x}}_{\mathrm{act}} \end{bmatrix} \qquad \widehat{H}(t) = \begin{bmatrix} \mathrm{x}_{\mathrm{est}} \\ \dot{\mathrm{x}}_{\mathrm{est}} \\ \ddot{\mathrm{x}}_{\mathrm{est}} \end{bmatrix} \tag{5.9}$$

where the "actual" condition (subscript "act") of Eq. (5.9) is obtained by integrating Eq. (5.4) using the inputs "filtered" by the FADEC subsystem, while the estimated condition (subscript "est") is obtained by integrating Eq. (5.4) using the inputs provided by the pilots in the cockpit.

The three states are presented in Figure 5.10, with solid lines representing the actual hazard level of Eq. (5.9), and dashed lines representing the estimated one. Figure 5.10 shows that up to about 33 seconds (the time at which the pilots called for thrust reversing), the split model outputs' evaluations follow the same trends. As soon as the request for TR engagement is overridden by the FADEC, the states start to diverge, with the input that accounts for the backward thrust schedule at maximum throttle (as requested by the pilots) showing the aircraft slowing down, and finally stopping

somewhere beyond the end of the runway but before the RESA (and the concrete highway marker post)[16].

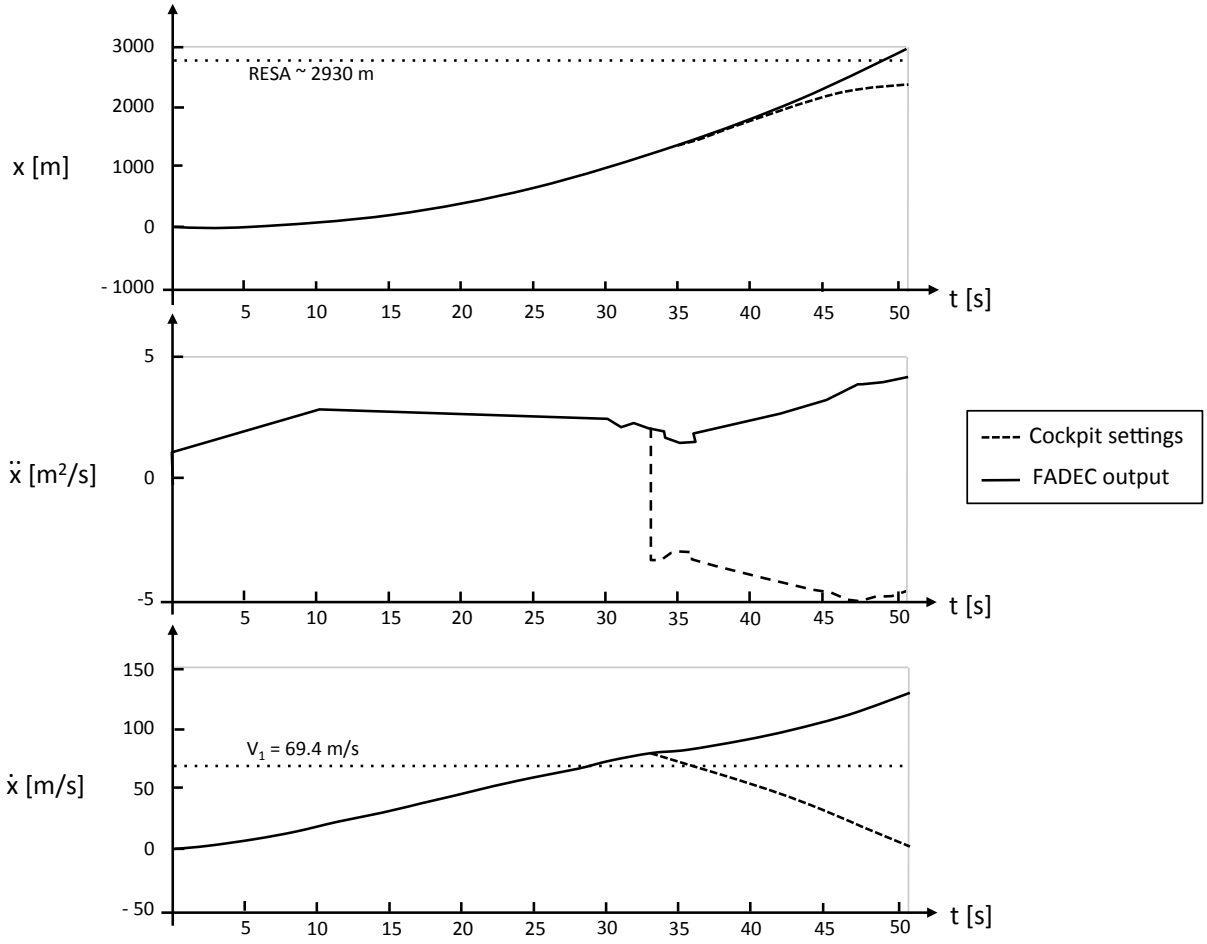

Figure 5.10 Position, acceleration, and velocity of the aircraft as per FADEC output and as corresponding to the cockpit inputs provided by the pilots

The plots of Figure 5.10 can be computed by "sampling" the hazard level, which in this case corresponds to the state itself (plus the value of the acceleration), at

---

[16] Note that the conditions that unfolded during the Learjet accident sequence were not as dire as the ones predicted in Figure 5.8, which considered a worst-case scenario with severely compromised breaking capabilities beyond those encountered in the actual sequence. The plots of Figures 5.8 and 5.9 provide thus a conservative estimation to better guide the RTO decision making and account for proper safety margins that should not be exceeded for safe procedures.

different nodes in the system (e.g., the filtered and unfiltered inputs). For instance, it is generally beneficial to check consistency of the expected controls to be executed on the system at the operator level and at the sensor/data bus level, especially at interfaces between different computers and digital components. When diverging states are discovered, the operators should use extreme caution in handling safety interventions. In fact, discrepancies in these estimations are indicative of violations of the OID principle. A violation of the OID principle during operations results in a degraded situational awareness, due to the fact that whenever different estimations of the hazard level are available, the operator can no longer rely on them, not knowing which one (if any) corresponds to the actual internal condition of the system. Violations of the OID principle are also indicative that additional instrumentation and sensoring are needed for the system, or that the existing sensoring is malfunctioning (as examined in [Saleh et al., 2014]). On a practical level, it is important that these consistency checks are executed during the design and development stages of a system, to avoid potential violations of the OID principle during operations.

The discrepancy between the outputs (of the two models in Figure 5.3) was in this case dictated by the fact that the different links in the chain of commands that started with the pilots in the cockpit and ended with the FADEC output were acting on inputs that were considered competing with each other (e.g., call for TR by the pilots and squat switch indicating AIR mode), and from requirements that did not support unconsidered scenarios (e.g., missing signal from squat switch treated as spurious AIR mode), which reflect the missing implementation of Eq. (5.5).

It is straightforward to compute the discrepancy between each component of *H(t)* and $\widehat{H}(t)$ of Eq. (5.9). For instance, Figures 5.11 and 5.12 show the discrepancy for position and acceleration. As noted earlier, the fact that a discrepancy is present constitutes a violation of the first OID property (detected violations are shown in gray bars in the figures).

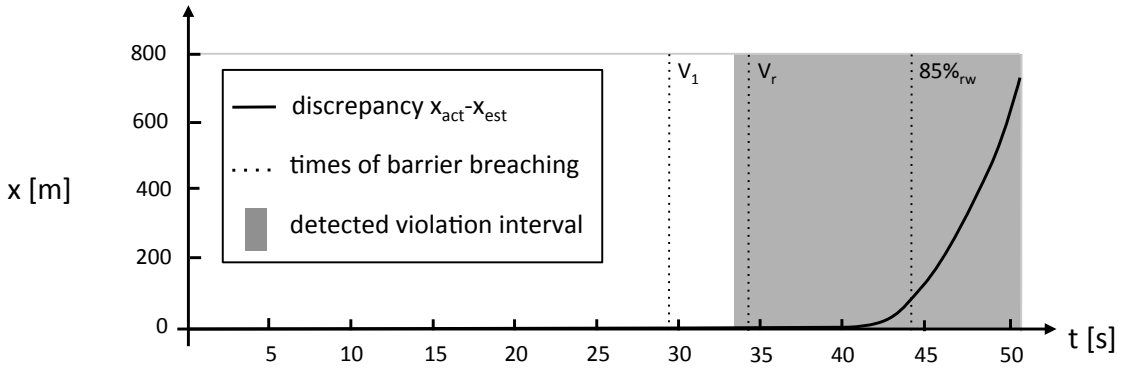

Figure 5.11 Position discrepancy, detection of OID violation, and times of barrier breaching

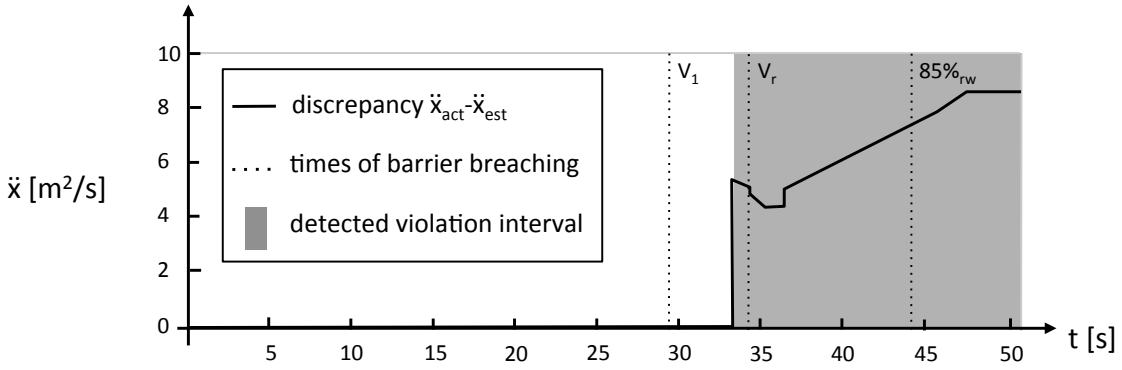

Figure 5.12 Acceleration discrepancy, detection of OID violation, and times of barrier breaching

The second OID property analyzes whether such discrepancy is increasing in time, and puts it in relation to the breaching of the defense barriers placed along the accident trajectory (basically analyzing the conjunction of OID and DID). For the case in Figures 5.11 and 5.12, three operational takeoff checks are considered, provided by: reaching the decision speed V_1; reaching rotation speed V_r; reaching 85% of the runway[17]. The times at which each barrier is breached are superimposed to the

[17] The FAA [2005] advises to choose runways that exceed by 15% the actual runway length required for takeoff by the aircraft.

discrepancy plots. As noted in [Favarò and Saleh, 2014], it is very important that proper warnings and alarms are set off and triggered by the breaching of subsequent barriers along the accident trajectory. Such warnings work toward improved situation awareness, and toward zeroing out the discrepancy between the actual and the estimated hazard level.

The superposition of barriers' breaching times with the dynamical behavior of the discrepancy of the hazard level can be instrumental for understanding where proper sensoring and warning signs are needed most. For instance, Saleh et al. [2014a] analyzed violations of OID in relation to malfunctioning sensoring of a raffinate oil tank tower that led to a LoC-type of accident using a hazard level such as the one analyzed in chapter 3. The tower was instrumented with sensors that only covered five of the 170 feet high tower. Beyond the simple recommendation to include further instrumentation, better informed decisions regarding the re-design and re-engineering of the system can be obtained by examining where steeper jumps in the discrepancy of actual and estimated hazard level occurs, and positioning additional sensors at the heights corresponding to those jumps (as dictated by and according to Eq. (3.8)).

Finally, Figure 5.13 provides a screenshot of the OID property implementation in Simulink. When a user wishes to adopt a particular safety constraint for the analysis of a different system, little changes are required for its implementation. Whenever the same constraint formulation is used, the only required change is the danger index *H(t)* used as input to the property.

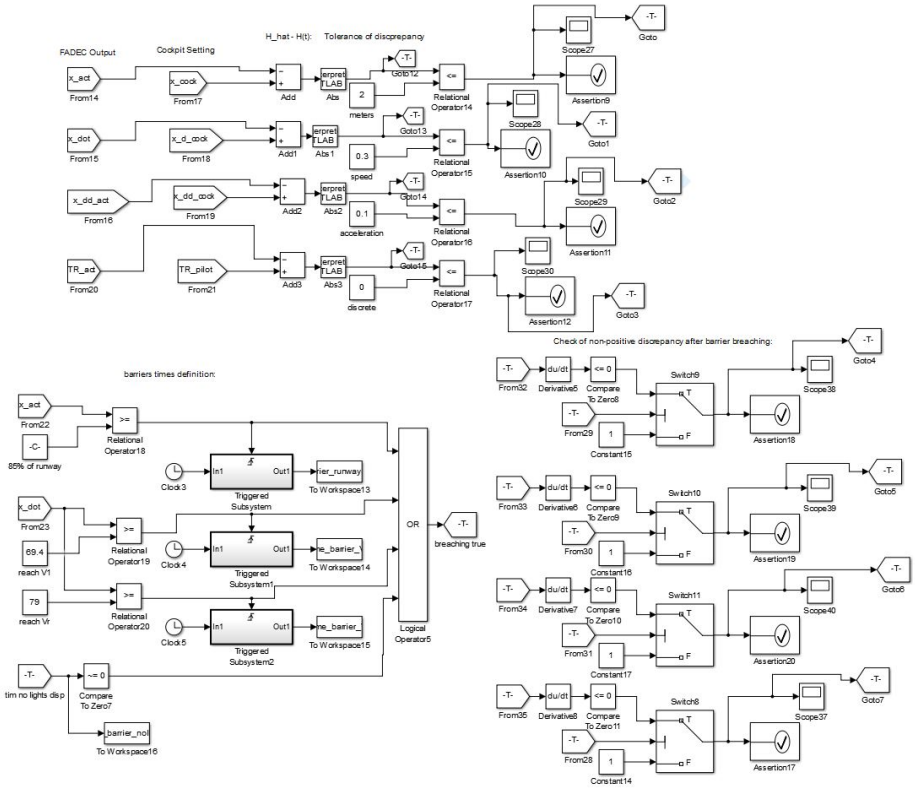

Figure 5.13 TL implementation of the OID property in Simulink

5.3 Insights and Advantages Enabled by the Approach

The case study showed how hardware, software, and operators' control actions and responses can be integrated within the framework for the proposed case. Software played a key role in the escalation of the accident sequence, which was analyzed in detail.

The framework and analytical tools here developed are meant to guide safety intervention (both online and offline), and to dynamically support in real-time operator's situational awareness and decision-making regarding emerging hazardous situations. As such, it already adds value and an important complementary perspective

to traditional approaches to risk assessment and system safety, which do not support real-time use to inform the decision-making or to guide online safety interventions[18].

On a general level, several advantages were highlighted deriving from the adoption of TL in conjunction with the proposed model-based framework. Those included:

- The use of TL allowed more complex expressions of *H(t)* that include temporal operators. Explicit considerations of ordering and timing of events, faults, or sequence of states can thus be accounted for in the definition of *H(t)* itself. This provides significant benefits for expressing complex conditions or situations in a compact form, which is exceedingly difficult to render based on state variables alone.
- The expression of TL safety properties is independent of the specific hazards functions *H(t)* of interest. While hazard functions are specialized and tailored to particular contexts, the TL safety properties are agnostic to the underlying system. They can be conceived of as elements within a broad library of safety properties that requires small adaptation efforts for the analysis of different dynamical systems.
- By leveraging a language that is typical of software systems, the proposed approach allows the integration of both software and hardware components within the same framework.
- By leveraging a formal language, the approach allows the automatic generation of warning signs (e.g., the display of error messages) whenever constraints are violated, or whenever critical thresholds for H(t) are about to be (b)reached.

[18] For a more detailed comparison with traditional PRA and DPRA capabilities see Appendix B.

This is an important capability for their online use, to support the operator's situational awareness and sensemaking of the system conditions and the timely execution of safety interventions.

On a particular level, the application of the proposed framework to the specific Learjet case study enabled to uncover important findings beyond those proposed in the official accident investigation, and can be summarized as follows.

- The analysis of the fail-safe safety property helped understanding an important condition predicated on airspeed thresholds needed to "debug" spurious AIR mode signals from the squat switches. By augmenting the flawed FADEC logic with a check for such condition it is possible to prevent the FADEC from wrongly estimating that the aircraft is in AIR mode, and remove a lurking accident pathogen from the system.
- A danger index in support of online pilots' decision-making for rejecting a takeoff was devised. This index depends on the distance necessary to bring the aircraft to a full stop and accounts for a region beyond the end of the runway, and before obstacles are encountered, to be used as safety area. Mappings for best-case and worst-case scenarios (depending on how compromised the aircraft performance/ braking capabilities are) could have informed the pilots of the Learjet and advised against RTO initiation.
- Important violations of the OID principle during the Learjet accident sequence were highlighted, which contributed to a degraded situational awareness on the part of the pilots. To counteract such violations it is of paramount importance to set up consistency checks among different samples of the hazard level captured at different subsystem nodes/interfaces, for instance comparing the system output resulting by different sets of inputs. For the Learjet case, two sets of

inputs were used: those "filtered" by the FADEC subsystem, and those "unfiltered" that bypassed the FADEC and acted upon the inputs provided by the pilots in the cockpit.

As noted by Leveson [2004b], "many of the problems found in human automation interaction lie in the human not getting appropriate feedback to monitor the automation and to make informed decisions". This was certainly the case of the present case study, and the analysis showed how the proposed approach can in this case better inform both the on-line decision-making, and the off-line system design during development stages, to ensure proper feedback is provided to the operators regarding the system internal conditions.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This chapter concludes the thesis. It is structured in the following way. Section 6.1 presents a summary of the contributions accomplished by the proposed framework. Section 6.2 explores the potential for further research opportunities and expansions related to the material here presented.

6.1 Summary of Contributions

The end-objective of this work was to contribute to improving (dynamic) risk assessment and accident prevention. To this effect, a synthesis of key limitations of PRA was provided first, together with the improvements currently proposed in the literature (Chapter 2). These issues constituted the main motivation for the present efforts. I then made the case for model-based approaches and the use state variables, in particular in relation to the development of danger indices and the monitoring of hazard dynamics for improved risk assessment. This allowed to introduce a novel safety supervisory control framework. The development of its analytical tools, and the notion of hazard temporal contingency for dynamic risk assessment and for guiding safety interventions to improve accident prevention were presented as one of the two ingredients that constitute the approach (Chapter 3). The second ingredient was that of Temporal Logic (TL) and its use for the verification of safety properties predicated on the notion of hazard level (Chapter 4).

The framework and analytical tools here developed were grounded in Control Theory and Computer Science.

On the one hand, Control Theory inspired the use of the state-space representation in modeling dynamical systems. The use of state variables allowed the

definition of hazard levels or danger indices, which measured the "proximity" of the system to adverse events. Furthermore, I showed that the adoption of state-space formalism enables the estimation of the times at which critical thresholds for the hazard level are (b)reached. This estimation process provides important prognostic information and produces a proxy for a time-to-accident metric or advance notice for an impending adverse event. These hazard coordinates were displayed in a hazard temporal contingency map to support operators' situational awareness, and help them prioritize attention and defensive resources for accident prevention. The monitoring of hazard levels and the estimation of the time window available for safety interventions provide important feedback for various stakeholders and decision-makers 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).

On the other hand, Computer Science inspired the use of TL for the specification of safety properties towards the creation of an *automatic safety verification* process. Properties expressed in TL are agnostic to the specifities of the system under consideration, and create a pervasive and universal library of safety properties that can be used for the analysis of any dynamical system. Moreover, Temporal Logic allowed to overcome some of the time-related limitations of traditional PRA. Through the adoption of TL, specific considerations on temporal ordering can be included directly in the analytical definition of the hazard level. Additionally, the formal language of TL and the choice of Simulink as simulation environment allowed to model both hardware and software components within the same framework and to automatically set up error messages displays and alarms to warn the operators of violations of the TL properties.

The capabilities of the framework were displayed through the detailed analysis of a case study involving a runway overrun (Chapter 5). The integrated framework showed that the proposed approach informed important recommendations for new TL

safety constraint that could have prevented the hazardous situation, in this case a rejected takeoff following tire explosion, from turning into a fatal accident. Moreover, novel metrics for online support of pilots' decision-making were developed, which can also better inform accident investigation and provide recommendations for the prevention of similar occurrences in the future.

The work here presented sought to augment the current perspective in traditional risk assessment and its reliance on probabilities as the fundamental modeling ingredient with the notion of temporal contingency, a novel dimension by which hazards are dynamically prioritized and ranked based on the temporal vicinity of their associated accident(s) to being released.

The proposed approach has the potential to eliminate the reliance on expert opinion for assessing the probabilities associated with the sequences of adverse events and conceiving of accident scenarios. However several new challenges are raised. For example, more reliance is placed on the analysts who develop the model of the system and identify the hazard levels of interest (i.e., high level of modeling expertise is required, as well as in-depth knowledge of the system). Note that the choice of the $H(t)$ functions of interest can be informed by the particular safety requirements imposed for the system. Another challenge for the practical implementation of any model-based approach to system design and operation is related to the proliferation of the number of states to consider (known as the state explosion problem). This problem requires careful consideration of model order reduction and computational implementation (especially for real-time hazard monitoring and estimations). Finally a set of challenges are raised in relation to the verification and validation of such analytics, as well as the human factors considerations in using/interfacing with the proposed safety supervisory control approach.

While more research is certainly needed, I believe the prospects and potential advantages offered by the framework and tools here introduced outweigh the challenges

they raise, and they constitute a rich area for further development. A case study was here presented as "proof-of-concept" of the proposed theoretical developments, and the tools presented allow for the creation of a test-bed that, with additional research, can provide important complementary insights to those provided by traditional approaches to risk assessment. Several research paths forward are possible and some were outlined throughout this text, and are summarized in the next section. Some authors have recently argued for the need to leverage automation for risk assessment and management; this model-based approach provides one step in this direction. I hope this work (and the ensuing publications) will enrich the intellectual toolkit of risk researchers and safety professionals, and will invite further contributions from the community to improve (dynamic) risk assessment and accident prevention.

6.2 Future Work

Two paths can be envisioned to guide future work and extensions of the proposed approach. The first involves detailed applications of the framework for additional benchmark and proof-of-concept examples. The second path involves further development to better exploit the benefits deriving from the application of the framework and from the use of Temporal Logic.

This second path is better analyzed next.

6.2.1 Monitoring vs. Model Checking: Towards Automated Safety Verification

In the present work the verification of compliance with the TL safety requirements was achieved through direct monitoring of the properties (implemented using the Design Verifier toolbox in Simulink). A second approach, other than monitoring, for the formal verification of TL properties exists, and it is that of model checking. Model checking verification is grounded in mathematical abstraction and automata theory, and does not require running a simulation or monitoring/displaying a

property directly. Model checking allows to automatically verify compliance of TL safety properties, and provides immediate counter-examples whenever violations are encountered (i.e., providing examples of unthought accident scenarios). Model checking can greatly help towards ensuring an exhaustive verification process. To this same end, note that the case study analyzed a single scenario, with pre-specified inputs to model the accident sequence of the Learjet 60. Multiple scenarios can be easily handled without changes to the model (e.g. providing different sets of inputs with switches to analyze one case at a time). Additionally, Simulink Design Verifier has the capability of automatically generating test cases for the system's inputs to extend the model coverage. This process makes use of structural verification techniques to make sure there are no "unused paths" in the system model, with the capability of integrating formal methods within the framework just presented.

A second idea is related to the offline use of the TL safety constraints for the validation of a design during the development stages. The use of TL allows to set up an automatically verifiable machinery, which in offline applications does not require to display and directly examine plots of the hazard level such as the ones presented in chapter 3 and chapter 5 (e.g., Figures 3.5, 5.9, 5.10) for the verification of compliance of the safety properties. This capability is important during development stages (i.e., offline applications) since in general hundreds of scenarios with different inputs (e.g., different profiles of thrust and the call for TR engagement by the pilots at different times) are considered at a time. In these situations, it is not possible for the analyst to actually display all the values of $H(t)$ for the different scenarios tested, and an automatic procedure is needed to warrant the analyst's attention only for relevant cases. The expression of constraints in TL allows to generate warnings of constraint violations that will warrant close attention and direct $H(t)$-plots analysis *only for unthought of scenarios*, which are the ones that will (most likely) fail the verification process, to validate the system design and/or support system re-engineering.

6.2.2 Displays and Visual Aides Development

A recurrent problem with modern techniques for probabilistic risk assessment and DPRA in general is that of output post-processing, as was analyzed in chapter 2. This issue concerns the display and generation of an output for ready and easy risk communication [Zio, 2014].

Although beyond the immediate scope of the present work, it is worth noting that the proposed framework lends itself to the development of intuitive control panels in support of safety-related decision-making. Intuitive displays that inform the operators of the system condition (in terms of the associated hazard levels) and of where attention is needed most can guide actionable insight for better-informed safety interventions.

Simulink allows to integrate and interface good visualization aides with the analysis carried out in this work. For instance, note that in general there are a few TL properties that the analyst wishes to verify for a range of different hazard levels. A control panel such as the one of Figure 6.1 can be set up to warrant the close attention only for those hazard indices that violates a principle.

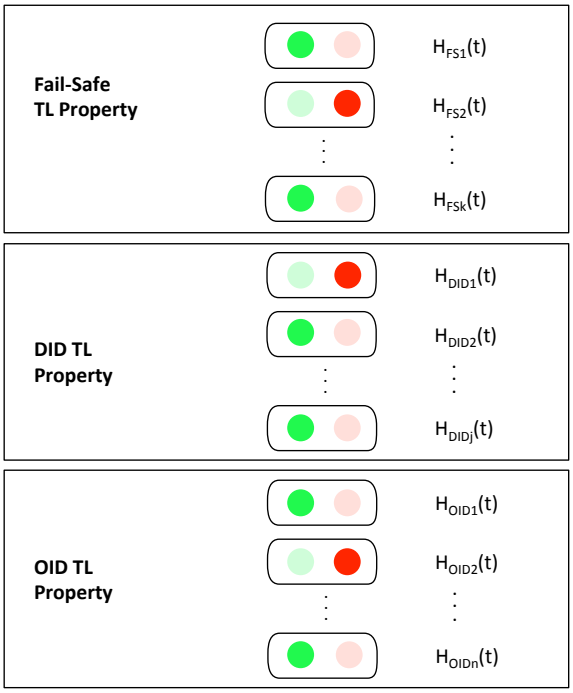

Figure 6.1 Illustrative "control panel" for monitoring the verification of TL properties against multiple hazard levels. The red circle is indicative of violated properties, the green circle stands for compliance. Hazard levels that indicate a violation of the property warrant closer attention (e.g., H_{FS2}, H_{DID1} and H_{OID2} in the Figure)

Moreover, when the analyst wishes to monitor multiple hazard levels at a time, different visual aides can be developed, for instance similar to the radar plot provided in Figure 6.2, which shows the concurrent monitoring of eight danger indices and is capable of providing a direct snapshot of "hot spots"/ hazardous conditions in the system.

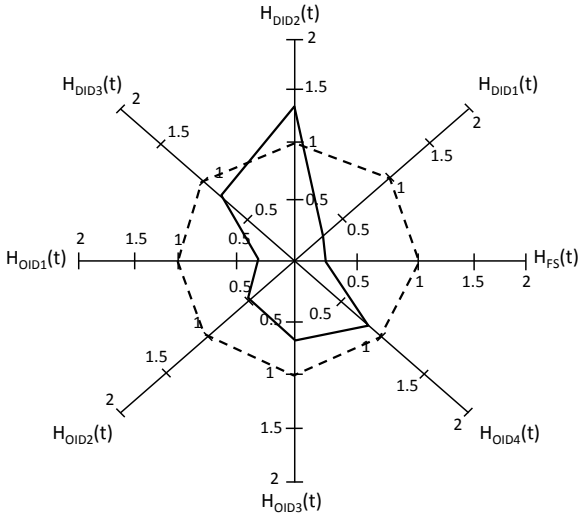

Figure 6.2 Illustrative "radar plot" for concurrent monitoring of multiple hazard levels. Dotted line represent the accident threshold for each index, solid line represents the instantaneous value of each index

I believe the ideas brought forward in this section are worth investigating as promising venues for future contributions to the risk and safety communities. They constitute good topics in the context of cognitive engineering and human factors. Design and experimental testing of displays based on the ideas presented here can work towards improvement of the operators' situational awareness in critical situations.

APPENDIX A

NOTES ON HUMAN SUPERVISORY CONTROL

The term "supervisory control" has been used in the past to describe the layout and role of software agents as an aid to collect and display system measurements, towards detection of flawed performance and failures. In particular, *human* supervisory control addresses the relationship between a human and a machine (or cyber-physical system) interacting with each other "to transform data or to produce control actions" [Sheridan, 2012]. The "human" qualification was introduced to highlight the fundamental role of the operators in relation to the sensemaking of the information provided by the software agents.

Supervisory control is found in a broad range of applications, from obstacle avoidance in the military context, to therapy and dosage control in the medical one. Over the past three decades the research on human supervisory control has focused on important aspects related to: the study of tradeoffs between the level of automation and the need of human operators in the loop (e.g., to whom assign authority and when); the design and monitoring of intuitive displays and, more generally, the strategies on how to provide the feedback collected by the software agents to the operator in a clear and concise manner (e.g., integrated displays in support of decision-making); regulations and policy regarding human supervisory control, and its social implications.

The generality of the idea of human supervisory control was developed by Sheridan and colleagues at MIT in the 1960s [Sheridan, 1960; Ferrell and Sheridan, 1967]. To date however, the idea of human supervisory control is still little understood and leveraged in a formal way [Sheridan, 2012]. There exists a multitude of models for the implementation of supervisory control (see [Sheridan, 2012] for a high-level review). In general, five functions are to be performed:

1) *Planning*: first phase for off-line understanding and planning of what task to accomplish and how;
2) *Teaching*: second phase for off-line programming of the software agents, which are taught what was planned;
3) *Monitoring*: third phase for on-line detection of failures (or confirmation of nominal conditions) through (possibly automatic) monitoring of the state information;
4) *Intervention*: fourth phase for on-line intervention on the task to specify a new goal for the system;
5) *Learning*: final phase for the off-line analysis of the lesson learnt from the experience, to improve future performance.

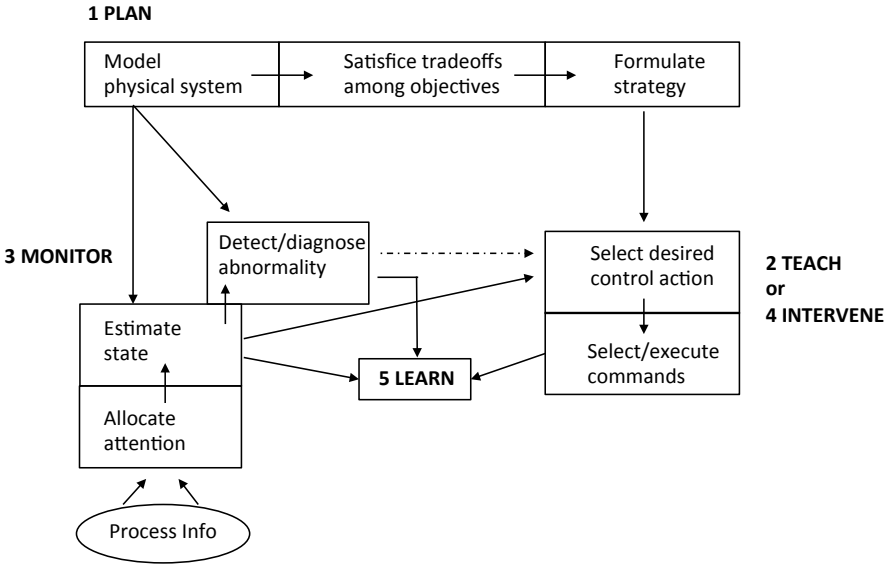

Figure A.1 Flowchart of the five supervisory functions. Adapted from [Sheridan, 1992]

The five supervisory functions are represented in the flowchart of Figure A.1. Sheridan [2012] frames these function within three nested loops (not shown in Figure A.1 so not to further clutter the figure):

(i) An inner loop of the monitoring function within itself that invites further monitoring and investigation whenever abnormal situations are detected;

(ii) An intermediate loop between the intervening and the teaching functions, since the intervention function ends with the specification of a new goal that has to be programmed/taught back to the computer;

(iii) An outer loop that informs new tasks planning based on the learnt experience.

The proposed framework targeted a specific application of supervisory control for model-based hazard monitoring and the verification of safety properties, towards the sustainment of system safety, improved accident prevention, and to guide and inform safety interventions. Although different techniques and tools were used in this work (in relation to the use of state-space representation for the definition of danger indices and the adoption of temporal logic to express safety constraints), many of the considerations brought forward in this work can be related to the realm of supervisory control.

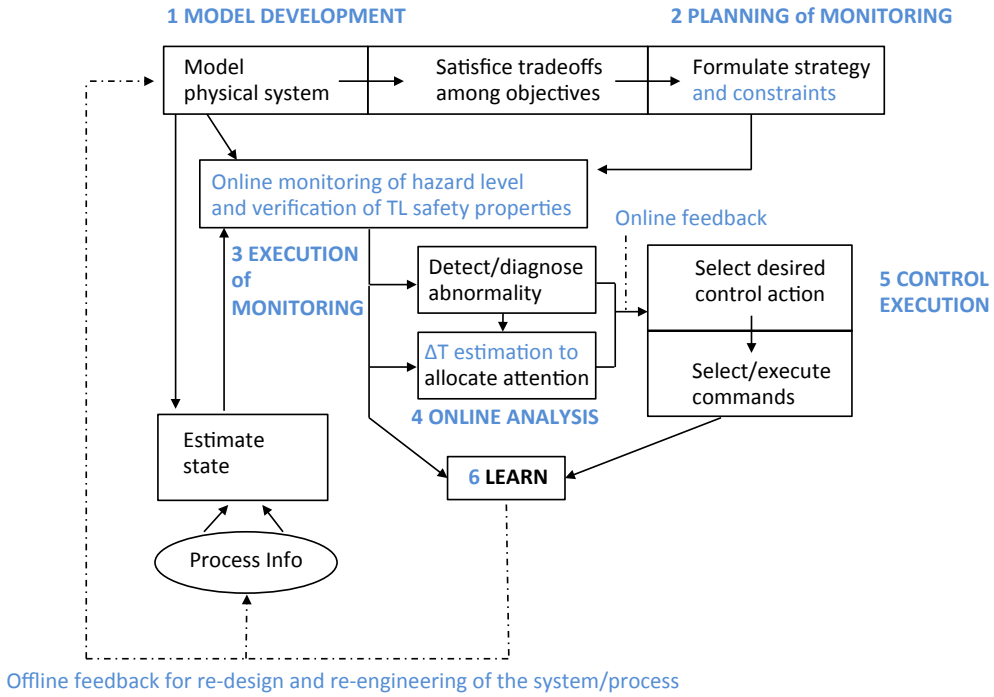

Figure A.2 Adaptation of the supervisory flowchart to the proposed framework. Changes shown in blue

The framework modeled and presented in Figure 1.2 can be compared to that of Figure A.1 proposed by Sheridan. A few distinctions can be highlighted. Figure A.2 shows a possible adaptation of the flowchart proposed by Sheridan to the work and framework here presented.

It is possible to highlight differences in three major areas:

1) **Functions/phases performed**: Figure A.2 highlights six phases, and divides them in accordance to the presentation of the work carried out in the thesis. The original Planning phase of Figure A.1 is now divided into the *"model development"* and the *"planning of the monitoring effort"* phases. The off-line planning of the task to accomplish is thus translated into: (i) the creation of the mathematical model for the dynamical system under consideration; and (ii) the identification of the hazard levels of interest, informed by the specific safety requirements and constraints imposed for the system. Additionally, the original Teaching phase was incorporated inside the first two functions, given the fact that the system controller synthesis is included in the model development phase and that no further software programming is required once the constraints are translated in temporal logic and the hazard levels are identified. The third original function of Monitoring now translates into the *"monitoring execution"* phase and includes the process of monitoring the danger indices as well as verifying the TL safety properties. I included a novel function termed *"online analysis"* to capture the supporting role of the hazard temporal contingency map to allocate the attention of operators and to support decision-making for safety interventions. The original Intervention phase now assumes the name of *"control execution"* given the proposed control-based framework. Similarly to the original flowchart, the approach ends with the *"learning"* phase.

2) **Internal wiring of the blocks**: The blocks of Figure A.2 are now wired differently, according to the process examined in the thesis. Important differences are related to: (i) the removal of the wiring between the strategy formulation (original Planning phase) and the execution of the control actions (original Teaching and Intervene phases); and (ii) the location of the "attention allocation" block, now moved as a final step before the control actions execution rather than as a first step to process the incoming system information. Both changes reflect the central and key role of the hazard-level monitoring, with the estimation of the time-to-accident metric, and of the verification of the TL safety properties, to inform safety interventions.
3) **Feedback loops definition**: Figure A.2 highlights the existence of two feedback loops. Rather than dividing the loops by "location" as in the original formulation of supervisory control, I prefer to divide them in the two categories of online feedback and offline feedback. Offline feedback (i.e., for offline safety interventions) acts on the whole process through changes in the model development and the monitoring planning phases, and through its effect on the process info (e.g., the decision on which quantities to measure in the system or to select as system output). Re-design and re-engineering of the system are examples of offline interventions, and so is the adjusting of the hazard level definition and of the ensuing constraints. Online feedback (i.e., for online safety interventions) informs the execution of the control actions based on the diagnostic and prognostic information deriving from the monitoring execution and the online analysis phases. Online interventions are reflected in the trimming and adjustments of the control matrix Ψ for the hazard equation.

Many authors regard failure detection and diagnosis as the most important human supervisory role [Moray, 1986]. While recognizing its importance, at the same

time this work added a prognostic dimension, through the estimation of the time-to-accident metric. This addition also enabled the prioritization of the interventions, based on the ranking of emerging hazards in the temporal contingency map. This is an important ingredient for allocating operators' attention. Means for discerning among the multitude of information coming from different sources are often advocated in supervisory control, and this work provided a novel metric to this end.

Finally, a separate mention should be given to the problem of observability-in-depth (OID). Other authors have highlighted the importance of being able to diagnose the lack of observability in a system [Ferris at al., 2010]. Especially in the context of interfacing humans with automation, it is important to guarantee that both components are acting on the same understanding of the system behavior (i.e., ensuring that there are no discrepancies between the actual and the estimated conditions). In human supervisory control, computer-based observers can work as an aid to a human supervisor according to the process described in Figure A.3. This process leverages the creation of a model for the process under examination that serves for direct output comparison with the actual system.

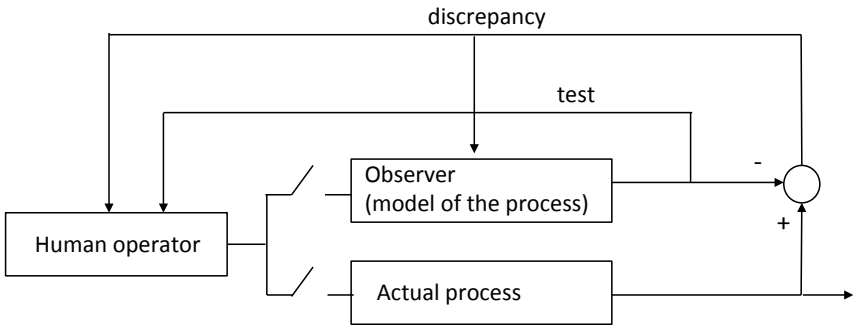

Figure A.3 Computer-based observer as an aid to the supervisor. Adapted from [Sheridan, 2012]

Sheridan [2012] advocates the use of computer-based observation to estimate quantities that are not directly measured in the system. In the proposed framework, the OID principle tackles many important aspects in this regard. First, it works toward ensuring that proper state feedback is in place (so that the approach not only accounts

for a comparison of the system output, as in Figure A.3, but also achieves a comparison of the estimated state). Secondly, it ensures that in case of discrepancy, proper warning signs and alarms are set off whenever safety barriers are (b)reached, in support of the operators' sensemaking. Moreover, this work highlighted the importance to apply the principles behind OID also to *different model-based estimations of the hazard level* (i.e., two or more estimations coming from the model of the plant itself). In other words, in addition to what described in Figure A.3 (comparison with system output) the work also reviewed applications of the OID principle within the process model block of Figure A.3 in isolation. Ensuring consistency checks in the estimations of the hazard level at different nodes of the system model served as an important indicator of proper instrumentation and sensoring of the system. Finally, chapter 5 highlighted the importance of executing the consistency checks during the design and development stages, so not to incur in degraded situational awareness scenarios during system operations.

APPENDIX B

A COMPARISON WITH PRA TECHNIQUES

The application of PRA techniques for assessing the risk associated to a given system is related in many cases to the need of certifying it according to specific regulations and policies.

This is the case of many aerospace systems, which abide by very strict certification procedures. The ARP4754 outlines the recommended practices for the development of civil aircraft and system in support of their future certification. The tools of PRA (e.g., fault and event tree analyses) have a fundamental role in this process, which is schematized in Figure B.1.

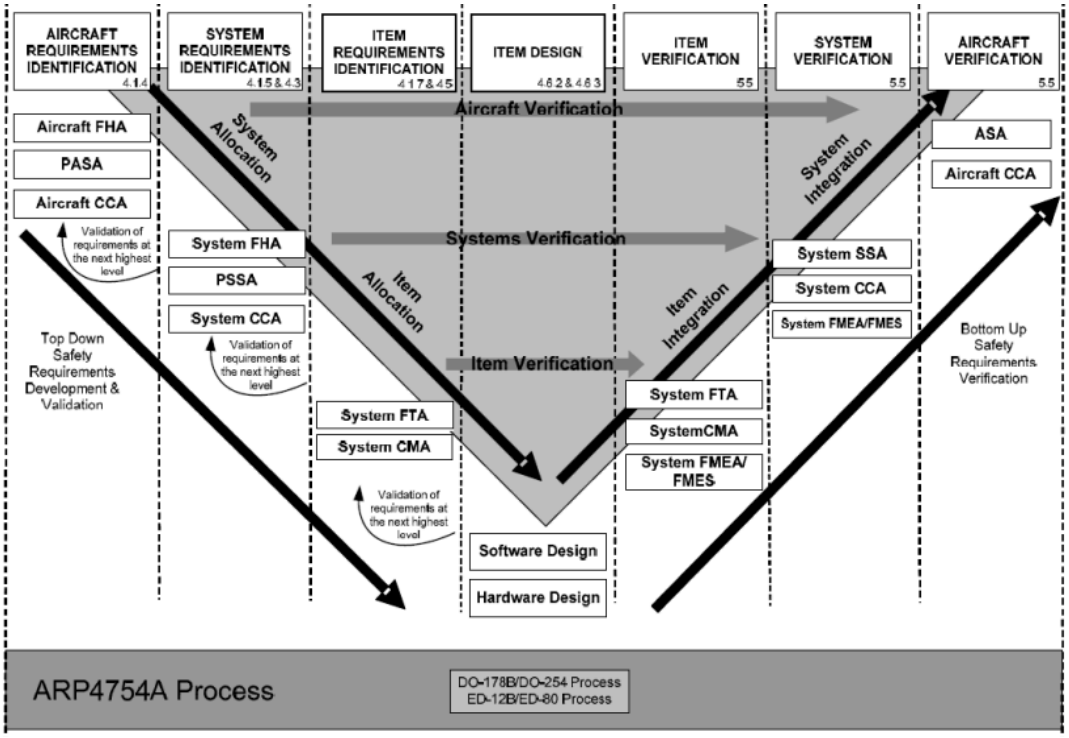

Figure B.1 Integrated process for aircraft safety design [ARP 4754]

Fault tree analysis (FTA), reliability block diagrams (RBD) and failure modes and effects analysis (FMEA) have an important role both for requirements development and identification, and for system verification. Generally, a complex system such as an aircraft undergoes a physical and a functional decomposition, and requirements are then set up for each component.

To compare how the proposed approach differs and relates to standard PRA techniques, it is possible to briefly analyze how the PSSA and SSA ((preliminary) system safety analysis) processes work and how they makes use of PRA tools.

In a general perspective, both (P)SSA and the proposed approach start with the definition of requirements for the system, which are then verified in order to validate the specific design. In the proposed approach, requirement definition is expressed in temporal logic through quantitative constraints on the hazard level H(t) and the definition of criticality thresholds for safe behavior. In traditional approaches requirements are defined in terms of accepted probabilities of failures. For instance, to draw a parallel with the case study analyzed in this work, for the FADEC subsystem, which combines both hardware and software components, requirements are derived by several standards such as the RTCA/DO-160/-178-C/-254, 14 CFR/AC 33.28 and present the following exemplary format:

" FADEC mode failure during takeoff and landing shall be less than 3.5E-9 and during cruise shall be less than 3.5E-7"

In other words, this type of requirements set up a limit for the "maximum tolerable failure" [NASA, 2002]. The problem now revolves around the estimation of the probability of failure of the component of interest. Traditional PRA relies on the physical and functional decompositions of the component to identify possible failure

modes and mechanisms. The overall probability of failure is then obtained by aggregating together the probabilities associated to each failure mode.

Fault trees are one of the means for assessing this overall probability, and are among the most used PRA tool in industry. FTA is a top-down deductive approach, where a top-level event (e.g., the failure of the component of interest, or generally any undesired event) is decomposed and analyzed using Boolean logic to combine a series of lower-level events down to basic or primitive events that are no further decomposed. The probabilities of the basic events are assessed based on field data or on expert judgment. Those probabilities are then combined according to the wiring of the logical gates and the events in the tree, up to the top-event failure. The requirement satisfaction is then analyzed by comparing the top-event probability with the maximum allowed value set up by the regulatory/certification agency. A typical fault tree is represented in Figure B.2.

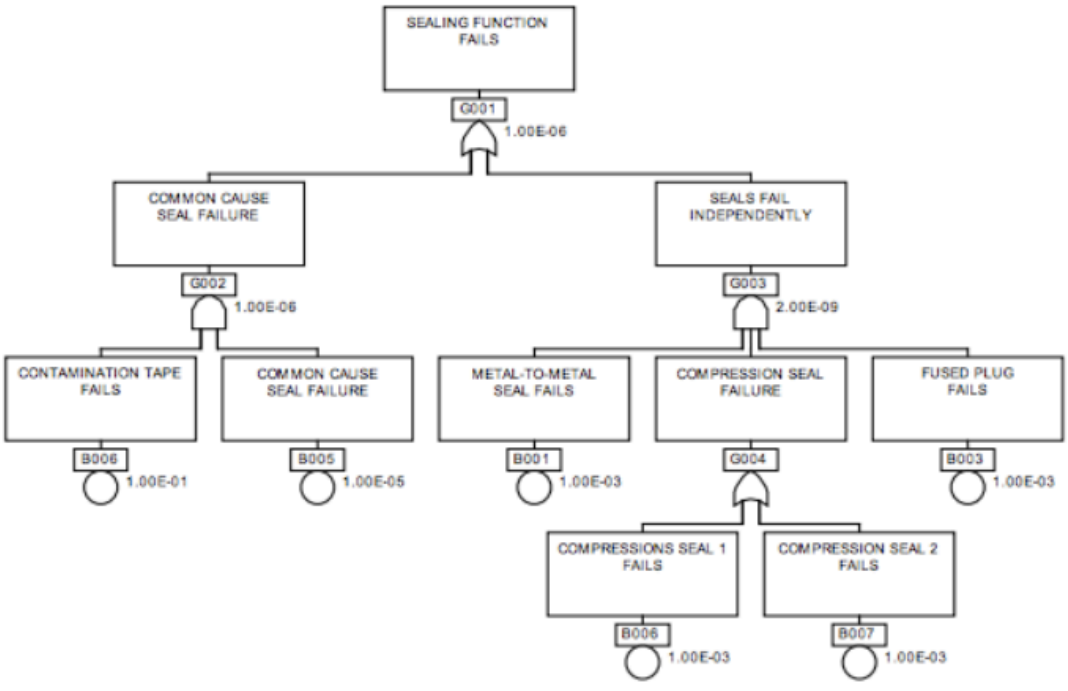

Figure B.2 Example of typical fault tree, [NASA, 2002]

Fault tree analysis can be used for requirement validation as well as a benchmarking tool to compare different designs. For instance, the engine control unit (ECU) of the FADEC subsystems is generally dual redundant, and the effect of this redundancy can be assessed by comparing two separate fault trees: one that considers redundancy and one that does not. The overall probability associated to the top-event lets then the designer know how positive the effect of redundancy is in order to satisfy a specific requirement.

The process briefly outlined before is fundamental for the certification of the system of interest. However, it does not provide any insight on online and real-time risk assessment during system operation, and it rarely provides insight on what safety features to embed in the system should a specific requirement be violated. The proposed approach complements this view by adding the dimension of temporal contingency, which guides online safety interventions, and the verification of safety constraints that can guide offline interventions for assessing the need to include additional safety features in the system. Moreover, PRA approaches rely on the existence of extensive field data for the systems of interest, which is rarely the case for new and avant-garde systems. Whenever field data is not available, these processes rely on the opinion of experts of the field, who estimate plausible values for the missing probabilities based on their personal experience. Relying on different experts leads to sometimes contradictory results, and is generally accounted for as one of the main limitations of PRA (both in its static and in its dynamic counter-part).

Table B.1 shows a summary of the capabilities and benefits of the proposed framework, when compared to traditional PRA and to the recently proposed DPRA.

Table B.1 Comparison of the approaches

Approaches / Elements	**Static/Traditional PRA**	**DPRA**	**Model-based SSC and hazard monitoring**
Risk calculation	Probability and scenario based	Probability and scenario based (possibly time-dependent probability)	Based on the notion of temporal contingency for a given hazard level function
Inclusion and modeling of physical phenomena (e.g. failure physics) and external environment	NO	YES	YES
Recovery modeling and handling	NO	YES	YES (can be included in dynamical model description)
Human Factor Analysis	HRA but no model for human response during an accident	Can include model of operator response in simulation	Can include model of operator response in simulation
Inclusion of Organizational factors	NO	NO (not in standard frameworks anyway)	NO
Provides insight on real-time risk mitigation and risk management strategies	NO	NO	YES (based on the monitoring and analysis of the hazard level dynamics)
Inclusion of software and interfaces handling	NO	YES (in part)	YES
Capable of handling scenario-based results	YES	YES	YES
Possibility of state exploration (i.e., verification not based on scenario)	NO	NO (but state exploration efforts for this approach are under research)	YES (through model checking, at least for the software sub-blocks)
Requires expert opinion and judgment for model's inputs computation (subjective data collection)	YES	YES (even though in reduced amount)	NO
Required development effort	Medium (considerable manual effort)	High	High (can potentially be automated)

The approach proposed is comparable in several respects with DPRA. Both DPRA and the proposed approach provide additional and complementary insights to traditional PRA. As highlighted in the thesis this comes from the analysis of time-

related issues modeled in the system's dynamics (hence allowing the analysis of failures ordering and time-dependent-performances) at the expenses of a higher computational effort.

The proposed approach eliminates the reliance on expert opinion and judgment for the calculation of the probabilities associated to the transition from nominal to off-nominal states. Conversely, more reliance is placed on the analyst that creates the model of the system and identifies the hazard levels of interest. This last consideration is related to the limitations and challenges that need to be addressed for the development and the practical implementation of the approach proposed in this thesis to a workable industry standard. Those include the following issues. First, model-based approaches are subject to the problem of state explosion. This problem can be potentially addressed by only considering the state-space models for the subsystems of interest and by interfacing models of different formats within the same simulation environment (e.g., only consider the transfer functions for elements that are not relevant for the analysis, consider state-charts and truth tables to model software components, etc.). Secondly, the approach requires a good process knowledge and substantial background in dynamical systems' modeling on the part of the analyst, in addition to the need of creative ingenuity to come up with meaningful forms of the hazard level and for the implementation of the safety constraints. Note however, that the choice of the H(t) functions of interest can be informed by the particular safety requirements imposed for the system and furthermore, that the creation of the safety constraints (and their implementation in Simulink) is a one-time effort, as a library of properties can be created and then applied to any dynamical system of interest.

Finally, it is interesting to note that the NTSB report filled out as part of the investigation for the Learjet accident highlights the fundamental role of the under-inflated tires as the main concern for recommendations to the FAA and for improving the certification process. Findings such as the following appear in the report, and show

also how traditional approaches to accident investigation are informed by the certification practices:

> *"The Federal Aviation Administration's legal interpretation that checking tire pressures on a Learjet 60 is preventive maintenance has an unintended negative effect on the safety of 14 Code of Federal Regulations (CFR) Part 135 operations because, according to the provisions of 14 CFR 43.3, a Learjet 60 pilot who is allowed to perform preventive maintenance, such as tire pressure checks, on the airplane for a flight operated under 14 CFR Part 91 is prohibited from performing the checks on the same airplane for a Part 135 flight." [NTSB, 2010]*

> *"The tire design and testing requirements of 14 Code of Federal Regulations 25.733 may not adequately ensure tire integrity because they do not reflect the actual static and dynamic loads that may be imposed on tires both during normal operating conditions and after the loss of one tire, especially if the tires are operated at their load rating, and the requirements may not adequately account for tires that are operated at less-than-optimal conditions." [NTSB, 2010]*

While the importance of this contributory factor (which can be viewed as the immediate initiating event) is recognized, the proposed analysis uncovered important flaws in both the decision-making process for the RTO (together with flaws in the regulatory suggestions related to the V_1 decision speed), and in the logic for the distinction of air/ground mode based on the inputs provided to the FADEC subsystem. The framework proposed in this work allowed novel insights also for accident investigation, that leveraged the notions of safety bounds and danger indices for understanding the critical conditions that unfolded during a particular accident sequence and informed novel decision-making support tools to improve current regulations. The

violation of specific constraints and safety principles also allowed to highlight findings that go beyond the pinpointing of flaws in the certification process and provided important recommendations for system re-design and re-engineering.

APPENDIX C

PRIMITIVES OF CAUSALITY AND THE NOTION OF AGONIST, ANTAGONIST, AND INVERSE AGONIST

The notion of hazard-level was intuitively conceived as the closeness of an accident to being released. 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 end-state A.

The dynamics of the hazard level can be defined in a similar manner to the behavior in time of the failure rate of a component in reliability engineering. For instance, the common (descriptive) bath tub curve (as the one in Figure C.1) describes failure behaviors of a component characterized by three separate regions: a region of decreasing failure rate that models infant mortality; a region of approximately constant failure rate that models random failure; and a region of increasing failure rate that models wear-out.

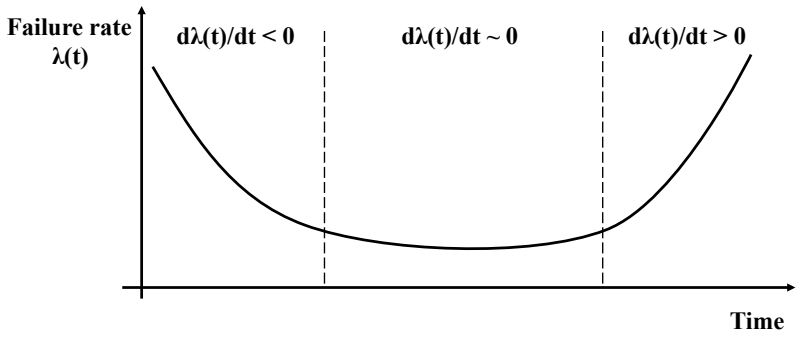

Figure C.1 Bath-tub curve

Similarly, three archetypes of dynamic behaviors for the hazard level can be defined: (i) hazard escalation, (ii) hazard de-escalation, and (iii) hazard constancy. Each

behavior corresponds to a transition from an initial system state to a subsequent state (with the two states possibly coinciding in the case of hazard constancy). This transition is traced back to the interaction of three categories of actions termed *Agonist*, *Antagonist*, and *Inverse Agonist actions*. These categories of actions help expand the terminology used to analyze accident trajectories and provide a better lexicon for describing the hazard level dynamics, which can provide insight towards prevention of similar accidents in the future (as Confucius said: "the beginning of wisdom is to call things by their proper name").

The three categories of actions are borrowed from different contexts:

- The notions of Agonist and Antagonist were originally proposed by Talmy [2000] in the context of cognitive linguistics to indicate the opposing effects of two forces.
- The concept of *Inverse Agonist* is adopted in biochemistry in the context of catalysts-aided chemical reactions where an inverse agonist is a chemical agent that binds to a receptor to induce a biochemical response that is the opposite of the one expected.

In the context of accident causation, the concepts of Agonist, Antagonist and Inverse Agonist actions are related to their effect on the system hazard level for a given accident sequence. Unimpeded Agonist actions push the system state on a trajectory of hazard escalation; if they are sustained over time, they can lead to accident unfolding. Conversely, Antagonist actions can block Agonists and prevent hazard escalation (or prevent further advancement of the accident sequence). Finally, Inverse Agonist actions engage the system in hazard de-escalation. These concepts are formalized as follows:

- An Agonist (indicated by a) is defined as an action applied to the system leading to a transition of the accident sequence towards a higher hazard level (hazard escalation). The hazard level dynamics due to an agonist action can be represented as in Figure C.2, and is symbolically expressed as:

$$a \rightarrow \frac{dH(t)}{dt} > 0 \qquad \text{(C1)}$$

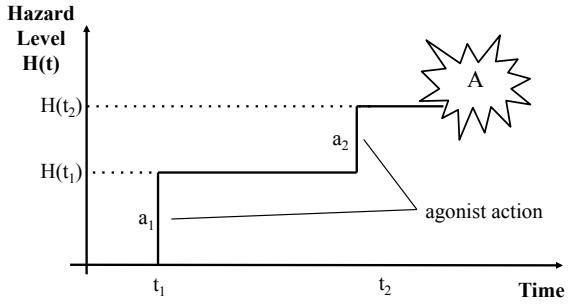

Figure C.2 Hazard dynamics due to Agonist action

- An Antagonist (indicated by $\bar{a}$) is defined as an action applied to the system that blocks an Agonist action. Therefore, the hazard level reaches a stationary point whenever a successful Antagonist action occurs, and the hazard dynamics is blocked (hazard constancy). The hazard dynamics due to an antagonist action is represented in Figure C.3, and is expressed as:

$$\bar{a} \rightarrow \frac{dH(t)}{dt} = 0 \qquad \text{(C2)}$$

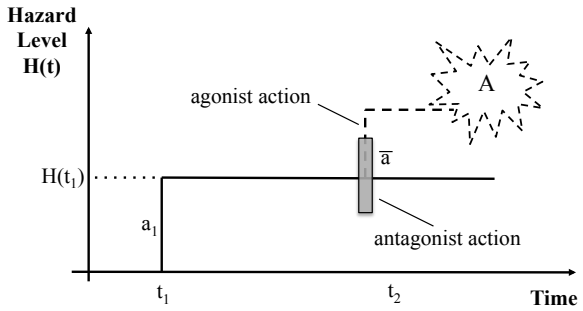

Figure C.3 Hazard dynamics due to an Antagonist action

- An Inverse Agonist (indicated by ia) is defined as an action applied to overcome the effects of an Agonist action, leading when successful to a transition of the state towards a lower hazard level (hazard de-escalation). The hazard level dynamic due to an Inverse Agonist action is represented in Figure C.4, and is symbolically expressed as:

$$ia \rightarrow \frac{dH(t)}{dt} < 0 \qquad \text{(C3)}$$

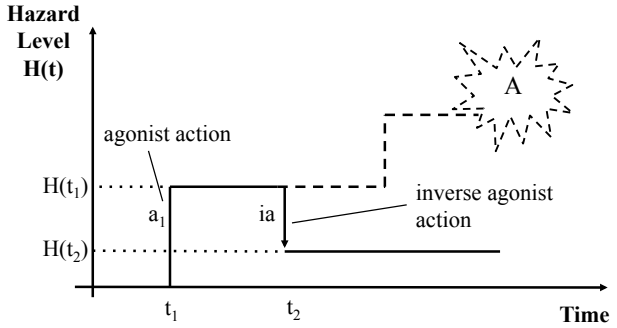

Figure C.4 Hazard dynamics due to an Inverse Agonist action

The notions of Agonist, Antagonist and Inverse Agonist should not be restricted to the idea of physical actions on the system (e.g., pushing an emergency button, activating a water sprinkler, or having a fire wall). Agonist, antagonist, and inverse

agonists can be of different nature, technical (hardware and software), operational, and regulatory. Different types of safety levers exist and can be acted upon to prevent an accident from unfolding (as antagonists and inverse agonists).

The presented concepts enable a rethinking of the traditional notion of causality, detailing accident causation into finer primitives, as presented next.

In his study on force dynamics for cognitive linguistics, Talmy [2000] introduces "force dynamics" as a generalization of the traditional notion of "causative". In studying how entities interact after the exertion of an external force, Talmy identifies what he calls "finer primitives" that can recombine in different patterns to produce a specific system behavior. The novelty of his work was to propose that what had been viewed as "an irreducible concept" (the "cause-effect implication" relationship) could be seen "as a complex build up of primitive concepts" [Talmy, 2000]. Borrowing some of these concepts and extending their application beyond the language and cognition context for the purpose of better detailing accident causation, interactions between Agonist and Antagonist actions and interactions between Agonist and Inverse Agonist actions are analyzed next and related to an accident sequence evolution.

The interactions of Agonist, Antagonist, and Inverse Agonist actions lead to the identification and the articulation of what is referred to next as *primitives of causality* (PoC). The implications of the introduction of the primitives of causality for accident prevention will be evident afterwards. These interactions involve both static and dynamic considerations: Agonist, Antagonist, and Inverse Agonist action will either be present in the system, or added/removed by external agents, as presented next.

Interactions between Agonist and Antagonist Actions

Talmy's work on force dynamics in language and cognition is based on the assumptions that "underlying all more complex force-dynamic patterns is the steady-state opposition of two forces", namely the Agonist and the Antagonist actions. This

approach is here extended to consider the possibility of presence, absence, and/or removal of Agonist and Antagonist forces. All the possible combinations of this interaction can be represented in a matrix form, as in Figure C.5, with each axis corresponding to one type of action. Agonist actions are located on the x-axis, and Antagonist actions on the y-axis. A value of 0 corresponds to absence; a value of 1 to presence. The possibility of removal of the Antagonist (defensive) action by an external agent corresponds to a value of -1.

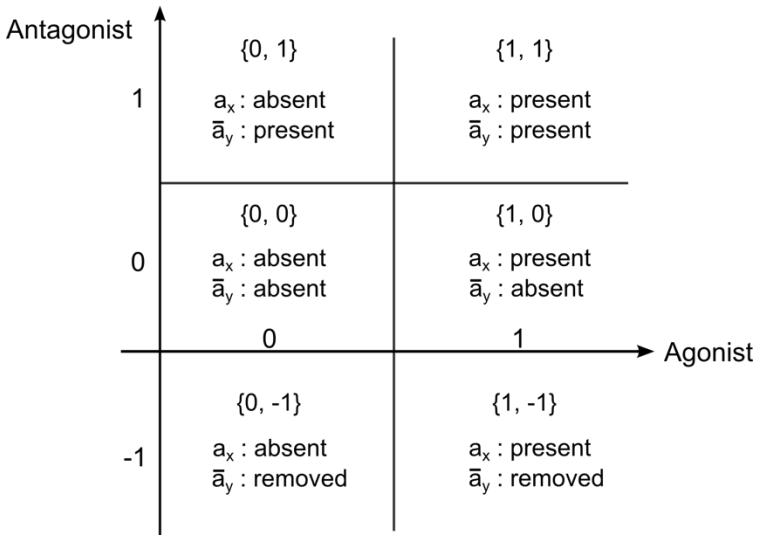

Figure C.5 Agonist and Antagonist Interactions in matrix form

Each combination of {x, y} coordinates represents a different primitive of causality:

- *Direct Causation: Coordinates {1, 0}:* this primitive originates from an unimpeded Agonist action pushing the system to a more hazardous state. The causal relationship between the cause "Agonist action presence" and the effect "hazard escalation" is defined as direct causation primitive of causality (Figure C.6)

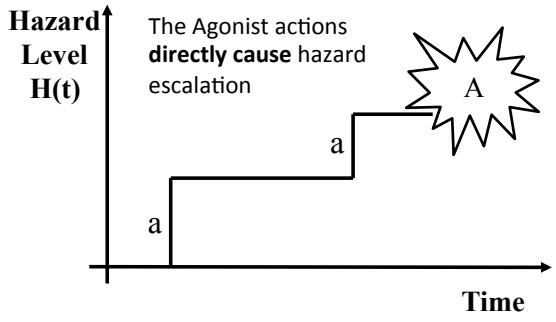

Figure C.6 *Direct Causation* primitive of causality

The direct causation primitive is what is traditionally understood as causality given the direct cause-effect implication stemming from the absence of any defensive resource.

- *Blocking: Coordinates {1, 1⁺}:* this primitive originates from the presence of both an agonist and an antagonist action on the system, with an Antagonist action stronger than the Agonist action[19]. The causal relationship between the cause "Agonist and antagonist action presence" and the effect "blocked hazard escalation" is defined as the blocking primitive of causality (Figure C.7).

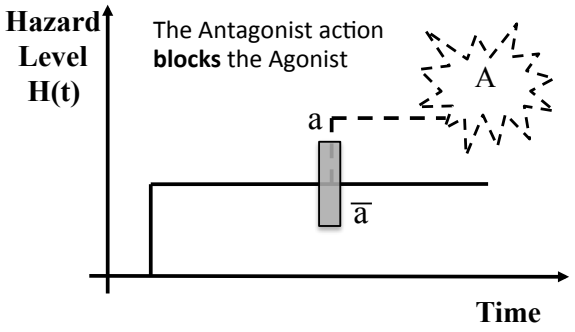

Figure C.7 *Blocking* primitive of causality

[19] The fact that the Antagonist is able to overcome the Agonist action is represented in the coordinate definition of the Antagonist as 1^{+}.

- *Despite: Coordinates {1, 1⁻}:* this primitive originates from the presence of both an Agonist and an Antagonist action on the system, with an Agonist action stronger than the Antagonist action[20]. The causal relationship between the cause "Agonist and Antagonist action presence" and the effect "unblocked hazard escalation" is defined as the despite primitive of causality (Figure C.8).

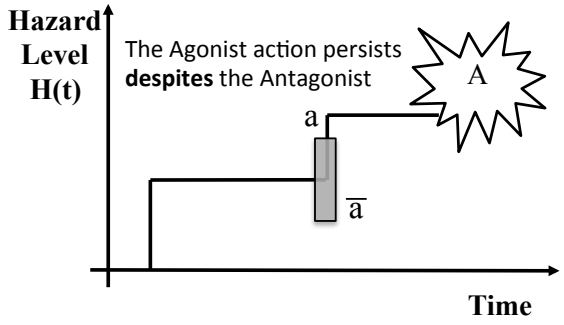

Figure C.8 *Despite* primitive of causality

- *Prevention: Coordinates {0, 1}:* this primitive originates from the presence of an Antagonist action with no occurrence of an Agonist action. The effect of a stationary persistence of the system in its original condition defines the prevention primitive of causality (Figure C.9).

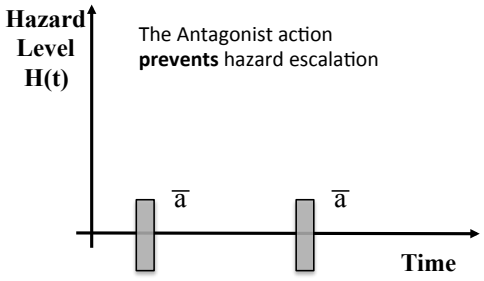

Figure C.9 *Prevention* primitive of causality

[20] The fact that the Antagonist is overcome by the stronger Agonist action is represented in the coordinate definition of the Antagonist as 1^{-}.

The primitives of causality introduced so far were characterized by a static nature: the antagonist action was either present or absent, and no changes were allowed on the system. Next it is possible to consider cases where dynamic considerations come into play by means of external agents acting on the system configuration.

- *Fragilizing: Coordinates {0, -1}*: this primitive originates from the removal of an Antagonist action with no occurrence of an Agonist action. The causal relationship between the cause "removed Antagonist action" and the effect "unblocked hazard escalation" with the system persisting in its original condition defines the fragilizing primitive of causality (Figure C.10).

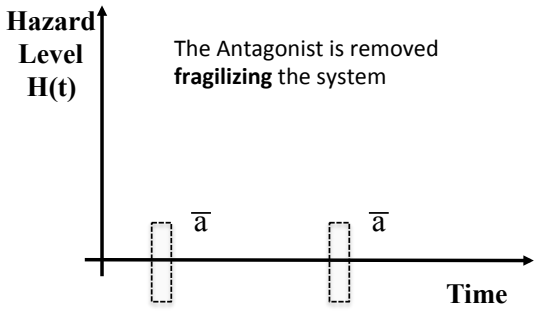

Figure C.10 *Fragilizing* primitive of causality

- *Letting: Coordinates {1, -1}:* this primitive originates from the presence of an Agonist action and the removal of an Antagonist action. The causal relationship between the cause "Agonist presence and removal of Antagonist action" and the effect "unblocked hazard escalation" defines the letting primitive of causality (Figure C.11).

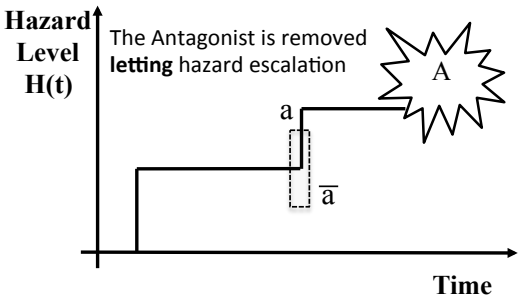

Figure C.11 *Letting* primitive of causality

The six primitives of causality identified so far can be summarized in the matrix form, as can be seen in Figure C.12. The {0, 0} coordinates indicate a "steady" condition, as this situation implies that neither Agonist nor Antagonist actions are present, and hence there are no dynamics occurring at the system level.

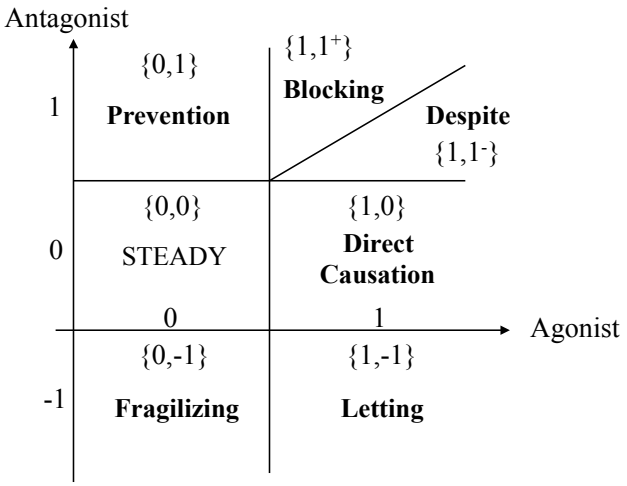

Figure C.12 Primitives of causality – Agonist and Antagonist interactions

Interactions between Agonist and Inverse Agonist Actions

The notion of Inverse Agonist is not present in Talmy's work. As mentioned previously, this concept was borrowed and extended from the biochemistry framework. By definition of Inverse Agonist, this category of actions requires the occurrence of a previous Agonist action, as hazard de-escalation only follows a system in a hazardous

state. Primitives of causality in this case are thus restricted to the case of presence of the Agonist.

As in the previous case, the primitives of causality are summarized in a matrix form, this time considering Inverse Agonist actions on the y-axis.

Note that even if the Inverse Agonist differs both in nature and in effect on the hazard level from the Antagonist action, they share the primitives of "direct causation", "despite" and "letting" given their defensive nature. However, the blocking primitive is now replaced by:

- *De-escalation: Coordinates {1, 1^{+}}:* this primitive originates from the presence of both an Agonist and an Inverse Agonist action on the system, with the Inverse Agonist action stronger than the Agonist force. The causal relationship between the cause "Agonist and Inverse Agonist action presence" and the effect "hazard de-escalation" is defined as de-escalation primitive of causality (Figure C.13).

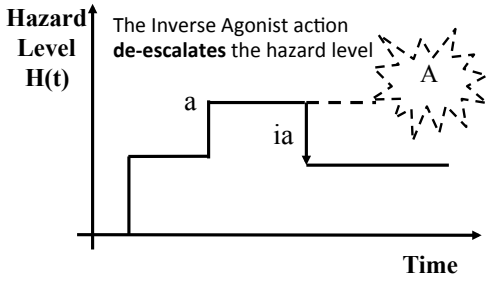

Figure C.13 *De-escalation* primitive of causality

The primitives of causality derived from the interactions between Agonist and Inverse Agonist are summarized in Figure C.14.

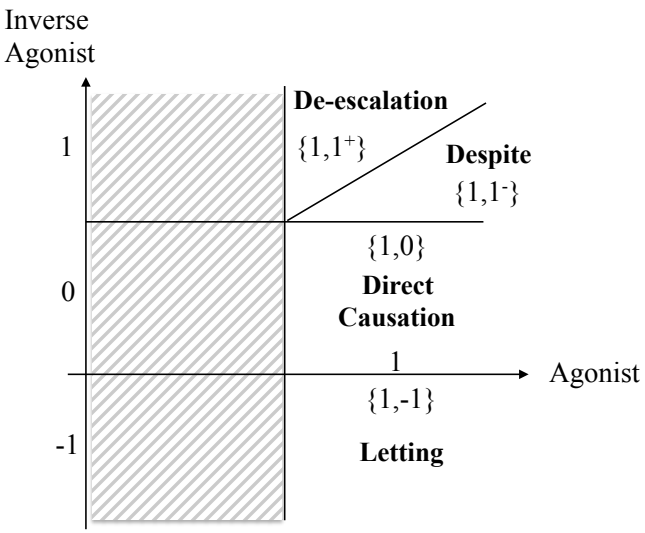

Figure C.14 Primitives of causality – Agonist and Inverse Agonist interactions

The one-to-one interactions analyzed so far can be combined together to generate the complex web of causality that characterizes real accidents. The introduction of the primitives of causality allows a finer description of the micro-causal transitions between different states, as if each transition (or part of the accident sequence) was analyzed under a microscope. The plurality and co-existence of primitives of causality then shapes the accident sequence by contributing to its escalation or its blocking, its prevention, etc.

APPENDIX D

OPERATIONAL GUIDELINES AND NOTES ON THE FRAMEWORK APPLICABILITY

The framework and analytical tools presented in this work leveraged the two ingredients of model-based hazard monitoring and of Temporal Logic. As noted in Chapter 5, other than for the specification and modeling of digital components (which is the prime use of TL in the software engineering community), TL was employed for the definition of more complex expressions of *H(t)*, which included temporal ordering inside the analytical definition of the danger indices of interest. This use, as well as the specification of safety property to constraint the system behavior, can also be achieved by leveraging the state-space representation formalisms alone, at the expenses of more complex expressions for *H(t)* (for instance, involving derivatives with respect to time in place of the temporal operators), and of bigger state-space models, possibly facing the problem of state explosion.

The adoption of TL thus allowed to reduce the number of states to model in detail for the system of interest and allowed to re-introduce the dynamical behavior of specific states without recurring to additional state equations. Important questions arise then in relation to the previous considerations, such as: Which states are best modeled using the state-space representation? When should TL be considered within the definition of *H(t)* itself?

The answers to these important questions are, in a sense, case dependent, and rely on the experience and the ingenuity of the analyst of the system. Nevertheless, there are high-level guidelines that can be devised and can provide important suggestions whenever an external user wishes to implement (or adopt part of) the framework presented in this work.

The process of adopting the proposed framework for the analysis of a novel system needs careful attention and research on the part of the user and can be summarized by looking at four important steps, which are described next.

1) ***Definition of the scope of the analysis and the accident of interest*:** The first step for the adoption of the proposed approach is the careful understanding of the scope of the analysis. For instance, the approach adopted in Chapter 5 was, in a sense, that of accident reconstruction for the purpose of investigating contributory causes, and of understanding potential flaws in the system design. Two main types of analysis can be devised: *ex-post analysis* (after the system has been designed and adverse events have occurred), and *ex-ante analysis* (before adverse events have unfolded, towards validation of the system design). In both cases there will be specific safety requirements that the system has to meet (for instance, dictated by regulations), or that the analyst wishes to verify to test his/her hypothesis of a potential failure. This in turn translates into the definition of the specific safety properties of interest to be used as constraints on the system behavior. Each constraint works towards the devising of a "safety envelope" or boundary of safe operations. At this stage, it is not necessary to express the constraint or the safety envelope in state-space or in Temporal Logic, and qualitative definitions are acceptable. In fact this first step is a high-level drafting of the following step 3, which will account for the detailed modeling. This first step is mostly aimed at better understanding which conditions will define an accident, or, in other words, the conditions that define when the system is breaching the safety envelope of operations. The understanding of the conditions that define an accident (for instance, the "loss of containment through tower overflow" for the oil tank example of Chapter 3) are fundamental for choosing the state variables to pick for the system analysis and

for the definition of the hazard levels of interest (for instance, the choice of translating the accident "loss of containment" into the requirement of monitoring the height of oil inside the tower).

On a practical level, it is good to start by looking at regulatory constraints for the system, and at similar systems that have in the past experienced mishaps and adverse events, to better understand the accident(s) of interest. Moreover, any risk assessment approach starts with a "hazard identification" effort, aimed at exploring the potential problems of a system (for instance through HAZOP tools, as indicated in Chapter 2). This information can be a good starting point also for the application of the proposed framework.

2) ***Development of the dynamical model*:** Once it is clear which accidents the user whishes to monitor/do prevention against, it is possible to start the actual modeling of the system under consideration. That is, based on the rough idea of the accident and constraint of interest, the user has to develop a dynamical model. The framework was in this work implemented in Simulink, as this environment provides multiple modeling tools to combine continuous and digital components. One of the most important choices is that of defining the level of detail necessary for the analysis, and understand for which parts of the system to develop the detailed state-space representation. Once more, this choice is dictated by the scope of the analysis of step 1, but it is also constrained by the information available on the system. For instance, it is intuitive that the modeling of digital components can occur in state-flow, with state charts and guarding functions to command the transitions between states more or less detailed depending on the information that is available to the user on the actual software codes. For continuous systems, which the user wishes to model with state-space, other considerations apply. For instance, a user could leverage

traditional tools of PRA such as fault trees (whenever they have been developed for the system of interest) to discover which functions/components contribute the most to the computation of the probability associated to a given failure event, and model in detail only those. In other cases the choice is informed and/or dictated by the accident of interest. For instance, in the accident sequence of Chapter 5, a much more refined model that considered the lateral movement of the plane and the states associated to the various control surfaces could have been developed. However, for that specific case the NTSB report shows that there wasn't a significant lateral movement of the aircraft, making a more refined model a huge expense in term of time, with little contribution for the analysis of interest. There is thus a *cost-benefit analysis* in relation to the difficulty of having more complex and detailed model and to the benefit that such refined models can bring to the analysis.

In general, it is a safe bet and good practice to start out with a simpler model that only takes into account a handful of states that are necessary for the monitoring of the accident(s) of interest defined in step 1. Only later, should more refined results be needed, modifications and additional information can be employed to refine the system model.

3) ***Detailed definition of the TL constraints and of the hazard levels of interest***: Once the model for the dynamical system has been developed, the user needs to devise and model the metrics of interest (in terms of danger indices or hazard levels) and express in a *quantifiable way* the constraints associated with the safety envelope for the system under consideration. Their *qualitative definition* is based on step 1, and their quantification requires a long process of trial-and-error, to make sure that the metrics picked for the analysis can capture the behavior that the user originally intended. As mentions in the concluding

chapters, some constraints (such as the four ones considered for the case of Chapter 5) are easily adaptable to different system and can be considered as a starting point for the analysis. There is not a strict rule that tells whether it is more convenient to start with the definition of the danger indices, or the definition of the constraints. In general, they will inform each other, as not all metrics are relevant for the selected constraints and vice versa. There are thus multiple iterations that occur within this step. A first issue to discuss is the understanding of whether TL is necessary for the expression of both constraints and *H(t)* definitions. As a rule of thumb, whenever temporal ordering is an issue and the user whishes to analyze the effects of *H(t)* dynamics (e.g. hazard escalation, or de-escalation) at different times, TL is a good candidate. This is so because the use of state-space alone may result in too complex model, resulting in a poor balance of that cost-benefit trade-off previously mentioned. Regarding the actual formulation of the metrics of interest, ingenuity and a good knowledge of the system are necessary ingredients.

In general, it is a good practice to start by looking at possible thresholds for the states of interest (i.e., understand whether there is a minimum or a maximum associated with safe or nominal system operations). For instance, in the case of the oil tank example, a maximum height was easily identifiable; or in the case of an aircraft, there are bounds of acceptable values of the angle of attack to prevent stall, or on the maximum stresses to be withstood by the fuselage. Whenever conditions similar to the previous ones can be defined, hazard levels and metrics can be devised as ratios between the actual state of the system and the maximum acceptable value (and/or range). When simple bounds cannot be defined, the second option adopted in this work was the construction of ratios between two quantities that are state-dependent. This was the case of the metric analyzed in relation to RTO initiation, where both the stopping

distance (the numerator) and the available distance to stop (the denominator) depended on the states of the system. Finally, a third option adopted in this work was that of combining simple ratios (such as the normalized position along the runway) with logical variables (i.e., true - 1 or false - 0 evaluations) that represent whether specific conditions are met by the system. This was the case of the fail-safe analysis carried out in conjunction with the identification of spurious AIR mode indications. Future applications are likely to provide additional operational guidelines towards this step.

4) ***Simulation of the system and post-processing analysis***: With the development of the dynamical system and of the associated indices completed, the user can now run the simulation and analyze the results. These results are provided in terms of a report of violation/compliance with the specified constraints, and in terms of plots of *H(t)* in time. As indicated in Chapter 6, simple control panels were developed for the thesis, and there are interesting implications and research opportunities in terms of human factor and human-machine interfaces that can aide the development of more complex visual aides, especially in relation to their on-line use. This is an important capability of the proposed approach, given the not so intuitive and easily understood results that traditional DPRA tools provide to the user/analyst, as discussed in Chapter 2, 6, and Appendix B.

The process previously outlined is not necessarily carried out in a linear fashion. Some steps can inform the definition of other ones, and some iterations are needed in most cases to achieved a good level of detail (satisfactory for the specific goal the user intended for the analysis). The considerations contained in the previous steps can furthermore be combined with some of the operational guidelines contained in

Appendix A, and in Figure A.2, which is re-proposed here for convenience of the reader.

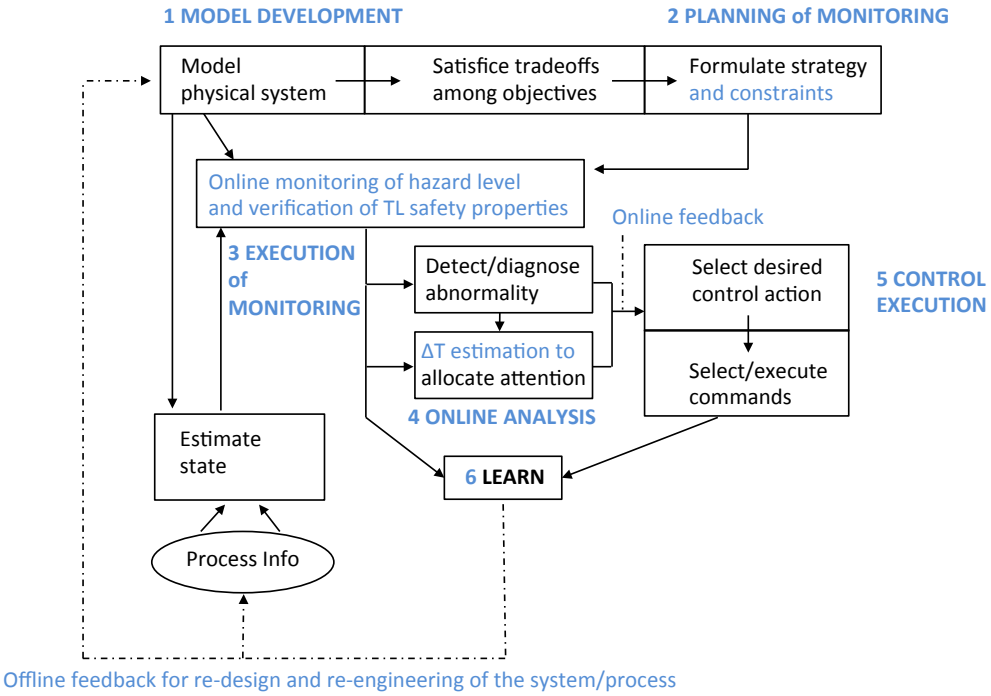

Figure D.4 Operational steps for the application of the proposed framework

The thesis provided a case study that served as "proof-of-concept" of the theoretical development carried out in chapters 1-4. It is hoped that its detailed development in chapter 5 together with the guidelines provided here can serve as an actionable cue to invite further contributions in the same spirit that inspired the present work.

REFRENCES

Airbus (2005). *Flight Operations Briefing Notes* – Supp. Tech., Rev. 2 – May 2005. http://www.airbus.com/fileadmin/media_gallery/files/safety_library_items/AirbusSafetyLib_-FLT_OPS-SUPP_TECH-SEQ02.pdf, (accessed on 01/21/2016).

ARP4754, S.A.E., 1996. *Certification considerations for highly-integrated or complex aircraft systems*. SAE, Warrendale*, PA*.

Aldemir, T., Guarro, S., Mandelli, D., Kirschenbaum, J., Mangan, L. A., Bucci, P., ... & Arndt, S. A. (2010). *Probabilistic risk assessment modeling of digital instrumentation and control systems using two dynamic methodologies.* Reliability Engineering & System Safety, 95(10), 1011-1039.

Aldemir, T. (2013). *A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants*. Annals of Nuclear Energy, 52, 113-124.

Apostolakis, G. E. (2004). *How useful is quantitative risk assessment?* Risk Analysis, 24(3), 515-520.

Bakolas, E., and Saleh, J. H. (2011). *Augmenting defense-in-depth with the concepts of observability and diagnosability from control theory and discrete event systems*. Reliability Engineering & System Safety, 96(1), 184-193.

Baier, C., & Katoen, J. P. (2008). *Principles of model checking* (Vol. 26202649). Cambridge: MIT press

Bellman, R. (1957). *Dynamic programming and Lagrange multipliers*. Proceedings of the National Academy of Sciences of the United States of America, 42(10), 767.

Bozzano, M., and Villafiorita, A. (2003). *Improving system reliability via model checking: The FSAP/NuSMV-SA safety analysis platform*. In: Computer Safety, Reliability, and Security (pp. 49-62). Springer Berlin Heidelberg.

Bozzano, M., Villafiorita, A., Åkerlund, O., Bieber, P., Bougnol, C., Böde, E., ... and Zacco, G. (2003). *ESACS: an integrated methodology for design and safety analysis of complex systems*. In Proceedings of ESREL 2003 Conference, Maastricht, The Netherlands, 15-18 June 2003.

Chen, C. T. (1995). *Linear system theory and design*. Oxford University Press, Oxford, UK.

Cowlagi, R. V., and Saleh, J. H. (2013). *Coordinability and consistency in accident causation and prevention: formal system theoretic concepts for safety in multilevel systems*. Risk analysis, 33(3), 420-433.

DOD – Department of Defense Standard Practice System Safety. (2012). *MIL – STD – 882E.* Available at http://www.system-safety.org/Documents/MIL-STD-882E.pdf , (accessed on 06/17/2015).

ECAST – European Commercial Aviation Safety Team (2016). *Runway Excursion Preventions*. Available at http://easa.europa.eu/essi/ecast/main-page-2/runway-safety/ , accessed on 01/21/2016.

Embrey, D. and Zaed, S., (2010). *"A Set of Computer Based Tools Identifying and Preventing Human Error in Plant Operations"*. Available at www.humanreliability.com [accessed on 06/07/2012].

Fainekos, G. E., Girard, A., Kress-Gazit, H., & Pappas, G. J. (2009). *Temporal logic motion planning for dynamic robots*. Automatica, *45*(2), 343-352.

Favarò, F. M., Jackson, D. W., Saleh, J. H., & Mavris, D. N. (2013). *Software contributions to aircraft adverse events: Case studies and analyses of recurrent accident patterns and failure mechanisms*. Reliability Engineering & System Safety, 113, 131-142.

Favarò, F. M., and Saleh, J. H. (2013). *Observability in Depth: novel safety strategy to complement defense-in-depth for dynamic real-time allocation of defensive resources*. In Proceeding of ESREL 2013 Conference, Amsterdam, The Netherlands, 29 September – 2 October 2013.

Favarò, F. M., and Saleh, J. H. (2014). *Observability-in-depth: an essential complement to the defense-in-depth safety strategy in the nuclear industry*. Nuclear Engineering and Technology, 46(6), 803-816.

Favarò, F. M., and Saleh, J. H. (2016a). *Toward Risk Analysis 2.0: Safety Supervisory Control and Model-based Hazard Monitoring for Dynamic Risk-informed Safety Interventions*. Submitted to Reliability Engineering and System Safety, January 2016.

Favarò, F. M., and Saleh, J. H. (2016b). *Application of Temporal Logic for Safety Supervisory Control and Model-based Hazard Monitoring*. Submitted to Reliability Engineering and System Safety, March 2016.

Favarò, F. M., and Saleh, J. H. (2016c). *Temporal Logic for System Safety Properties and Hazard Monitoring*. Submitted to the Journal of Loss Prevention in the Process Industries, January 2016.

Federal Aviation Administration (FAA). (2005). Advisory Circular AC No: 150/5325-4B, *Runway Length Requirements for Airport Design*. AAS-100, July 2005

Ferrell, W. R., and Sheridan, T. B. (1967). *Supervisory control of remote manipulation*. Spectrum, IEEE, 4(10), pp.81-88.

Ferris, T., Sarter, N., and Wickens, C. D. (2010). *Cockput Automation: still struggling to catch up.* In Salas, E., Jentsch, F. and Maurino, D. Human factors in aviation, Academic Press, San Diego, CA.

Fisher, M. (2011). An Introduction to Practical Formal Methods Using Temporal Logic. John Wiley & Sons.

Foreman, V.L., Favarò, F.M., Saleh, J.H. and Johnson, C.W. (2015). *Software in military aviation and drone mishaps: Analysis and recommendations for the investigation process*. Reliability Engineering & System Safety, 137, pp.101-111.

Galton, A. (1987). *Temporal logic and computer science: An overview*. In: Temporal logics and their applications, pp. 1-52. Academic Press Professional, Inc..

Gnoni, M.G. and Lettera, G., (2012). *Near-miss management systems: A methodological comparison.* Journal of Loss Prevention in the Process Industries, 25(3), pp.609-616.

Gnoni, M.G., Andriulo, S., Maggio, G. and Nardone, P., (2013). *"Lean occupational" safety: An application for a Near-miss Management System design.* Safety science, 53, pp.96-104.

Gong, C. and Chan, W.N., 2002, October. Using flight manual data to derive aero-propulsive models for predicting aircraft trajectories. In *Proc. of AIAA's Aircraft Technology, Integration, and Operations (ATIO) Forum, Los Angeles, CA*.

Haimes, Y. Y. (2009). *On the Complex Definition of Risk: A Systems Based Approach*. Risk analysis, 29(12), 1647-1654.

Hansen, K. M., Ravn, A. P., & Stavridou, V. (1998). *From safety analysis to software requirements*. Software Engineering, IEEE Transactions on, 24(7), 573-584.

Hollnagel, E. (2004). *Barriers and accident prevention*. Aldershot: Ashgate, Hampshire, UK.

Hopkins, A. (2001). *Was Three Mile Island a 'Normal Accident'?*. Journal of contingencies and crisis management, 9(2), pp.65-72.

Ikuta, K., Ishii, H. and Nokata, M. (2003). *Safety evaluation method of design and control for human-care robots*. The International Journal of Robotics Research, 22(5), 281-297.

Isermann, R. (2005). *Model-based fault-detection and diagnosis–status and applications*. Annual Reviews in control, 29(1), 71-85.

Jahanian, F., and Mok, A. K. L. (1986). *Safety analysis of timing properties in real-time systems.* Software Engineering, IEEE Transactions on, (9), 890-904.

Jahanian, F., and Mok, A. K. (1994). *Modechart: A specification language for real-time systems*. Software Engineering, IEEE Transactions on, 20(12), 933-947.

Johnson, C. W. (1995). *Decision theory and safety-critical interfaces*. In Proceeding of Interact 1995, Lillehammer, Norway, 25-29 June 1995.

Johnson, C. W. (2000). *Proving properties of accidents.* Reliability Engineering & System Safety, 67(2), 175-191.

Johnson, C. W., & Harrison, M. D. (1992). *Using temporal logic to support the specification and prototyping of interactive control systems*. International Journal of Man-Machine Studies, 37(3), 357-385.

Joshi, A. and Heimdahl, M.P. (2005). *Model-based safety analysis of Simulink models using SCADE design verifier*. In Computer Safety, Reliability, and Security (pp. 122-135). Springer Berlin Heidelberg.

Kalman, R.E (1960). *Contributions to the theory of optimal control*. Boletin Sociedad Matematica Mexicana, Vol. 5, 102–119.

Kaplan, S., & Garrick, B. J. (1981). *On the quantitative definition of risk*. Risk analysis, 1(1), 11-27.

Kirschenbaum, J., Bucci, P., Stovsky, M., Mandelli, D., Aldemir, T., Yau, M., Guarro, S., Ekici, E. & Arndt, S. A. (2009). *A benchmark system for comparing reliability modeling approaches for digital instrumentation and control systems*. Nuclear Technology, 165(1), 53-95.

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.

Kulić, D. and Croft, E.A. (2005). *Safe planning for human–robot interaction*. Journal of Robotic Systems, 22(7), 383-396.

Leveson, N. (1995). *Safeware: System Safety and Computers, Sphigs Software*. Addison-Wesley Professional.

Leveson, N. (2004). *A new accident model for engineering safer systems*. Safety science, 42(4), 237-270.

Magott, J., & Skrobanek, P. (2012). *Timing analysis of safety properties using fault trees with time dependencies and timed state-charts*. Reliability Engineering & System Safety, 97(1), 14-26.

Manna, Z., Pnueli, A. (1992). *Temporal logic of reactive and concurrent systems: specifications* (Vol. 1). Springer

Moray, N. (1986). *Monitoring Behavior and Supervisory Control*. In Handbook of Perception and Human Performance Vol. 2, K. Boff, L. Kaufman, J. P. Thomas, John Wiley & Sons, New York, NY.

Mosleh, A. (2014). *PRA: A Perspective on Strengths, Current Limitations, And Possible Improvements*. Nuclear Engineering and Technology, (1), 1-10.

Mosleh, A., Bier, V.M. and Apostolakis, G. (1988). *A critique of current practice for the use of expert opinions in probabilistic risk assessment*. Reliability Engineering & System Safety, 20(1), pp.63-85.

NASA - Vesely, W., Stamatelatos, M., Dugan, J., Fragola, J., Minarick III, J. and Railsback, J. (2002). *Fault tree handbook with aerospace applications*. Office of safety and mission assurance NASA headquarters, 2002.

National Transportation Safety Board (NTSB). (2010). *Runway Overrun During Rejected Takeoff, Global Exec Aviation, Bombardier Learjet 60, N999LJ, Columbia, South Carolina, September 19, 2008*. Aircraft Accident Report NTSB/AAR-10/02. Washington, DC.

NRC, US (2000). *Causes and Significance of Design Basis Issues at US Nuclear Power Plants*. Draft Report, Washington, DC: US Nuclear Regulatory Commission, Office of Nuclear Regulatory Research.

Palshikar, G. K. (2002). *Temporal fault trees*. Information and Software Technology, 44(3), 137-150.

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.

Perrow, Charles (1984). *Normal Accidents: Living with High-risk Technologies*. New York: Basic Books. Print.

Rasmussen, N. C. (1975). *Reactor Safety Study: An assessment of accident risks*. In US. Commercial Nuclear Power Plants, Nuclear Regulatory commission – NUREG 1975.

Rasmussen, J. (1997). *Risk management in a dynamic society: a modelling problem*. Safety science, 27(2), 183-213.

Reason, J. T., (1997). *Managing the risks of organizational accidents* (Vol. 6). Aldershot: Ashgate, Hampshire, UK.

Rescher, N., & Urquhart, A. (1971*). Temporal Logic*, Vol. 3 of Library of Exact Philosophy. Springer–Verlag, Heidelberg, Germany, 42, 140.

Roberts, K. H. (1990a). *Managing high reliability organizations*. California Management Review, 32(4).

Roberts, K. H. (1990b). *Some characteristics of one type of high reliability organization*. Organization Science, 1(2), 160-176.

Saleh, J. H., Marais, K. B., Bakolas, E., & Cowlagi, R. V. (2010). *Highlights from the literature on accident causation and system safety: Review of major ideas, recent contributions, and challenges*. Reliability Engineering & System Safety,95(11), 1105-1116.

Saleh, J. H., Haga, R. A., Favarò, F. M., & Bakolas, E. (2014a). *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.

Saleh, J. H., Marais, K. B., & Favaró, F. M. (2014b). *System safety principles: A multidisciplinary engineering perspective*. Journal of Loss Prevention in the Process Industries, 29, 283-294.

Sheridan, T.B. (1960). *Human metacontrol*. In Proceedings of the Annual Conference on Manual Control, Wright Patterson Air Force Base, OH.

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control*. MIT press, Cambridge, MA.

Sheridan, T.B., (2012). *Human supervisory control*. In Salvendy, G., Handbook of Human Factors and Ergonomics, Fourth Edition, John Wiley & Sons, New York, NY.

Singer, E., and Endreny, P. M. (1993). *Reporting on Risk: How the Mass Media Portray Accidents, Diseases, Disasters, and Other Hazards*. Russell Sage Foundation RSF Editor, New York, NY.

Sørensen, J. N., Apostolakis, G. E., Kress, T. S., and Powers, D. A. (1999). *On the Role of Defense in Depth in Risk-Informed Regulation*. In: Proceedings of the PSA '99. International topical meeting on probabilistic safety assessment, Washington, DC, August 22–26, 1999, American Nuclear Society, La Grange Park, Illinois. p. 408–413.

Sørensen, J. N., Apostolakis, G. E., and Powers, D. A. (2000). *On the role of safety culture in risk-informed regulation*. In: Kondo S, Furuta K, editors. Psam 5: Probabilistic Safety Assessment and Management, Volumes 1–4, pp. 2205–2210.

Svedung, I., & Rasmussen, J. (2002). *Graphic representation of accident scenarios: mapping system structure and the causation of accidents*. Safety Science, 40(5), 397-417.

Swain, A. D. (1990). *Human reliability analysis: Need, status, trends and limitations.* Reliability Engineering & System Safety, 29(3), 301-313.

Talmy, L. (2000). *Toward a cognitive semantics*, *(Vol. 1)*. MIT press, MA

TSTA – Takeoff Safety Training Aid (2016). *Pilot Guide to Takeoff Safety*. Available at https://www.faa.gov/other_visit/aviation_industry/airline_operators/training/media/takeoff_safety.pdf , (accessed on 01/21/2016).

Turner, B. A. (1978). *Man-made Disasters*. Wykeham Publications, London, UK.

US Nuclear Regulatory Commission [NUREG]. (1975). *Reactor safety study: An assessment of accident risks in US commercial nuclear power plants*. WASH-1400, NUREG-75/014. Washington, DC.

US Nuclear Regulatory Commission [NUREG]. (1995). *High integrity software for nuclear power plants*. NUREG/CR-6263 Volume 1. Washington, DC.

US Nuclear Regulatory Commission [NUREG]. (1996). *Review guidelines on software languages for use in nuclear power plant safety systems*. NUREG/CR-6463. Washington, DC.

Zhang, J., & Cheng, B. H. (2006). Using temporal logic to specify adaptive program semantics. *Journal of Systems and Software*, *79*(10), 1361-1369.

Zio, E. (2014). *Integrated deterministic and probabilistic safety assessment: Concepts, challenges, research directions*. Nuclear Engineering and Design, 280, 413-419