Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P5799-cd

From Classical to Advanced Risk Methods: Demonstrator for Industrial Cyber-Physical Systems

Andrey Morozov[1], Tagir Fabarisov[1], Silvia Vock[2], Thorben Schey[1], Artur Karimov[1], Georg Siedel[2], Joachim Grimstad[1], Arne Sonnenburg[2], Thomas Mössner[2]

[1] *University of Stuttgart, Institute of Industrial Automation and Software Engineering (IAS), Germany. E-mail: {first.last}@ias.uni-stuttgart.de*
[2] *Federal Institute for Occupational Safety and Health (Bundesanstalt für Arbeitsschutz und Arbeitsmedizin, BAuA), Germany. E-mail: {last.first}@baua.bund.de*

Modern industrial Cyber-Physical Systems (CPS) exhibit high levels of reconfigurability and heterogeneity, posing significant challenges for risk assessment in dynamic environments. Traditional risk assessment methods, originally developed for simpler systems, often fall short when dealing with the complexity of modern CPS. This paper introduces a hardware/software demonstrator designed to simulate flexible production lines, showcasing how variations in system configuration impact the balance between production costs, reliability, and safety. The demonstrator dynamically evaluates a selected system configuration using a set of risk assessment methods, including Fault Trees, Stochastic Petri Nets, and Dual Graph Error Propagation models using Probabilistic Model Checking. The demonstrator incorporates three production tasks of increasing complexity and a risk assessment method each, highlighting the strengths and limitations of each approach. Based on these findings, we propose enhancements to existing risk models. We advocate for a hybrid approach that integrates traditional and advanced methods to meet the demands of next-generation industrial systems. Our demonstrator concept can be used to evaluate how different risk assessment methods address the challenge of reconfigurability in modern industrial CPS.

Keywords: Risk Assessment, Industrial Cyber-Physical Systems, Fault Tree Analysis, Stochastic Petri Nets, Probabilistic Model Checking, Dual-Graph Error Propagation Model, Reconfigurable Systems

1. Introduction

Industrial Cyber-Physical Systems (CPS) are becoming increasingly complex, posing a challenge to their thorough risk assessment. Recent literature surveys Bolbot et al. (2019); Leimeister and Kolios (2018); Häring and Häring (2021); Kabir and Papadopoulos (2019); Villani et al. (2018); Huck et al. (2021); Giallanza et al. (2024); Zacharaki et al. (2020) reveal that existing risk methods often struggle with the challenges posed by complex systems with dynamic reconfigurations. Such methods are particularly limited in addressing error propagation within modular, software-defined manufacturing environments.

This paper presents a physical demonstrator that consists of small hardware units, each representing a specific machine in a production line for the manufacturing of printed circuit boards (PCBs). The demonstrator software enables the display of the manufacturing setup and the simulation of failure scenarios. It models several types of machines and supports to easily calculate and display what risk factors the reconfiguration of the given production lines poses. Users can intuitively understand the trade-offs between system safety and operational costs.

The demonstrator reflects major aspects of industrial CPS, including: (i) heterogeneity, (ii) structural and behavioral complexity, (iii) reconfigurability, and (iv) complex failure scenarios with dynamic dependencies between hazardous events and failure modes. The demonstrators allows to show the challenges that these aspects pose for such a system with respect to risk methods.

We demonstrate these challenges through a comparative analysis of Fault Tree Analysis (FTA) Vesely et al. (1981), Stochastic Petri Nets (PN) Marsan (1990); Molloy (1982), and Probabilistic Model Checking (PMC) Baier and Katoen (2008) implemented using Dual-graph Error Propagation

Table 1.: Comparison on selected risk assessment methods.

Aspect	**FTA** (static)	**Stochastic PN** (dynamic)	**PMC** (advanced)
Heterogeneity	+	++	+++
Complexity (structural, behavioral, distributed)	+	++	+++
Flexibility (reconfigurability, repurposability)	++	++	+++
Complex failure scenarios	-	+	+++

Models (DEPM) Morozov and Janschek (2011); Morozov et al. (2019) on our demonstrator system. Through this method selection we cover classical static probabilistic risk methods (FTA), dynamic Markov-based methods capable of handling parallel processes (SPN), and advanced, flexible approaches leveraging PMC with DEPM. The demonstrator underscores the limitations of traditional methods in capturing dynamic risks and highlights the advantages of advanced techniques in addressing complex error chains.

Table 1 provides a comparison of the selected risk assessment methods according to the discussed aspects. By showcasing the limitations and strengths of each approach, the demonstrator serves as both an educational tool and a research platform.

Contributions: The key contributions of this paper are: (i) the development of a modular hardware/software demonstrator to explore risk assessment under dynamic CPS configurations; (ii) comparative analysis of FTA, SPN, and PMC methods on the same system; and (iii) demonstration of the strengths and limitations of traditional and advanced methods.

2. Demonstrator architecture

The demonstrator is a modular system that simulates a reconfigurable production line. The setup of the demonstrator is shown in Figure 1. It consists of the following components:

Production Line Units (PLUs): Each PLU is a standalone hardware/software unit that represents a specific type of machine in the production line. The PLUs communicate with the Central Controller.

Central Controller: A standard Linux personal computer functions as the Central Controller, running software that simulates the production process, generates risk models, and performs risk and cost calculations. It also manages connections with each PLU using a base station dongle and an overhead camera. The Central Controller comprises four main modules: (i) Machine Vision – Interprets the PLU configuration captured with the overhead camera; (ii) Graphical User Interface (GUI); (iii) Network Module – Handles wireless communication with PLUs; (iv) Risk Assessment Module – Computes reliability/risk models based on the PLU configuration.

Overhead Camera: The overhead camera captures the spatial configuration of the PLUs. Users can adjust this configuration before starting the simulation by arranging the PLUs to model a production line.

Fig. 1.: The demonstrator setup: A PC as Central Controller, an Overhead Camera, and devices representing production line units (PLU). Users adjust the production line by spatially varying the positions of the PLUs. The Overhead Camera identifies this layout and the Central Controller generates analytical models for reliability, risk, and configuration costs of the production line.

2.1. *PLU hardware design*

Each PLU features a compact 3D-printed housing that was optimized for automated SMD manufacturing and for simple and secure mounting of all its subcomponents. Those include a 2.8-inch TFT display with a 320x240 pixel resolution that plays an animation. Power is supplied by a lithium-polymer battery with 2500 mAh capacity and managed by a dedicated IC for battery charging and monitoring, as well as a voltage regulator to deliver a stable 3.3V supply. User interaction is achieved by a rotary encoder that allows allows the selection of operational modes.

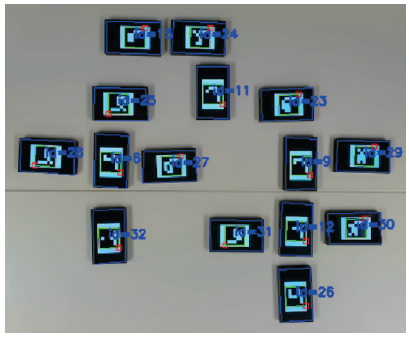
An embedded ESP32 board, directly integrated onto the PCB, serves as the core component, managing communication, display, and user input. The ESP32 was chosen for its integrated Wi-Fi and Bluetooth, dual-core processing, SRAM, and low power consumption, making it suitable for battery-operated devices. Its support for ESP-NOW enables low-latency, peer-to-peer communication, crucial for real-time system operation. Figure 2 illustrates a PLU and the base station dongle connected to the Central Controller.

Fig. 2.: A PLU and the base station dongle.

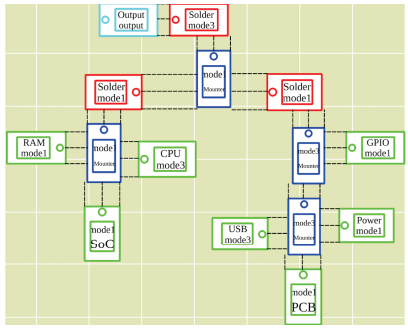
2.2. *Machine vision*

The system uses a 1080p USB camera to capture clear images of the PLU layout from above. This information is fed to the central controller to build a simulation model. Figures 3a and 3b show how a PLU configuration is interpreted.

The camera, mounted on a fixed stand, was calibrated to minimize distortion and to cover the entire PLU layout with its field of view. An image captures the spatial configuration as well as ArUco codes displayed on every PLU. It then undergoes preprocessing to enhance contrast and remove noise. Afterwards, ArUco codes are detected, which encode each PLU's type and mode



(a) PLU layout captured by the Overhead Camera.



(b) Interpreted layout of the production line.

Fig. 3.: An example production system layout modeled with PLUs and captured with the overhead camera.

of machine that it stands for in the production line. This information is used by the Central Controller for simulation and risk assessment.

2.3. *Base station and communication*

The Base Station is a USB dongle that serves as the central communication hub for the Central Controller to connect to the PLUs. Equipped with a microcontroller, the Base Station features communication protocols like ESP-NOW, enabling fast, low-latency, peer-to-peer wireless communication. In typical use, the Base Station receives a command from the Central Controller, translates it into a format compatible with the PLUs, and transmits it wirelessly. On the PLU side, the ESP32 microcontroller handles the communication, executes the command and send a confirmation back to the Base Station, which then forwards the response to the Central Controller.

While not meeting strict "real-time" timing requirements, the communication speed is sufficient for maintaining system responsiveness in the lower millisecond range. ESP-NOW operates in the 2.4 GHz band, enabling direct communication without a central router. We use unique message identifiers and timeouts to detect duplicates and handle transmission failures. Over-the-Air updates enable firmware and software updates of the PLUs without physical access, streamlining maintenance and minimizing downtime.

3. Production process simulation

The Demonstrator provides a case study for risk assessment methods. Based on the PLUs configuration, the Central Controller simulates the production process and recalculates associated risk metrics after any reconfiguration. The simulation focuses on a production line assembling and soldering PCB components such as RAM, System-on-Chip (SoC), etc. Each PLU fulfills one of four roles:

(1) **Storage**: Delivers a selected type of PCB component to mounters. It has no input and one output (to a mounter).
(2) **Mounter**: Assembles components onto PCBs. It can have up to three inputs (storages or other mounters).
(3) **Soldering Station**: Solders assembled components. It has one input (a mounter) and one output (another mounter or the Output).
(4) **Output**: The final PLU, accepting finished PCBs. It has one input (a soldering station).

Each PLU has a production cost and an operational mode affecting its failure probability. Operational states of each PLU are configured via selectable modes (low, medium, high). Users can select one of three modes via a rotary encoder: higher modes increase cost but reduce failure probability, while lower modes reduce cost but increase failure likelihood. These modes directly affect the failure parameters in the risk models. For example, in the FTA, these probabilities define basic event failure rates; in SPN, they modify transition rates; and in PMC, they influence the likelihood of fault activations. For simplicity, it is assumed that if a mounter fails, it can still pass through assembled PCBs. The user must balance these modes to achieve desired production goals.

The demonstrator is designed to evaluate risk assessment methodologies through three progressively complex tasks with different PLU roles and PCB components. Each task addresses distinct aspects of production systems, highlighting the strengths and limitations of various techniques:

- **Task 1** (Section 3.1) focuses on redundancy modeling using FTA. The goal is to explore different redundant configurations to improve reliability by minimizing the likelihood of system failures.
- **Task 2** (Section 3.2) shifts focus to parallel processes, using SPN to model and analyze the behavior of systems with concurrent operations and to study the impact of transient and permanent failures.
- **Task 3** (Section 3.3) leverages PMC with DEPM to evaluate complex error propagation scenarios and assess the impact of stochastic failures on system safety and reliability.

For each operational situation, the corresponding formalism is instantiated by mapping the detected PLU layout and modes to the model structure: system redundancies to FTA, concurrent operations to SPN, and error propagation paths to PMC.

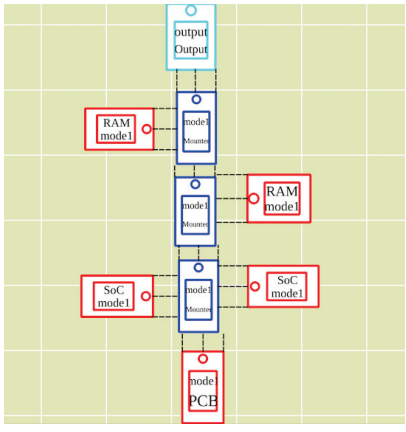
3.1. *Task 1: Redundancies with FTA*

Task 1 focuses on modeling system reliability with redundancy using FTA. FTA is a deductive method for identifying the root causes of system failures by mapping logical relationships in a fault tree Vesely et al. (1981). A Fault Tree (FT) describes how failures of system components and other undesired events (basic events) combine to produce a specific failure scenario (top event).

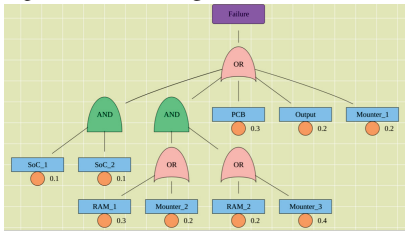
FTs employ logical gates such as AND, OR, and K-out-of-N to represent failure relationships. Redundancy is modeled in fault trees using AND gates, which signify alternative components or subsystems. For instance, a production line might have redundant mounters to assemble identical components. If one mounter fails, the other takes

over, thereby reducing the likelihood of the top event. Similarly, redundant input storages supplying the same components can also be modeled.

The task involves assembling a PCB with RAM and SoC modules. Users can configure redundant mounters or suppliers. Figure 4a illustrates a production line with three mounters: the first mounts a RAM module, the second serves as a redundant mounter for RAM, and the third mounts one of two SoC modules, demonstrating redundancy in SoC assembly. Note that these redundancies apply to the assembly process, not the assembled PCBs. The fault tree in Figure 4b visualizes these redundancies using AND gates. From this visualization, the user can identify potential improvements to the production setup. For example, making the RAM module connected to the first mounter redundant could increase reliability and reduce costs. Alternatively, users could configure two redundant mounters to handle the RAM and SoC modules. This way, FTAs provide valuable insights for optimizing production line reliability.



(a) Production line layout determined from captured camera image.



(b) Automatically created fault tree.

Fig. 4.: An example of a production line with redundancies and the corresponding fault tree.

3.2. *Task 2: Parallel compositions with SPN*

Task 2 focuses on modeling more complex scenarios involving parallelism, as well as permanent and transient failures, using SPN. Petri Nets are particularly effective for representing systems with parallel processes. An SPN consists of places, transitions, and tokens, which represent system states and activities. Tokens move between places via transitions, modeling parallel processes such as production lines where multiple components are assembled simultaneously. For example, in Figure 3b, the production line splits into two parallel parts that later merge.

In our SPN model: (i) Storages are places holding tokens that represent available components; (ii) Mounters and soldering stations are transitions that process tokens and pass them to the next stage; (iii) The output place represents the completion of the assembly process, holding finished PCBs. SPNs also account for transient and permanent failures. Each transition has a failure pattern linked to an ”ok” place, which holds a token indicating a healthy state. A failure transition, firing stochastically, removes this token, simulating a failure. Transient failures are modeled with a loop back to the ”ok” place, where the recoverability depends on the PLU’s mode.

Figure 5 illustrates an SPN for the production line layout shown in Figure 3b. This model allows users to analyze system behavior under various failure scenarios. Our Risk Assessment Module computes several safety metrics for Task 2.

Unlike Fault Trees, which only estimate the probability of a top event, Petri Nets leverage Probabilistic Computation Tree Logic (PCTL) to evaluate a wide range of stochastic system properties, including:

(1) Probability of producing N units:
$”P =? [F\ outputPlace = N]”$.
(2) Probability of a transient failure:
$”P =? [F(m\_failure\_state\_1 = 1)]”$.
(3) Probability of a process jams
$”P =? [F(s\_done]”$.

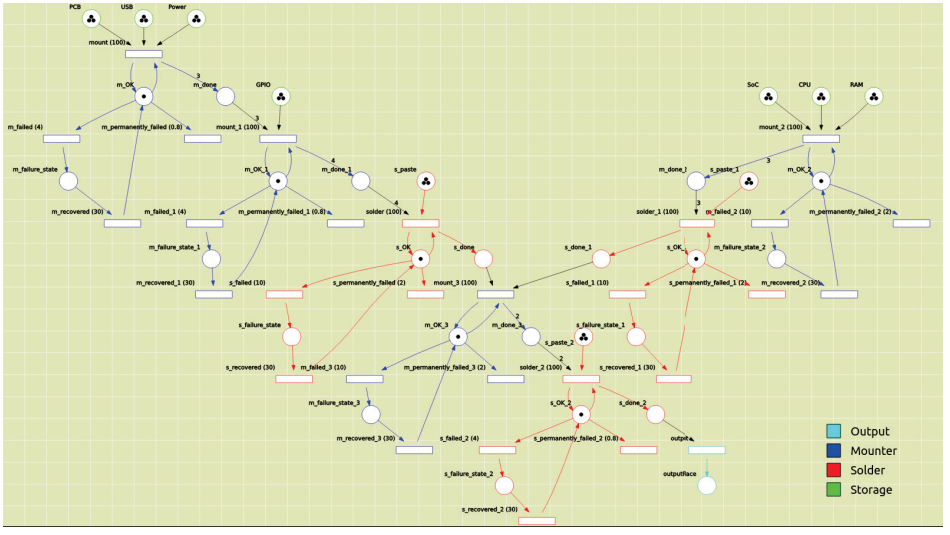


Fig. 5.: Petri Net model for the production line layout in Figure 3b.

3.3. *Task 3: Probabilistic Model Checking with DEPM*

Task 3 addresses safety aspects using PMC with custom PCTL metrics. The system is modeled using the highly customizable Dual-Graph Error Propagation Model (DEPM) Morozov and Janschek (2011). PMC is a powerful technique for verifying the reliability and safety properties of stochastic systems, such as those represented by DEPM. It allows engineers to quantitatively evaluate the likelihood of various failure scenarios, including transient and permanent failures, error propagation, and complex mitigation mechanisms.

In the DEPM model, data nodes (gray rectangles in Figure 6) represent PCB components, while executable element nodes (dark blue rounded rectangles) correspond to processes like mounting and soldering. Black arrows indicate control flow between executable elements, while blue arrows depict data flows. Red arrow-shaped rectangles denote specific failures to be probabilistically evaluated.

Technically, DEPM is grounded in a Discrete-Time Markov Chain (DTMC) formalism, modeled using PRISM language and analyzed with model checkers like PRISM Kwiatkowska et al. (2011) or STORM Dehnert et al. (2017). The STORM model checker is integrated into the demonstrator's software to evaluate safety metrics.

By incorporating stochastic failures, PMC captures real-world uncertainties. For example, each soldering process not only assembles components but also increases the PCB's temperature. A stochastic failure during soldering could result in overheating, posing a burn risk. Similarly, components may be mounted incorrectly, introducing defects that affect overall system reliability. PMC enables the modeling of such failure scenarios, providing quantitative risk assessments.

4. Key findings

This paper introduced a novel hardware/software demonstrator designed to explore risk assessment methodologies for industrial CPS. Through a series of tasks with increasing complexity, we analyzed the application of classical FTA, SPN, and advanced PMC using DEPM. The demonstrator highlights the strengths and limitations of each method, particularly in addressing dynamic reconfigurations, parallel processes, and complex failure scenarios, see details in Table 2.

A key theoretical contribution of this work is the systematic comparison of these methods on the same physical system, revealing their respective strengths and limitations in relation to re-

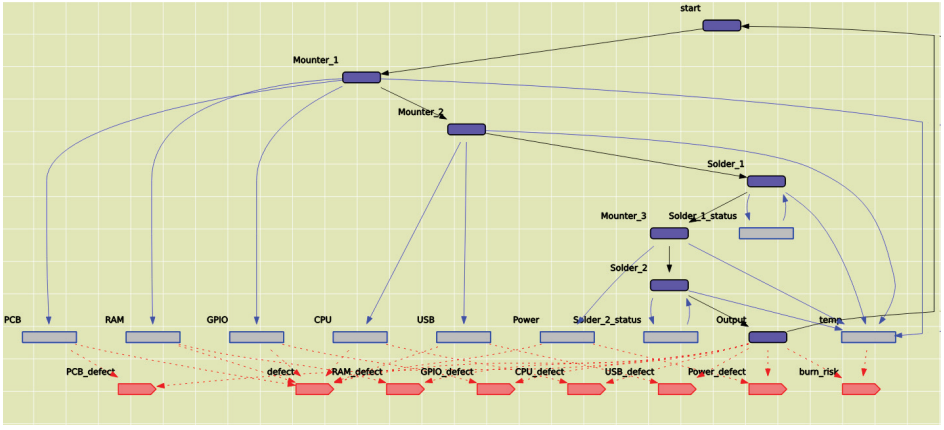


Fig. 6.: Example of a Dual-Graph Error Propagation Model (DEPM).

Table 2.: Comparison of the applied risk assessment methods.

Feature	**FTA**	**Stochastic PN**	**PMC with DEPM**
Modeling redundancies	easy	possible	possible
Modeling process-related failures	not supported	supported	supported
Modeling parallel processes	not supported	supported	supported
Handling multiple failure modes	limited	supported	supported
Customized failure metrics	not supported	limited	fully supported
Required expertise	low (basic risk analysis knowledge)	moderate (understanding of dynamic systems)	high (formal methods, stochastic modeling)
Required input data	low (basic event probabilities)	moderate (transition rates, failure patterns)	high (fault activation, error propagation, control flow probabilities)
Scalability and execution time	fast (seconds)	moderate (minutes; risk of state space explosion)	moderate to high (minutes to hours; risk of state space explosion)
Suitability for reconfigurations	moderate	high	high

configurability, parallelism, and complex failure propagation. Specifically, FTA provides intuitive insights for static systems and facilitates the modeling of various redundancy strategies. SPNs, in contrast, excel at handling parallel processes and can be easily adapted to model different types of failure modes, including both permanent and transient failures. Additionally, tools like STORM enable probabilistic model checking of SPN models, allowing not only for the computation of top-event probabilities (as in FTA) but also the evaluation of a wide range of reliability and safety metrics. PMC, when paired with DEPM, offers a robust framework for analyzing dynamic behaviors in highly complex industrial environments. This approach fully leverages the power of PMC, enabling the modeling of physical parameters, such as product temperature, and assessing the associated risks, such as overheating or related failures. However, as we progress to more advanced methods, greater expertise and richer input data are required. While FTA is intuitive and relies only on the probabilities of basic events, SPNs are more challenging to construct and demand a deeper understanding of system dynamics. Model checking SPNs additionally requires proficiency

in PCTL and careful model construction to avoid state-space explosion. DEPM offers significant flexibility but involves more complex formalism and demands probabilistic input parameters for each event.

5. Conclusion

Our findings demonstrate that hybrid risk modeling approaches, combining traditional and advanced techniques, are essential to address the challenges of dynamic, reconfigurable CPS. The developed demonstrator not only provides a testbed for such approaches but also serves as an educational tool for exploring trade-offs between safety, reliability, and operational costs. These insights contribute to advancing risk assessment methodologies for smart factories and guiding the design of future integrated frameworks.

Acknowledgement

This research is supported by the Bundesanstalt für Arbeitsschutz und Arbeitsmedizin (BAuA, Germany) under project number F2497.

References

Baier, C. and J.-P. Katoen (2008). *Principles of model checking*. MIT press.

Bolbot, V., G. Theotokatos, L. M. Bujorianu, E. Boulougouris, and D. Vassalos (2019). Vulnerabilities and safety assurance methods in cyber-physical systems: A comprehensive review. *Reliability Engineering & System Safety 182*, 179–193.

Dehnert, C., S. Junges, J.-P. Katoen, and M. Volk (2017). A storm is coming: A modern probabilistic model checker. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II 30*, pp. 592–600. Springer.

Giallanza, A., G. La Scalia, R. Micale, and C. M. La Fata (2024). Occupational health and safety issues in human-robot collaboration: State of the art and open challenges. *Safety Science 169*, 106313.

Häring, I. and I. Häring (2021). Technical safety and reliability methods for resilience engineering. *Technical Safety, Reliability and Resilience: Methods and Processes*, 9–26.

Huck, T. P., N. Münch, L. Hornung, C. Ledermann, and C. Wurll (2021). Risk assessment tools for industrial human-robot collaboration: Novel approaches and practical needs. *Safety Science 141*.

Kabir, S. and Y. Papadopoulos (2019). Applications of bayesian networks and petri nets in safety, reliability, and risk assessments: A review. *Safety science 115*, 154–175.

Kwiatkowska, M., G. Norman, and D. Parker (2011). Prism 4.0: Verification of probabilistic real-time systems. In *International conference on computer aided verification*, pp. 585–591. Springer.

Leimeister, M. and A. Kolios (2018). A review of reliability-based methods for risk analysis and their application in the offshore wind industry. *Renewable and Sustainable Energy Reviews 91*, 1065–1076.

Marsan, M. A. (1990). Stochastic petri nets: An elementary introduction. In G. Rozenberg (Ed.), *Advances in Petri Nets 1989*, Berlin, Heidelberg, pp. 1–29. Springer Berlin Heidelberg.

Molloy (1982). Performance analysis using stochastic petri nets. *IEEE Transactions on computers 100*(9), 913–917.

Morozov, A., K. Ding, M. Steurer, and K. Janschek (2019). Openerrorpro: A new tool for stochastic model-based reliability and resilience analysis. In *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 303–312.

Morozov, A. and K. Janschek (2011). Dual graph error propagation model for mechatronic system analysis. *IFAC Proceedings Volumes 44*(1), 9893–9898. 18th IFAC World Congress.

Vesely, W. E., F. F. Goldberg, N. H. Roberts, and D. F. Haasl (1981). Fault tree handbook. Technical report, Nuclear Regulatory Commission Washington dc.

Villani, V., F. Pini, F. Leali, and C. Secchi (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics 55*, 248–266.

Zacharaki, A., I. Kostavelis, A. Gasteratos, and I. Dokas (2020). Safety bounds in human robot interaction: A survey. *Safety Science 127*.