Collision hazard Modeling and Analysis in a multi-mobile robots system transportation task with STPA and SPN

Chaima Bensaci[(1)], Youcef Zennir[(2)], Denis Pomorski[(3)], Fares Innal[(2)], Mary Ann Lundteigen[(4)]

[(1)]Université 20 Août 1955 Skikda, LGCES Laboratory
Skikda, Algeria
ch.bensaci@univ-skikda.dz

[(2)]Université 20 Août 1955 Skikda, Automatic Laboratory of Skikda
21000 Skikda, Algeria, tel.:00213664735277/fax.:0021338723128
y.zennir@univ-skikda.dz;
[(2)]Université 20 Août 1955 Skikda, Automatic Laboratory of Skikda
2100 Skikda, Algeria, tel
fares.innal@univ-skikda.dz
[(3)]University ofLille, CRIStAL Laboratory– UMR 9189
Lille, France
denis.pomorski@univ-lille.fr
[(4)]University of NTNU, Departement of Engineering Cybernetics
Trondheim, Norway
mary.a.lundteigen@ntnu.no

Preprint version published in Journal of Safety and Reliability Engineering: https://doi.org/10.1016/j.ress.2023.109138

Abstract

This paper investigates the ability to use complex multi mobile robotic systems in risky and dynamic environments, such as industrial plants and laboratories, with the presence of human factor. More specifically, it presents an approach to analyze the dominant risk, extract, model and quantify the hazard scenarios, then propose requirements using the combination between System Theoretic Process Analysis (STPA) and Stochastic Petri Nets (SPN).This approach is demonstrated with a case study related to a chemical transportation task within a miniature analysis laboratory in oil and gas industry. The main purposes of this article are: to investigate how risk and safety of these systems should be managed, to create a framework for modeling and quantifying collision hazard, and to generate the needed constraints and requirements to improve the operation of robots and preserve safety in the laboratory.

The results obtained reveal that the main problems that may hinder the safe operation of robots are those of control, communication, power and retrieving sensors information. The proposed safety measures contribute to improve autonomy features in mobile robots. The STPA-SPN combination offers a better modeling and assessment of robots performance as well as their collision hazard frequency.

Key words

Hazard Analysis,STAMP, System Theoretic Process Analysis (STPA), Autonomous multi-mobile robots, Stochastic Petri Nets (SPN), Collision Hazard Frequency.

1. Introduction

The current progress in mobile robots technology pushes the world to revolutionize several industry sectors (automotive, nuclear, petrochemical, …); mobile robots will facilitate human's labor by helping him/her to perform different tasks that are very often difficult and tiring for

workers like cleaning, environmental surveillance, objects transport,... especially with the advent of so-called autonomous control which allows the robot self-navigating without any human intervention. Recently, a lot of researchers care about robotics and deal in their works with safe navigation issues (Liu, 2017),(Fan et al., 2018),(Tang, Thomas and Kumar, 2016),(Szatmary and Richert, 2017),(Liu et al., 2017), (Pandey and Parhi, 2016), (Li and Savkin, 2018); coordination between multi robots(Li et al., 2012),(Li, Kong and Guo, 2014),(Yoo and Kim, 2015), (Mendiburu, Morais and Lima, 2016), (Jin et al., 2019) and with Human-Robot Interaction (Lasota, Fong and Shah, 2017). Moreover, as safety in every institution and for every application is of prime importance, it is a barrier against the widespread of robotic applications(Saenz et al., 2018). Therefore, a number of researchers directed toward managing risks and safety design sectors in which a set of methods or combinations are employed to dealing with the issue of safety assessment in robotic applications.(Kazanzides, 2009)has combined between two conventional methods Failure Mode and Effects Analysis (FMEA) and Fault Tree Analysis (FTA), the aim is to analyze hazards and implement a safety control loop for a surgical robot. A reverse of the previous approach has been proposed by (Lee and Yam, 2012)where FTA and FMEA are used to examine the potential failures of Human-cooperative robots (HCRs) which lead to a hazardous movement of the robot and provide the suitable safety functions against failures with the required safety level. A novel Hazard and Operability studies (HAZOP) has been proposed by (Böhm and Gruber, 2010) applied for a therapeutic mobile robot. The authors tried to coverage by their approach all kinds of hazards associated with components and related to operations.(Alexander, Herbert and Kelly, 2009)have combined a set of techniques to identify hazards and derive safety requirements for autonomous systems; Energy Trace and Barrier Analysis (ETBA) and checklists firstly used to provide a basic hazard identification then a detailed analysis provided using Functional Failure Analysis (FFA)and HAZOP method. Another approach based on SHARD analysis (Software Hazard Analysis and Resolution in Design) along with a hazard check list is implemented for a personal autonomous mobile robot by (Woodman *et al.*, 2012).(Dogramadzi *et al.*, 2014)have developed an environmental approach called Environmental Survey Hazard Analysis (ESHA) for hazard identification of autonomous mobile robot applications.(Martin-Guillerez *et al.*, 2010), (Machin, 2015) and (Guiochet, 2016) have discussed in their studies the issue of interaction between human and robot by developing the HAZOP technique basing on a unified modeling language (UML) which have been implemented together for a mobile manipulator robot. Nevertheless, the increased complexity of new robotic systems composition, due to the intensive use of software, level of autonomy which characterize their control systems and the number of interaction among components, lead to the appearance of new hazard type which caused by new causal factors rather than failure of components. Traditional risk analysis techniques such as FMEA, HAZOP, FTA…,based on reliability theory and functional decomposition, that describe the system as a number of almost independent subsystems(Scarinci *et al.*, 2019), are no longer adequate to represent indirect interactions among system components and to consider conditions which cause inappropriate behaviors.

Last decades, Leveson developed a method called System Theoretic Process Analysis STPA(Leveson, 2011),(Leveson and Thomas, 2018)based on system theory which considers safety as a system's control problem rather than a component failure problem. STPA is a hazard analysis technique dedicated to identify hazards of automated complex systems and their control/ command architectures which include interactions between their components; that interests to handle hardware, software, human operators and integrate them in a unified process (Scarinci et al., 2019). The STPA method has proven its strength and capability over last few years through its analytical capacity of many varieties of automated and high complex systems including:driving systems(Abdulkhaleq et al., 2018),aircraft Systems(Plioutsias and Karanikas, 2015), (Scarinci et al., 2019), aerial transportation systems(Fleming et al., 2013),aerospace exploration systems (Ishimatsu et al., 2014),robotic surgical systems(Alemzadeh et al., 2014); even autonomous ones such as: autonomous vehicles,(Wróbel, Montewka and Kujala, 2018), autonomous vessels(Banda and Kannos, 2017),autonomous ships (Rokseth, Haugen and Utne, 2019). In addition, STPA has been applied for several industries such as power industry(Khan, Madnick and Moulton, 2018); Subsea (Rachman and Ratnayake, 2015),(Kim et al., 2018), (Zhang et al., 2019b); process industry (Hardy and Guarnieri, 2011), (Rodríguez and Díaz, 2016) and Nuclear industry(Song, 2012),(Thomas, Lemos and Leveson, 2012),(Lee et al., 2013), (Uesako, 2016). Furthermore, a number of comparison studies of STPA hazard analysis with different conventional risk analysis methods have reported positive evaluation about applying STPA on various complex systems, such as(Ishimatsu et al., 2010)who have compared STPA analysis results with FTA results. This comparison showed that STPA identifies additional causal factors than the ones identified by FTA. (Nakao et al., 2011)conclude that STPA makes the generation of safety requirements and constraints easily and flexibly. Also, (Fleming et al., 2013)argue that STPA allows the identification of faults related to software and dynamic behavior of systems, unlike the classical methods such as FMEA, FTA, etc.

It is true that STPA has attracted a big attention from a large number of researchers in a short period of time due to its suitability for complex systems and due to its benefits mentioned above. However, the original STPA is still fully qualitative technique. It does not focus on modeling and assessing hazard scenarios which is considered as a significant step in safety analysis. It is incapable yet to quantify risks. In this context, we aim in this study to deal with this issue by combining STPA with other method allowing to obtain a quantification of risk scenarios. In this article, we adopt as a quantitative method Stochastic Petri Net (SPN), which is best suited in modeling of complex and dynamic systems behavior. Petri Nets (PN)are a state-transition approach that has been proposed to generate large-scale Markov processes (SIGNORET, 2008). It has shown excellent modeling and calculation skills especially when it is coupled with Monte-Carlo simulation in the field of dependability and safety(Malhotra and Trivedi, 1995; Dutuit et al., 1997; Signoret, 2009).

This combination intends to make the analysis more effective and to better assess hazard scenarios. From literature, some research studies have proposed to combine STPA with other methods that offer an evaluation to the risk such as FMEA(La and Kwon, 2018) and Bowtie (Bensaci *et al.*, 2020) for semi-quantitative analysis. Further, there exist a proposal to combine STPA with SPN for process industry by (Zhang *et al.*, 2019b).Their aim was to conduct quantitative models based on STPA for only feedback control loops behavior, not for system behavior as a whole.

In this study, the approach is conducted for a multi-robots system by modeling the whole system behavior during its operation. The main purposes of this article are therefore to investigate how risk and safety of autonomous multi-robots systems should be managed, to create a framework for modeling and quantifying the collision hazard, and to generate the needed constraints and requirements to ensure the safe operation of robots. The case study used in this article investigates the use of a set of mobile robots for chemical transportation tasks within high-risk environments such as industrial laboratories in oil and gas industries.

The remainder of the paper is organized as follows: Section 2 presents the methodology proposed in this article, introduces the System-Theoretic Process Analysis (STPA) and a background on Petri nets for risk and safety uses is also presented. The application of the proposed methodology to an autonomous multi-robots work in chemical laboratory is given in section 3. Section 4 is devoted to results discussion. Finally, conclusion is made in Section 5.

2. Methodology Description

The proposed methodology is a combination of two methods, namely:

- System-Theoretic Process Analysis (STPA)that is used to identify and analyze hazard scenarios
- Petri net method (PN) is performed to model the functional and dysfunctional behaviors of robots and their coordinating actions, in order to assess the occurrence frequency of collision scenarios and robots performance.

The proposed methodology follows the steps illustrated in figure 1.

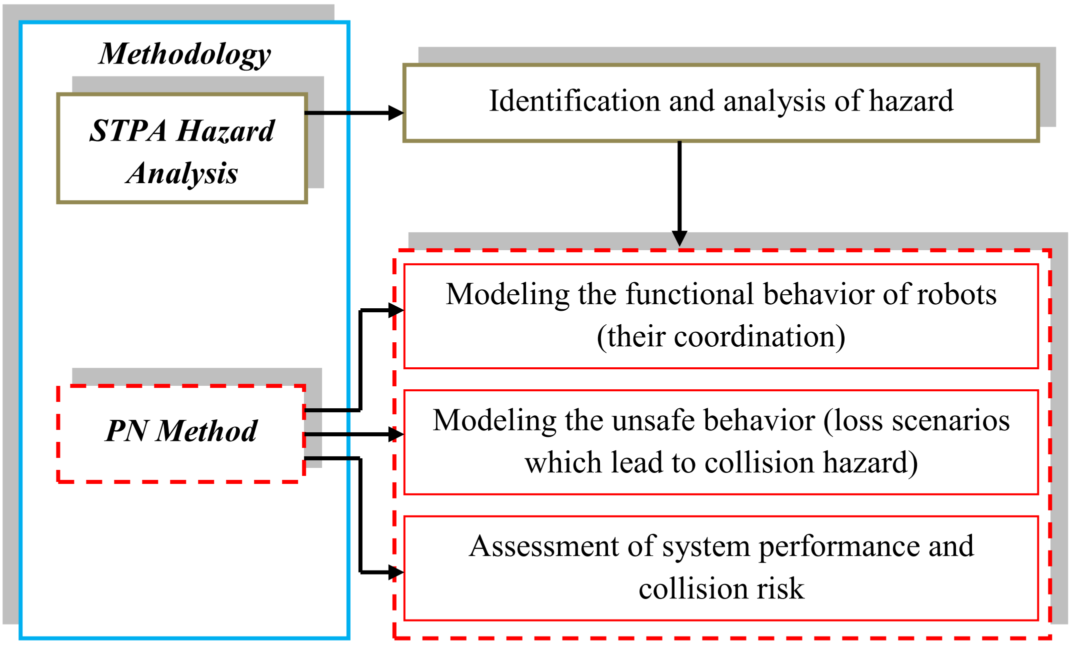

Methodology
STPA Hazard Analysis
PN Method
Identification and analysis of hazard
Modeling the functional behavior of robots (their coordination)
Modeling the unsafe behavior (loss scenarios which lead to collision hazard)
Assessment of system performance and collision risk

Fig. 1.Proposed methodology

2.1.STPA Hazard Analysis

System Theoretic Process Analysis (STPA) is a purely qualitative method for hazard analysis based on Systems-Theoretic Accident Model and Processes (STAMP) which has been developed by (Leveson, 2004). It has been specially created for complex automated systems as long as today the automation invaded multiple domains and applied first to identify undesired/unsafe system behaviors through a structured, top-down approach. Requirements are subsequently generated from the results of STPA in order to handle these unsafe behaviors. STPA treats safety as a control problem rather than a failure prevention problem(Khan, Madnick and Moulton, 2018).In this context, the STPA method is suited to distinguish potentially dangerous design imperfections, including software and hardware errors, external disturbances and improper interactions among various control structure components, which result mainly from lacking control or violation of safety constraints related to the development, design, and operation of the system (Ishimatsu et al, 2014).

The STPA analysis is carried out in five main steps as shown in the following figure 2 (Leveson and Thomas, 2018).

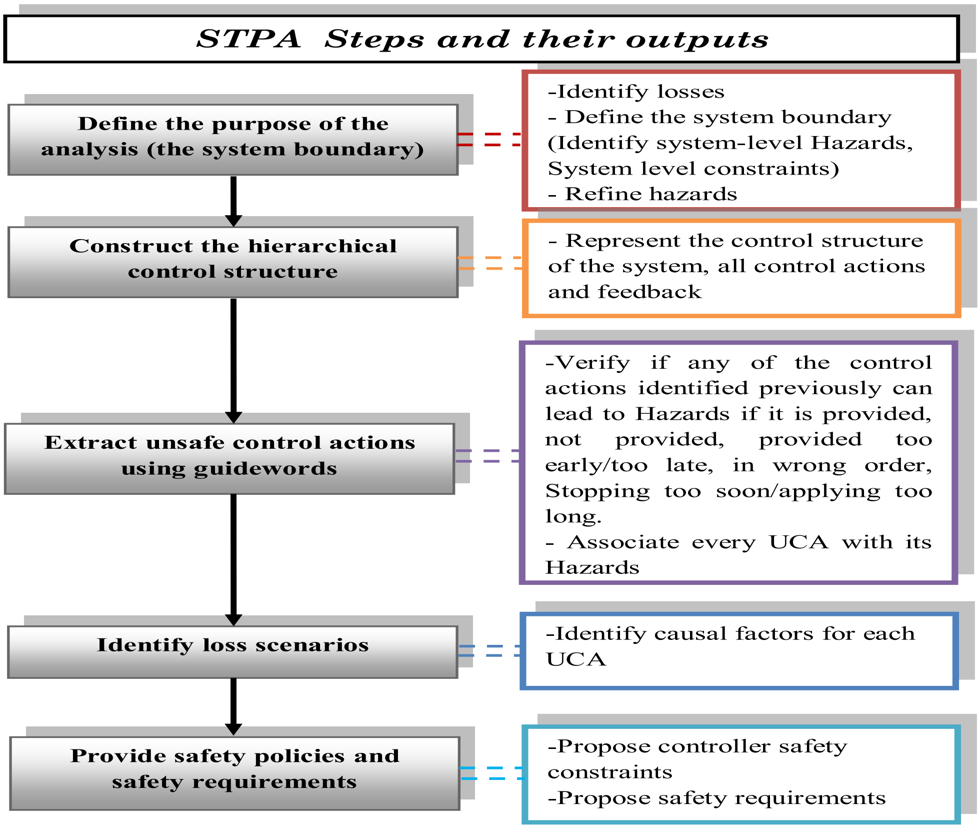

STPA Steps and their outputs
Define the purpose of the analysis (the system boundary)
-Identify losses
- Define the system boundary (Identify system-level Hazards, System level constraints)
- Refine hazards
Construct the hierarchical control structure
- Represent the control structure of the system, all control actions and feedback
Extract unsafe control actions using guidewords
-Verify if any of the control actions identified previously can lead to Hazards if it is provided, not provided, provided too early/too late, in wrong order, Stopping too soon/applying too long.
- Associate every UCA with its Hazards
Identify loss scenarios
-Identify causal factors for each UCA
Provide safety policies and safety requirements
-Propose controller safety constraints
-Propose safety requirements

Fig. 2.STPA methodology. (UCA: Unsafe Control Action)

- The first step in any analytical method is to define its purpose, mainly by identifying the types of losses, accidents and high level hazards of the system under study.
- The second step is to model the control structure of the system. The control structure constructs as a set of feedback control loops, which defines functional relationships and interactions between its components.
- The third step is to identify unsafe control actions and connect every unsafe control action with their hazards defined in the first step. Control actions are analyzed using predefined guide words by verifying:
  - The control action causes a hazard if it "is provided".
  - The control action or the preventive measure causes a hazard if it is "not provided".
  - The control action causes a hazard if it "is provided too early or too late".
  - The control action causes a hazard if it is “applied for a long time or lose it too soon”.
- The fourth step is to extract the causal factors for which unsafe control action could occur at the system level.
- Finally, the fifth step consists of proposing safety constraints and safety requirements to generate safety policies for the development, design, control algorithm and operating part of the system.

2.2.Petri Net (PN)

Petri nets (PNs) were first introduced by Carl Adam Petri in his doctoral thesis presented in 1962. Its initial goal was a graphic representation of the behavior of automata. At the beginning of the 1980s, they were proposed for the first time to be used in dependability in order to generate Markov processes on a large scale, then they were considered as a model suitable for Monte-Carlo simulation(SIGNORET, 2008). The Petri net is a powerful modeling formalism suitable for dynamic discrete event systems, combining a well-defined mathematical theory with a graphical representation of the behavior of the systems. Its theoretical aspect provides a modeling and behavioral analysis of the dynamics of the system, while its graphical representation allows to visualize the changes in the modeled state of the system (Wang, 2012). Therefore, PNs have been used to model various types of complex systems such as control systems (Andreadakis and Levis, 1988), robotic systems (Milutinovic and Lima, 2002), communication systems (Wang, 2007). Thus, they have shown excellent modeling and calculation capacities in the field of dependability and safety(Malhotra and Trivedi, 1995; Dutuit *et al.*, 1997; Signoret, 2009). They are very useful for modeling the functional and dysfunctional behavior of systems and for evaluating several performances(SIGNORET, 2008). For example, many existing works have used PNs to assess reliability (Kumar and Aggarwal, 1993; Yang *et al.*, 2011) and availability (Jian, Shaoping and Yaoxing, 2008; Kumar, Jain and Gandhi, 2012; Zhang *et al.*, 2019a).

As described by Petri, the basic PN is a special type of directed bipartite graph, mainly composed of three complementary parts:

- ***A static graph***, which does not change over time. It is made up of three basic elements:

  • Places, represented as circles, each place represents a state or condition.

  • Transitions, represented by bars, each transition represents an event, a transformation or a change of state.

  • Directed arcs, represented by arrows, connect squares to transitions or transitions to squares. Each arc can be assigned with a natural number, named a weight (normally assumed to be 1).
- ***Dynamic elements***, called tokens and represented by small balls which indicate the state of the system at a given moment. That makes it possible to describe the behavior of the modeled system. Tokens can represent objects, machines, humans, information, conditions, etc. The distribution of tokens in the squares is called network marking. The presence or absence of a token in a place can indicate whether a condition associated with that place is true or false. If each entry place "*p"* of the transition "*t*" contains at least the number of tokens equal to the weight of the directed arc connecting "*p"* to "*t"*, then this transition "*t"* is activated.
- ***Predicates and assertions***, which can be introduced in the PN by means of variables. The predicate (often preceded by two question marks "??") is a condition to enable or disable transitions when variables are checked or not checked, and the assertion (often preceded by two exclamation marks "!!") is a formula for updating variables after crossing the associated transition (Signoret, 2008; Wang, 2012). The use of these two elements can improve the readability of complex PN models, thus improving their comprehensibility(Signoret *et al.*, 2013).

In the real world, almost all events are related to time. For this, the need to include temporal variables in models of dynamical systems is obvious. When a Petri net integrates the time variable into the model of the system, it becomes a timed Petri net. Timed PNs are an extension of basic Petri Nets for which a time variable called delay with its probability density function is associated with each transition. Tokens can move between squares when the activated transition is fired. The transition is fired when the associated delay elapses (since the transition remains on during the delays). The delay between the activated transition and the crossing can be characterized as fixed (deterministic) or random (stochastic). Once the transitions are characterized by deterministic and stochastic delays in a model, Petri nets will become difficult to solve analytically. In this case, a Monte Carlo simulation is generally coupled with the SPN model in order to facilitate the evaluation of system's performance (SIGNORET, 2008).

3. Application to Autonomous Multi-mobile robots system work in petrochemical plants

In this section, STPA is applied to analyze the functioning of a multi-mobile robot moving products in a chemical laboratory, and then SPN is used to model their behavior as well to assess the collision hazard. A brief description of the object of study is presented first, followed by the presentation of the approach application with the derived results.

3.1. System description

Our system is composed of two wheeled mobile robots that cooperate together in order to transport dangerous chemicals (toxic, flammable, explosive ...) within a chemical analysis laboratory, in the presence of analysis machines and human workers. The laboratory has five main rooms represented in Figure 3, a big room for analysis, rooms for chemicals storage and one other room for providing analysis results. The multi-robot system coordinates their motion according to a distributed architecture. The presence of these robots in such hazardous plants should carry a high level of dependability and safety, which inevitably demands the need for systems that are increasingly available, reliable and the need for optimal control and good communication between robotic entities. The objective of our study is therefore to collect the constraints and safety requirements necessary to ensure the good operating conditions of these robots in such environment, to assess the system performance and the occurrence frequency of collision hazard.

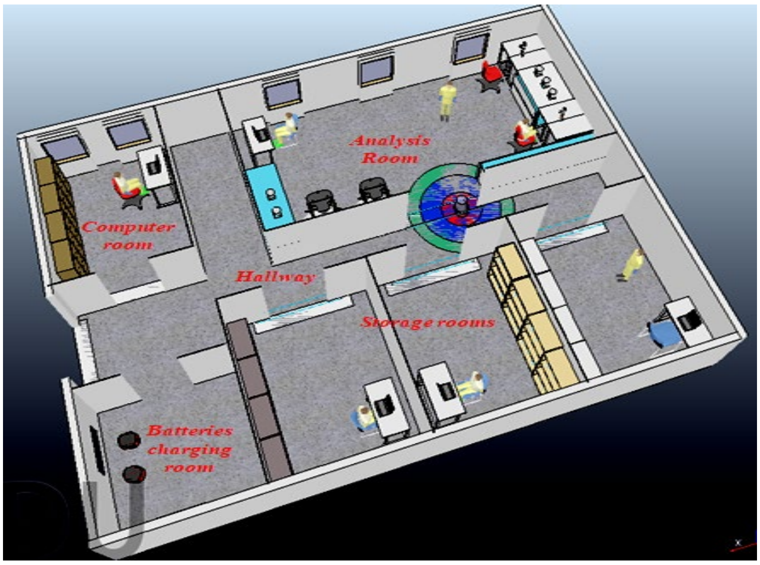


Fig. 3.Model of the working environment of robots.

3.2. Hazard analysis with STPA

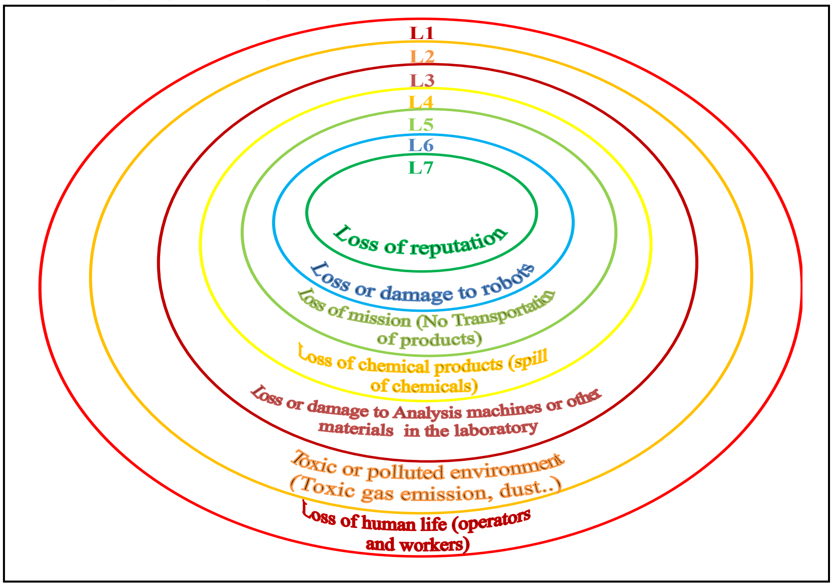
We present here the application of the STPA analysis step by step as well as its obtained results.

- **Step1**: defining the purpose of the analysis (define system boundary). This step contians the following substeps :

- **Identify losses:**

Seven types of losses are identified and classified in decreasing way in figure 4:

- **L1:** Loss of human life (operators and workers), human injuries, illnesses, asphyxia, poisoning…
- **L2:** Toxic or polluted environment (toxic gas emission, dust, fume of chemicals emission, chemicals leakage...)
- **L3:** Loss or damage to analysis machines or other materials in the laboratory
- **L4:** Loss of chemical products (spill of chemicals)
- **L5:** Loss of mission (no Transportation of products→ no analysis)
- **L6:** Loss or damage to robots
- **L7:** Loss of reputation of the company producing the robots.


L1
L2
L3
L4
L5
L6
L7
Loss of reputation
Loss or damage to robots
Loss of mission (No Transportation of products)
Loss of chemical products (spill of chemicals)
Loss or damage to Analysis machines or other materials in the laboratory
Toxic or polluted environment (Toxic gas emission, dust..)
Loss of human life (operators and workers)

-

Fig. 4.Classification of losses levels.

- **Identify system-level Accidents, Hazards and safety constraints:**

The table 1 below clearly identifies the accidents, hazards and high level constraints of the system. For each hazard, a constraint must be formulated.

Table 1. Table lists system-level accidents, hazards and constraints.

System- level Accidents (S-lA)	*System-level Hazards (S-lH)*	*System-level constraints (S-lC)*
S-lA1: Collision between robots loaded with chemicals (two or more) **S-lA2:** Collision between robot (loaded with chemicals) and human worker **S-lA3:** Collision between robots (without chemicals) **S-lA4:** Collision between robot and human worker (without chemicals) **S-lA5:** Robot crashes to the wall/ or other static obstacles (analysis machines) or falls down **S-lA6:** Chemicals spill	**S-lH1:** Robots (loaded with dangerous chemicals or not) enter together in a narrow area (Robots doesn't respect the safe distance between them or to humans) (L1- L7) **S-lH2:** Robot (loaded with dangerous chemicals or not) doesn't respect the safe distance to other static obstacles (L2- L7) **S-lH3:** Robot (loaded with dangerous chemicals or not)doesn't respect the safe velocity in its navigation (L1- L7) **S-lH4:** Robot (loaded with dangerous chemicals or not) enters into a slippery area (products spill) (L1- L7) **S-lH5:** Robot (loaded with dangerous chemicals or not) enters in an uncontrolled state (L1- L7)	**S-lC1:** The robots must not enter in the same area together especially if the area is narrow (the robot must detect if there is a place to enter) (one by one) → **S-lH1** **S-lC2:** The robots have to respect the safe distance and doesn't violate the minimum distance of separation and if it violated, then the violation must be detected to avoid any collision → **S-lH2** **S-lC3:** The robots must not exceed certain limit speed, especially in the case of carrying the products and if the limit speed exceeded, then the robot must be stopped → **S-lH3** **S-lC4:** Must maintain permanent cleanliness of the floor and not throw slippery materials on the floor → (**safety constraint to S-lH4 outside the boundary of the robot control**) **S-lC5:** The robots must detect slippery to avoid and if it not the case they must adjust their velocity and slow down → (**safety constraint to S-lH4 Relating to robot control**) **S-lC6**: The control parameters and algorithms of robots must be monitored and if there is any issue in the control (or indications that the robots are not well controlled) the robot must be stopped → **S-lH5**

- **Step2:** model the control structure for autonomous control

The goal of the STAMP approach is to increase our understanding of systems and processes control structure, and to model the control structure based on functional diagram. The studied system is composed of two autonomous controllers. The two controllers communicate together to coordinate the motion of the robots.

- **High level Functional control block diagram of one robot:**

Before moving on to the control architecture of an autonomous mobile robot we thought to start by defining its functional model to identify the main components and the function of each component. As the following figure 5 is shown, an autonomous robot's navigation revolves around three main tasks: perception and location (sensors), planning (controller), and control (motors).

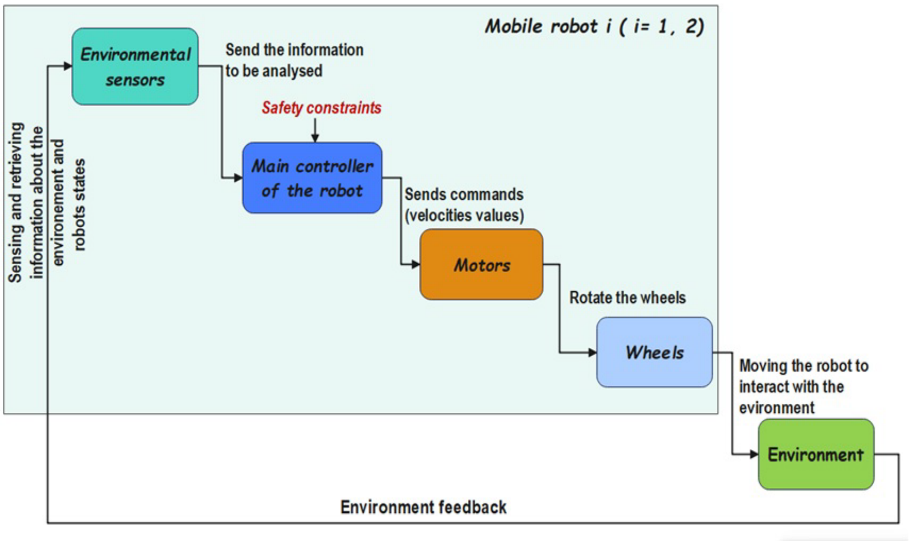

Mobile robot i (i= 1, 2)
Environmental sensors
Send the information to be analysed
Safety constraints
Main controller of the robot
Sends commands (velocities values)
Motors
Rotate the wheels
Wheels
Moving the robot to interact with the evironment
Environment
Environment feedback
Sensing and retrieving information about the environement and robots states

Fig. 5.Functional control block for an autonomous mobile robot.

- **Detailed functional control block diagram for one robot:**

For more details, detailed functional control architecture is proposed according to the STAMP model and illustrated in figure 6. This architecture defines the main parts of an autonomous mobile robot identified by the function of each part; these parts interact with each other by control actions (arrows in red) and its feedback (arrows in green).

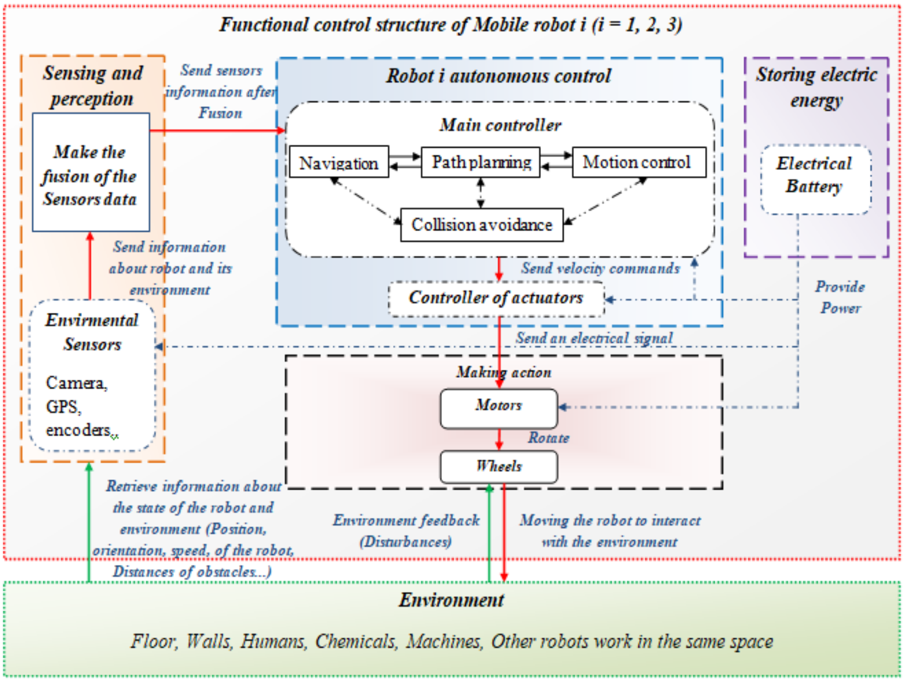

Functional control structure of Mobile robot i (i = 1, 2, 3)
Sensing and perception
Make the fusion of the Sensors data
Send sensors information after Fusion
Robot i autonomous control
Main controller
Navigation
Path planning
Motion control
Collision avoidance
Storing electric energy
Electrical Battery
Send information about robot and its environment
Send velocity commands
Controller of actuators
Provide Power
Envirmental Sensors
Camera, GPS, encoders,..
Send an electrical signal
Making action
Motors
Rotate
Wheels
Retrieve information about the state of the robot and environment (Position, orientation, speed, of the robot, Distances of obstacles...)
Environment feedback (Disturbances)
Moving the robot to interact with the environment
Environment
Floor, Walls, Humans, Chemicals, Machines, Other robots work in the same space

Fig. 6.Part 1- detailed functional control architecture for one autonomous mobile robot.

- **Detailed Functional control block diagram for two robots:**

In the case of an autonomous multi-robots (case of two robots) operating in a common indoor environment, coordination of actions is of prime importance between robots, in particular, for distributed control. This coordination is carried out through communication protocols (messages, sharing information, etc.). The functional control architecture for two robots coordination is shown in figure 7.

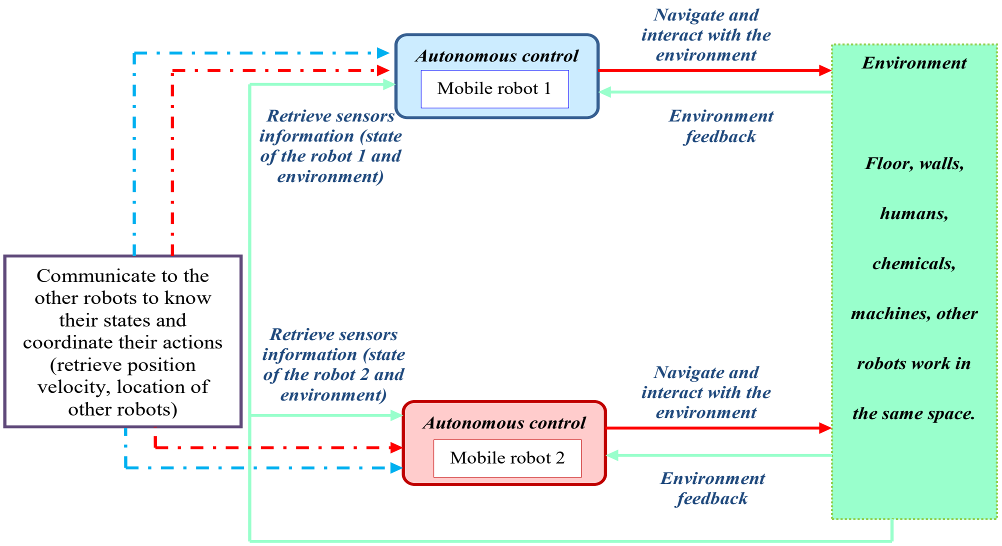

Navigate and interact with the environment
Autonomous control
Mobile robot 1
Environment
Retrieve sensors information (state of the robot 1 and environment)
Environment feedback
Floor, walls,
humans,
chemicals,
machines, other
robots work in
the same space.
Communicate to the other robots to know their states and coordinate their actions (retrieve position velocity, location of other robots)
Retrieve sensors information (state of the robot 2 and environment)
Navigate and interact with the environment
Autonomous control
Mobile robot 2
Environment feedback

Fig. 7.Part 2-functional control architecture for multi- robots coordination (case of two robots).

- **Step3&4**: identify unsafe control actions and their causal factors

Table 2 presents the set of hazard scenarios identified for each functionality of autonomous mobile robot controller and its causal factors.

Table 2. Possible scenarios obtained from STPA application.

Control action/ communication	*Guideword*	*Scenario (UCA)*	*S-IH*	*Causal factors*
Plan and generate the path *(From path planning controller)*	*Not providing causes hazard*	**UCA1**: The controller doesn't calculate and generate the path	**H5**	- Planning and calculation units fail to perform their function (component failure) - Insufficient information to support calculation (unclear map or sensors problem) - High calculation number → the calculation is blocked because of the huge number of equations - Memory card saturation
	Providing causes hazard	**UCA2**: The controller generates the wrong path which lead to the wrong destination	**H5**	- Insufficient sensor information - Failure of sensors - False algorithm - programmer error - Wrong order of operator
		UCA3: The controller generates the long path (without optimization)	**H5**	- Problem in the optimization algorithm , ineffective approach, inappropriate optimization criteria
		UCA4: The controller doesn't update the path in a dynamic environment	**H1**	- Repeated sensors data , - Failure of controller, sensors - Inadequate Algorithm
	Too early, too late, out of order	**UCA5**: Take a long time to calculate and generate the path	**H1,H2, H5**	- High Number of calculation - Lock of controller - Delay in sensor information - Problems in data fusion - Inadvertent lock of software, calculation
	Stopped too soon, applied too long	**UCA6**: The planning of path stopped too soon	**H1, H2,H5**	- Instant cut of data - Discharged battery
	Not providing causes hazard	**UCA7**: Collision avoidance feature is not provided during the robot navigation	**H1, H2, H4, H5**	- Failure of sensors (doesn't indicate if there is an obstacle) - Failure of collision avoidance controller

Avoid collision ***(from collision avoidance controller)***				-Inadequate control algorithm (which relate between features) - No dependency between sub-controllers
	Providing causes hazard	**UCA8**: Providing the wrong command to avoid collision (speed up / speed down, stop, turn left, turn right)	**H1, H2, H5**	- Inadequate calibration of sensors/ data fusion - Missing sensors /communication data
	Too early, too late, out of order	**UCA9**: Collision avoidance feature provided too late (static obstacle, other robots or human)	**H1, H2**	- Delay of sensors information - Missing data from sensors - Missing Data from communication (too late or too early) - Loose dependency between controllers
	Stopped too soon, applied too long	**UCA10**: The collision avoidance feature stopped too soon during the robot navigation	**H1,H2, H5**	- Inadequate collision avoidance Algorithm (non robust) - Failure of controller, the sensor which detects the distance between the obstacles and the robot - Inadequate sensors calibration
Establish and maintain dependency between sub-controllers (Software)	*Not providing causes hazard*	**UCA11**: Doesn't provide the dependency between sub-controllers (path tracking or path planning controller and collision avoidance controller…)	**H1,H2, H5**	- No data input from the sensor which detect obstacles (failure of sensors) or insufficient. - Programmer error - Inadequate control algorithm - Failure of software - Failure of node connecting between controllers
	Too early, too late, out of order	**UCA12**: Providing the dependency too late	**H1,H2, H5**	- Input data sent from sensors too late - Take more time in data fusion - Bug of software
	Stopped too soon	**UCA13**: Loose dependency between sub-controllers	**H1,H2, H5**	- Lock of software - Program corruption - Failure of node connecting between controllers / or lose the node. - Lose connection with master
Establish and maintain communication between robots	*Not providing causes hazard*	**UCA14**: Doesn't provide communication between robots	**H1**	- No Connection between robots - Failure of communication components
	Providing causes hazard	**UCA15**: Provide insufficient information **UCA16**: Provide huge information even unnecessary ones	**H1,H5**	- Programmer error - Cyber-security issues
	Too early, too late, out of order	**UCA17**: Provide communication with delay (too late)	**H1**	- Weak flow of network - Strong flow of data - Loss of power
	Stopped too soon, applied too long	**UCA18**: Loose communication between robots	**H1**	- Lock of communication software - Failure of communication components - Received information higher than processor capacity - Weak flow of network
		UCA19: The communication stopped too soon	**H1**	- Network interruption
Track the path ***(from Path tracking controller)***	*Not providing causes hazard*	**UCA20**: The controller doesn't track the path	**H2,H5**	- Inadequate path tracking control algorithm - Failure of motion controller - Failure of position sensors, - Inadequate sensors fusion - Inadequate sensors calibration - Failure of actuators(blockage of wheels)
	Providing causes hazard	**UCA21**: Track the path with no precision	**H5**	- Inadequate tracking control algorithm - Failure of sensors, actuators - Inadequate sensors calibration - Provide sensors data with delay - Inadequate data fusion

	Stopped too soon, applied too long	**UCA22**: Stooped to track too soon (interruption)	**H5**	- Interruption in sensors information - No Power (battery failure or Discharge) - Failure of controllers, sensors or actuators
Send the velocity command to the wheels ***(from motion controller after aggregation of the set of features)***	*Not providing causes hazard*	**UCA23**: The velocity command is not provided to the wheels when the robot is near to a dynamic obstacle (human , other robots)	**H1**	- Problem in communication block (command publisher) - Inadequate aggregation Algorithm (decision problem) - Failures of components of the main controller - Lock of software (of controllers and sensors) - Sensors information doesn't received by controller (problem in subscriber blocks) - Sensors information badly fused - Failure of sensors - Failure of actuators controller (no signal provided to the wheels)
		UCA24: The controller does not provide the velocity command to the right/left wheel during the robot navigation	**H5**	- Inadequate sensors information - Failure of actuators controller (no signal provided to the wheel)
	Providing causes hazard	**UCA25**: The controller provides high velocity in inappropriate situations (Robot carry chemicals, slippery floor, smoke, human work with robots in the same room...)	**H3,H4**	- Inadequate control algorithm (the controller not adapt to these situations) - Sensors doesn't detect slippery/ the load - Inadequate sensors calibration - Lack of sensors
		UCA26: The controller provides the same value of velocity even if the robot in front of an obstacle	**H1,H2, H3**	- Missing sensors indication/ or wrong fusion - Lock of software/ controller
	Too early, too late, out of order	**UCA27**: Send the command too late after a delay time	**H1,H2, H5**	- Send sensors information too late (delay) - Problem controller software - Memory saturation
	Stopped too soon, applied too long	**UCA28**: The sending command stopped too soon	**H1,H2**	- Failure of main controller/ controller of actuators, sensors. - Inadequate sensors information (doesn't update information about the environment) - Loose communication with master
		UCA 29: The same value of command applied too long even in new situations	**H1, H2,H3**	- Failure of sensors. - Inadequate sensors information The sensor indicates the same environment state (don't update new data about the environment)

- **Step5**: generate the important safety constraints and safety requirements

The important safety constraints and safety requirements proposed to ensure the good control of autonomous multi-robot in hazardous environment are detailed in Table 3.

Table 3. Important safety constraints and safety requirements proposed from STPA results.

Control Action/ communication	***UCA***	***Safety constraints***	***Suggested safety requirements***
Plan and generate the path	UCA1/ UCA 3	The controller must calculate and generate the correct and the optimized path.	The controller must choose the safe trajectory which takes the shortest time and that requires to choose the good optimization criterion
	UCA4	Planning should always be adapted to changes in the environment	The path shall be updated after each action of the robot or make the combination with the obstacle avoidance controller
	UCA5	The controller must not exceed the calculation time specified by the programmer	If the controller exceeds a certain time predefined during the planning, it is

			necessary to stop manually the operation of the controller
Avoid collision	UCA 7/ UCA 10	The controller should not let the robot gets in contact with any of other dynamic or static obstacles (Objects, Humans, robots) during its navigation.	The robot shall respect an adequate minimum safety distance MSD (not less than 0.5 m) between robots and any other obstacles during its navigation to avoid any contact. It must not be violated under any condition.
	UCA 8	The controller should not specify the wrong command to avoid collision	The controller shall check all the directions around the robot before specifying the correct avoidance command
	UCA 9	The collision avoidance controller must not let the robot get in touch with any static or dynamic obstacles especially human or a robot carrying a product	The robot shall respect a specified safety distances between other static or dynamic obstacles to avoid any collision and three layers of safety distances (zones) surrounding the robot should be made. If there is an obstacle detected on the third one, the controller begin to reduce the speed of the robot and if it isn't detected until the first zone, the controller must stop the motion immediately
	UCA 10	The collision avoidance feature must operating all the navigation time	A contact sensor must be in working order to stop the robot, in case of a contact or a collision.
Establish and maintain communication between robots	UCA 14	The robots should communicate and negotiate with each other before doing action.	Tasks, motion and actions of robots must be organized between them by ensuring a continuous exchange of information inter-robot using messages to avoid entanglements and unwanted action in order to ensure an effective cooperation
	UCA 15/ UCA 16	Sufficient information (to be exchanged) provided by communication should be specified	the type of information should be specified (robots positions, velocities, type of tasks, information about the rooms state if they are occupied or empty)
	UCA 17	Communication should be maintained	It is necessary to use a multilayer software like ROS which supports the huge number of information diffusion and facilitates its transmission and if there is a loose of connection between the nodes it is necessary to indicate to the master
Track the path	UCA 20/ UCA 23	Robots should not deviate from the specified path during their navigation	- The controller shall detect any deviation from the specified path using multi-sensors which detects the robots position (odometer, laser telemeter...)
	UCA 21	The precision should be considered in the tracking feature	- Use sensors redundancy (multi-sensors) to minimize the risk of failure - It is necessary to choose the good filter for sensors information to increase the precision of tracking - The risk of sensor defect must be minimized by the use of the fault tolerant approach - To minimize the risk of non-observability, an estimator must be used
Send the velocity command	UCA 23, 24, 25, 26, 27, 28, 29	The velocity command must be always available	- The programmer shall specify multi-level of safety distances: SD 1 for humans / SD 2 for robots / SD 3 for other objects - Detect slippery (using Camera) and reduce the speed of the robot in that case - Detect the load using sensor for weight - Speed down the robot when it is carrying products or in front of human, or other robots

- **Step 6**:modeling and quantfyingcollisionhazard using SPN

This section presents the integration part of the SPN model for modeling and assessing collision scenarios of two mobile robots on the basis of their normal operating scenario. The operating state of these multi-robot is decribed using petri net modeling by illustrating coordination between their tasks.

1) Normal operation modeling of robots with SPN:

Figure 8 illustrates a model represented by SPN, drawn using the GRIF software (Grif, 2020). This model describes the normal operating scenario of two robots (R1 and R2) in an analysis laboratory. The laboratory consists mainly of an analysis room, two chemical storage rooms, and a battery charging room (see figure 3).

Part 1, shown in figure 8, presents the phase of sending orders and preparing the robots for an operation. The two robots represented by two places (P2 and P3), the two markings indicate that the robots are ready, and that they are waiting for the command issued by the person in the analysis room (Pa free) to move. The "demands_ S1" and "demands_ S2" transitions represent the product demands from storage room 1 (roorm1) and storage room 2 (room 2) respectively. The order is made according to the needs of the analyzes (every 4 hours a demand is launched for each room). One list for product requests for storage room 1 and another for storage room 2. Places P5 andP6 indicate the number of demands!D1,!D2 respectively. Once a demand is initiated, the command will be sent to the robots depending on the availability of each of the two robots (!Robot 1 = true or! Robot 2 = true). If both robots are available, there will be a priority for robot 1. Two counters are provided to count the number of cycles per robot (Nc1 = Nc1 + 1 and Nc2 = Nc2 + 1). If there are no robots in room 1, the value of the room1 variable is false (! Room1 = false). Once one of the robots enters room 1, the room 1 variable will return true and the same with room 2.

To know which robot enters each room, 4 variables are defined as follows:

- (! R11 = true) if robot 1 goes to room 1;

- (! R21 = true) if robot 2 goes to room 1;

- (! R22 = true) if robot 2 goes to room 2;

- (! R12 = true) if robot 1 goes to room 2.

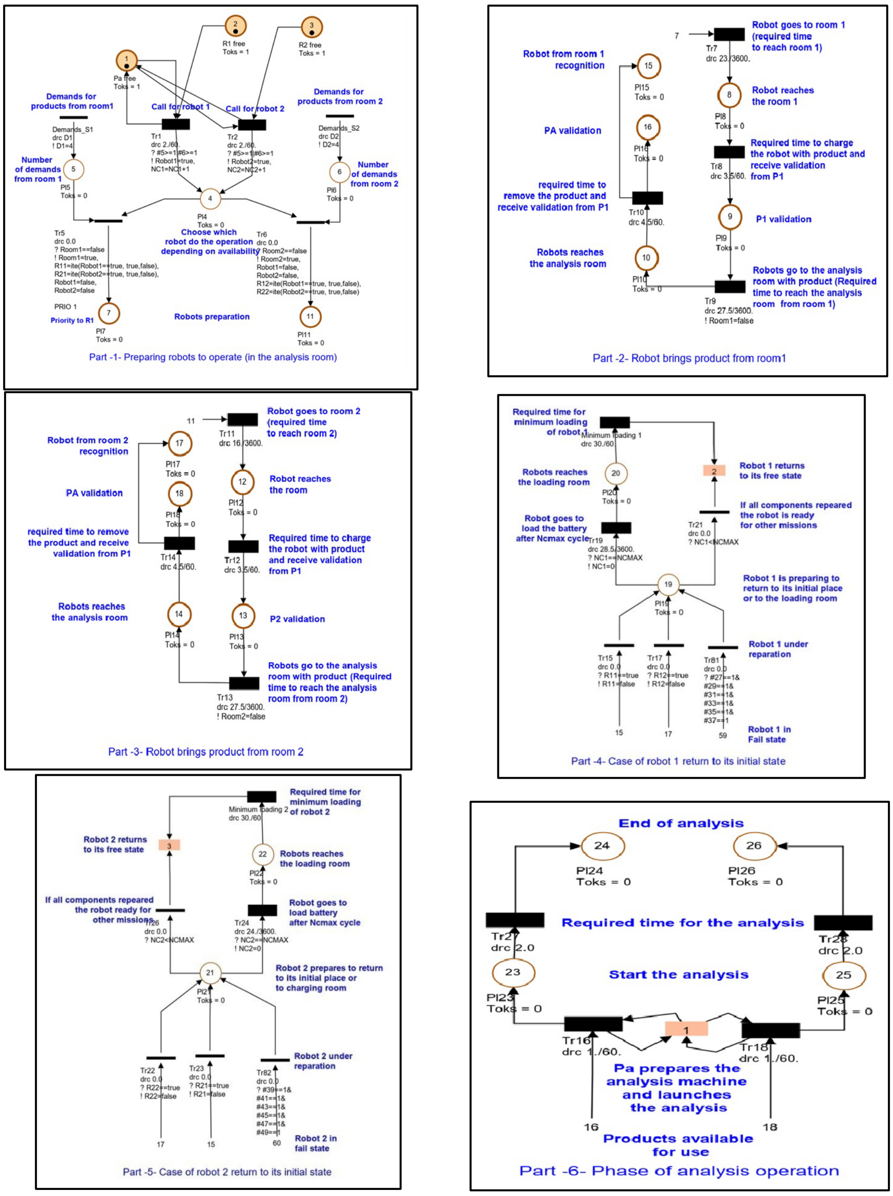

Part -1- Preparing robots to operate (in the analysis room)
Demands for products from room1
Demands for products from room 2
Call for robot 1
Call for robot 2
Number of demands from room 1
Number of demands from room 2
Choose which robot do the operation depending on availability
Priority to R1
Robots preparation
Part -2- Robot brings product from room1
Robot goes to room 1 (required time to reach room 1)
Robot from room 1 recognition
Robot reaches the room 1
PA validation
Required time to charge the robot with product and receive validation from P1
required time to remove the product and receive validation from P1
P1 validation
Robots reaches the analysis room
Robots go to the analysis room with product (Required time to reach the analysis room from room 1)
Part -3- Robot brings product from room 2
Robot goes to room 2 (required time to reach room 2)
Robot from room 2 recognition
Robot reaches the room
PA validation
Required time to charge the robot with product and receive validation from P1
required time to remove the product and receive validation from P1
Robots reaches the analysis room
P2 validation
Robots go to the analysis room with product (Required time to reach the analysis room from room 2)
Part -4- Case of robot 1 return to its initial state
Required time for minimum loading of robot 1
Robots reaches the loading room
Robot 1 returns to its free state
If all components repeared the robot is ready for other missions
Robot goes to load the battery after Ncmax cycle
Robot 1 is preparing to return to its initial place or to the loading room
Robot 1 under reparation
Robot 1 in Fail state
Part -5- Case of robot 2 return to its initial state
Required time for minimum loading of robot 2
Robot 2 returns to its free state
Robots reaches the loading room
If all components repeared the robot ready for other missions
Robot goes to load battery after Ncmax cycle
Robot 2 prepares to return to its initial place or to charging room
Robot 2 under reparation
Robot 2 in fail state
Part -6- Phase of analysis operation
End of analysis
Required time for the analysis
Start the analysis
Pa prepares the analysis machine and launches the analysis
Products available for use

Fig. 8.Normal operating model of the two robots.

Parts 2 and 3, shown in figure 8, present the transportation of products with robots. The robots move from the analysis room to storage rooms 1 and 2 respectively. Persons in rooms 1 and 2 (P1 and P2 respectively) place the products on the robots. Once the robots leave room 1 or room 2, the room1 or room2 variables return to their free states (! Room 1 = false,! Room 2 = false), i.e. there is no robot in storage rooms. The robots return to the analysis room with products. The transitions (Tr7, Tr8, Tr9, Tr10, Tr11, Tr12, Tr13, Tr14) are defined by a dirac law with deterministic delays, which represent transportation periods for the two robots.

Then, in parts 4 and 5, shown in figure 8, a recognition of robots (the two robots define by two tokens) is performed to identify them and each robot can return to its initial place. If robots makes a number of turns (Nc) greater than Ncmax, they go to the battery loading room before being available again and the Nc counter returns to the value 0.

Finally, part 6, shown in figure 8, represents the phase of preparing machines by the person in the analysis room (Pa) to perform the analysis.

2) *Modeling of collision scenarios*

Figure 9 illustrates the SPN model for the main loss scenarios that lead to a collision state. We identified the six main causes from the results obtained by the STPA method (failure of the motor of the left wheel or that of the right wheel, failure of the laser scanner sensor, failure of the communication system, failure of the control card and battery problem). Note that the failure of the two motors of each robot can occur independently or be caused by a common cause failure (CCF) event.

Based on STPA results, the SPN model is used to evaluate the collision frequency according to the characteristics determined by the studied system.

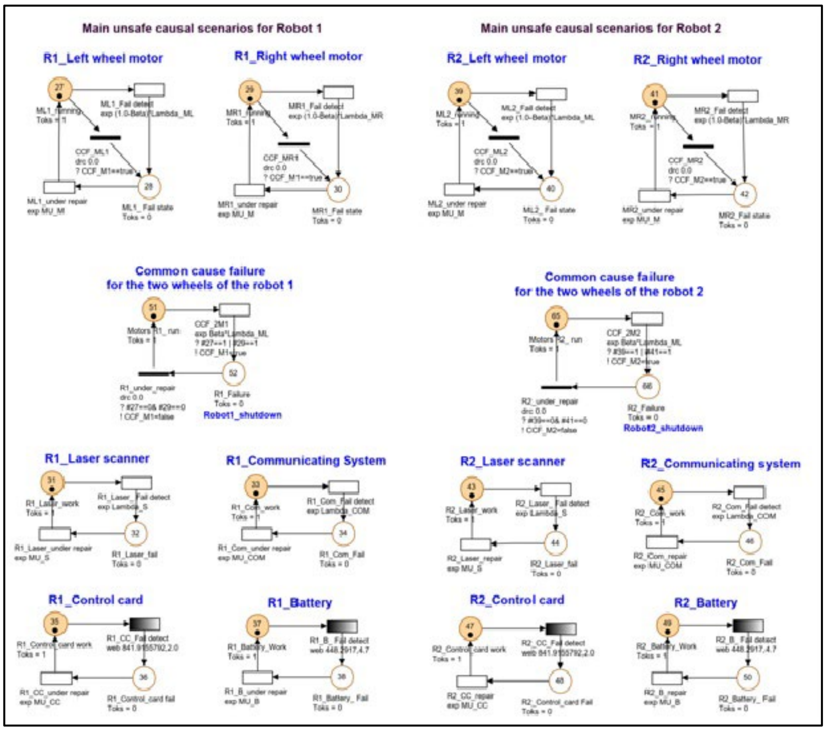


Fig. 9.Main causal scenarios leading to collision for the two robots.

The tokens initially remain in places (P27, P29, P31, P33, P35, P37) for robot 1, and in places (P39, P41, P43, P45, P47, P51) for robot 2, indicating the good functioning of the two robots and all their components. In addition, the tokens in P55 and P57 indicating the safety state of the two robots, which means that there is no collision (see figure 10).

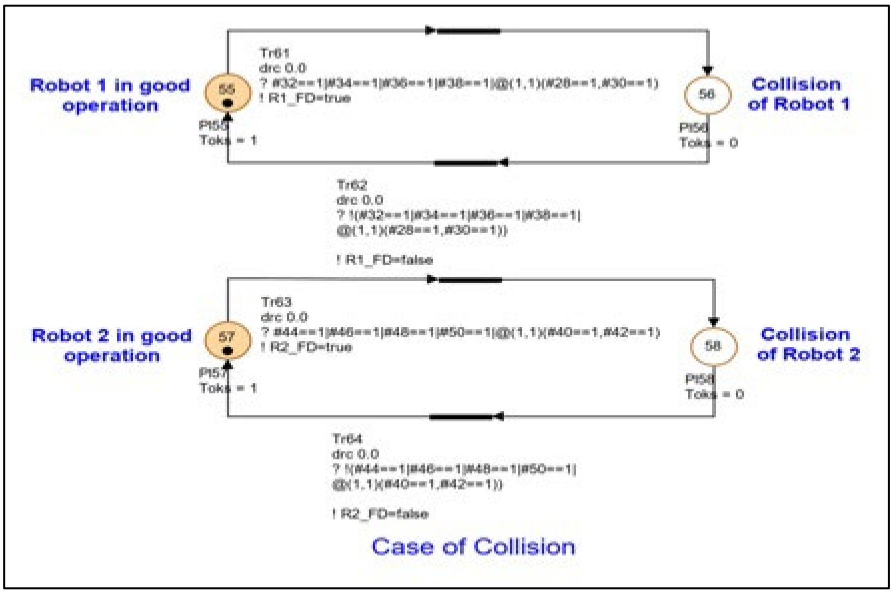


Fig. 10. Collision scenarios.

When tokens fire failure detection transitions and reach places (P28, P30, P32, P34, P36, P38) for robot 1 or places (P40, P42, P44, P46, P48, P50) for robot 2, it means that robots enter in an unsafe state,which is caused by a failure of one or more of its components. Those failures are the main factors of collision event (the tokens fire the Tr 61 or Tr 63 transitions according to the predicate " ? # 32 == 1 | # 34 == 1 | # 36 == 1 | # 38 == 1 | @ (1,1) (# 28 == 1, # 30 == 1) ”or“? # 44 == 1 | # 46 == 1 | # 48 == 1 | # 50 == 1 | @ (1,1) (# 40 == 1, # 42 == 1) ”, where the palces 28, 30, 32, 34, 36, 38 represent the failure state of the components of the first robot and places 40, 42, 44, 46, 48, 50 represent the failure state of the components of the second robot, and reach places P56 or P58 from P55 or P57, which realizes the assertion "! R1_FD = true " if robot 1 fails to operate or"! R2_FD = true " if robot 2 fails to operate.

The defective robots will be transferred from places (P7, P8, P9, P10, P11, P12, P13, P14) to the maintenance service (P59 or P60), shown in figure 11, and the tokens remain in these places until the repair of the problem is completed. When the robots return to their initial states through the transitions Tr62 and Tr64, the variables “R1_FD and R2_FD” will be assigned as false, the transitions Tr81 and Tr82 will be crossed to reach the places P19 and P21 and the robots continue their missions.

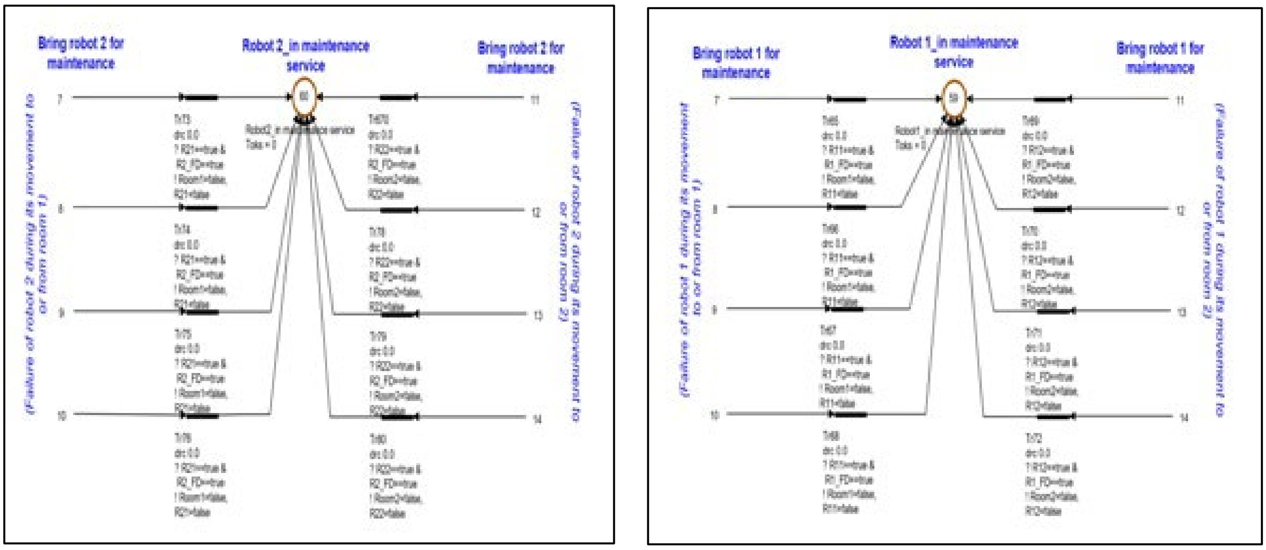

Bring robot 2 for maintenance
Robot 2_in maintenance service
Bring robot 2 for maintenance
(Failure of robot 2 during its movement to or from room 1)
(Failure of robot 2 during its movement to or from room 2)
7
8
9
10
11
12
13
14
60
Bring robot 1 for maintenance
Robot 1_in maintenance service
Bring robot 1 for maintenance
(Failure of robot 1 during its movement to or from room 1)
(Failure of robot 1 during its movement to or from room 2)
59

Fig. 11. Model of robots in maintenance service.

3) *Simulation results*

Failure probability data retrieved from references (IEC 61508:2010 ; Fazlollahtabar and Niaki, 2017)are re-evaluated based on new operating conditions. The table 4 below shows the estimated failure data of robots components.

Table 4. Failure probability data.

Components	Probability distribution
Electric motor	Exp (9×10^{-6})
Laser scanner	Exp (5×10^{-6})
Communicating system	Exp (9×10^{-4})
Control card	Web $(2, 9.5\times 10^{4})$
Battery	Web $(4.7, 4.9\times 10^{4})$

Failure events are assumed to be distributed according to two probability laws: the exponential law and the Weibull law. The duration of the experiment for the simulation is four years (34,560 hours), the number of histories is 10^{6}history, and the confidence interval is estimated at 90%. Figures 12, 13 and14below illustrate The instant unavailability of the two robots, the mean time to collision for the two robots and the frequency of collision respectively for an experiment period of 4 years with the two lower and upper confidence interval (CI) bound.

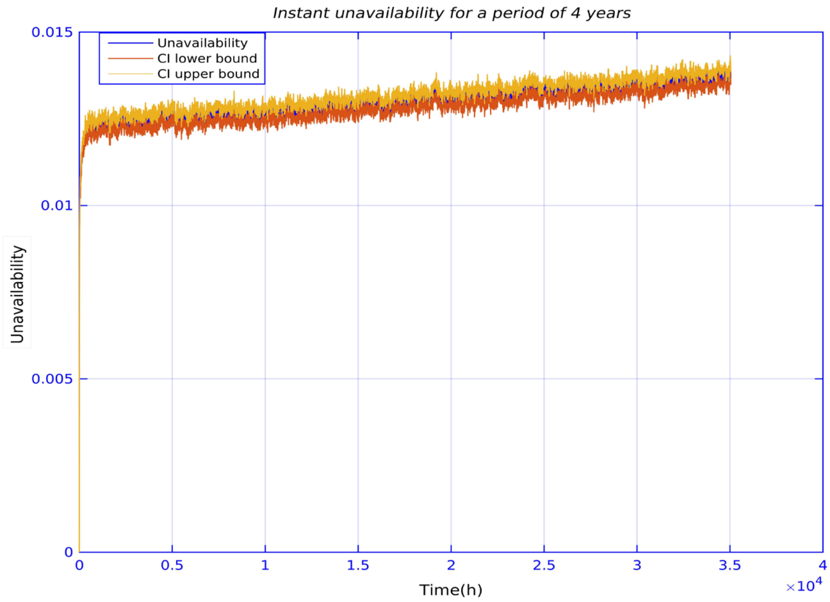


Fig. 12. Instant unavailability of the two robots.

The robots unavailability, the mean time to collision and the collision frequency were directly calculated by Monte-Carlo simulation using Grif software (Grif, 2020).

As shown in figures 12 and 13, the unavailabilty and the mean time of collision of the two robots are the 0.014 and 542 hours respectively. Which means that the fisrt collision happenes after 542 hours of the two robots continuous operation.

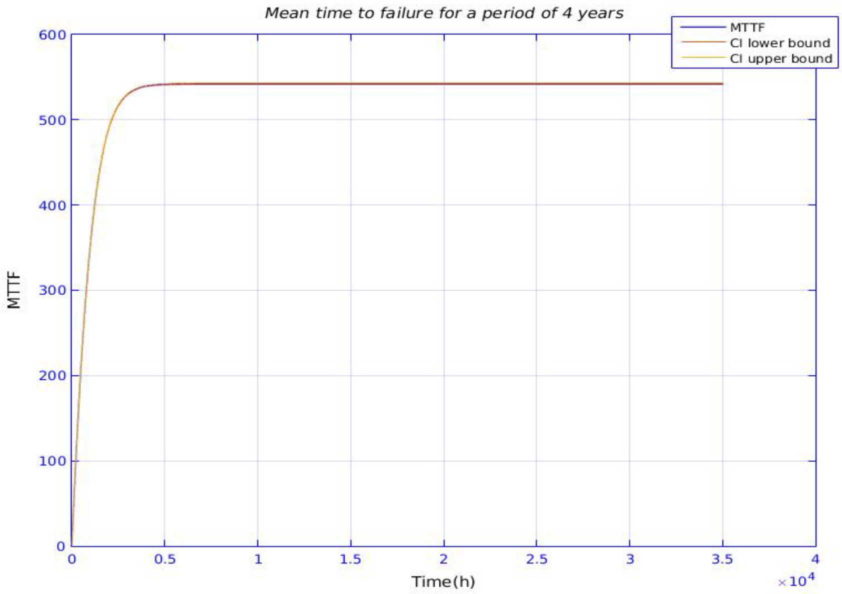


Fig. 13. Mean time to collision of the two robots.

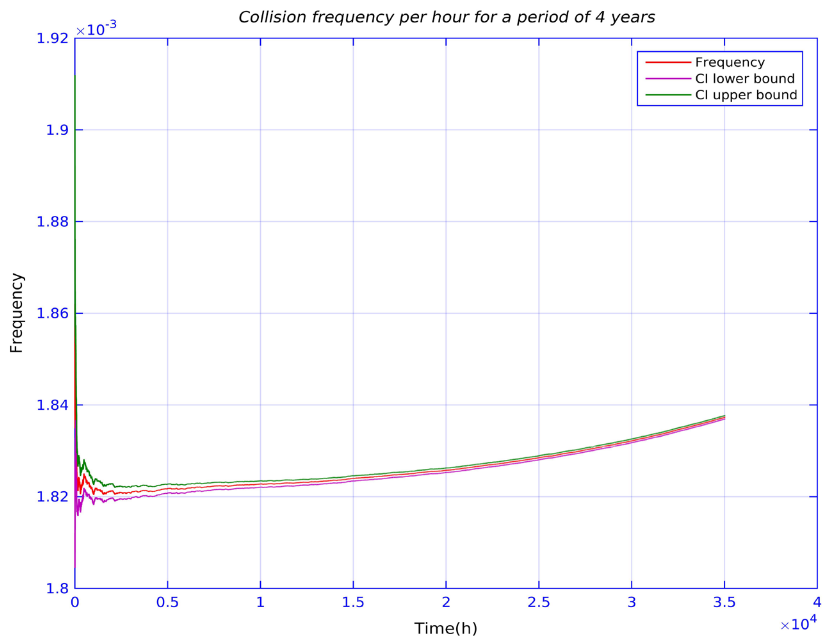


Fig. 14. Collision frequency related to the two robots.

According to the graph infigure 14, we observe that the value of frequency is between $1.82\ \times 10^{-3}h^{-1}and1.86 \times 10^{-3}h^{-1}$.Itmeans that every 22 to 23 days there is a risk that one of the two robots will collide in the lab.

It should be noted that the degradation of performance begins after 23 days of continuous operation, and therefore the collision frequency is considerable.

The critical parts of the system represented in comunicating systems and batteries.

4. Discussion

The approach used in this article is a combination between the STPA method and the Stochastic Petri Net (SPN) model. This approach provides a detailed analysis of the various functionalities of autonomous mobile robot controllers using more detailed control architecture by STPA analysis. The result of this analysis identifies the necessary safety constraints and also proposes requirements to optimize the control of these robots in complex high-risk environments. In addition, the SPN helps us to model the behavior of robots during their normal operation and also to model hazard scenarios as well as to quantitatively assess the risk of collision.

The results reveal that the main problems that may hinder the safe operation of robots are the problems of control, communication, power and retrieving environment information from sensors.

The most marked property of STPA analysis is the ability to extract many sets of hazardous events, including those caused by failures of system elements, such as risks resulting from unintentional interactions between elements. In addition, the STPA analysis provides a large number of potential scenarios, which does not only deal with the system in operating modes / malfunctions but also takes into account the time factor and its influence (action executed too late or too early).

Among limitations that we faced are:

The complexity of the whole behavior modeling for a multi-robots system using SPN, in particular, when dealing with a large number of robots (more than two). Increasing the number of robots makes a large scale model, which causes a problem of readability, understandability and evaluability.

In our study, we select only the collision scenarios caused by robot components failure for the numerical experiment, while ignoring other factors such as human errors, degraded component operation...

5. Conclusion

In this article, we investigate how risk and safety of autonomous multi-robots systems should be managed during a chemical transportation task in high risk environment using the combination STPA and SPN, The STPA method is applied to identify and analyze unsafe robot behavior, also to generate the needed constraints and requirements to ensure the safe operation of these robots in such environment. Whereas the SPN is used mainly to complement the STPA in order to create a framework for modeling and quantifying the collision hazard.

The proposed methodology allowed us to provide a detailed analysis of complex industrial systems equipped with autonomous mobile robots, to also collect a large number of potential risk scenarios, to model the behavior of the system during its normal operation and its dysfunctional and unsafe behavior. Further, to assess the robots performance and the occurrence frequency of collision hazard. Thus, safety requirements and recommendations have been suggested to resolve the navigation and control problems of differential wheel mobile robots, while maintaining the safety and security of the entire laboratory. In our case study, we limited the number of robots to two robots because of the high complexity of the Petri net model. The goal is to facilitate the comprehensibility of the model and to minimize the simulation time. In addition, we considered the collision scenarios caused by robot component failures, while ignoring other factors, such as human errors, degraded component operation ... In future work; to have more rigorous quantification of loss scenarios it is preferable to take into account each unsafe control action.

6. References

Abdulkhaleq, A. *et al.* (2018) 'Missing no Interaction—Using STPA for Identifying Hazardous Interactions of Automated Driving Systems', *International Journal of Safety Science*, 02(01), pp. 115–124. https://doi.org/ 10.24900/ijss/0201115124.2018.0301.

Alemzadeh, H. *et al.* (2014) 'Systems-theoretic safety assessment of robotic telesurgical systems', in *International conference on computer safety, reliability, and security*. Springer, pp. 213–227.

Alexander, R., Herbert, N. and Kelly, T. (2009) 'Deriving safety requirements for autonomous systems', in *4th SEAS DTC Technical Conference*.

Andreadakis, S. K. and Levis, A. H. (1988) *Synthesis of distributed command and control for the outer air battle*. Massachusetts inst of tech cambridge lab for information and decision systems.

Banda, O. A. V. and Kannos, S. (2017) 'Hazard Analysis Process for Autonomous Vessels', p. 69.

Bensaci, C. *et al.* (2020) 'STPA and Bowtie risk analysis study for centralized and hierarchical control architectures comparison', *Alexandria Engineering Journal*. Elsevier, 59(5), pp. 3799–3816.

Böhm, P. and Gruber, T. (2010) 'A novel HAZOP study approach in the RAMS analysis of a therapeutic robot for disabled children', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6351 LNCS, pp. 15–27. https://doi.org/10.1007/978-3-642-15651-9_2.

Dogramadzi, S. *et al.* (2014) 'Environmental Hazard Analysis - a Variant of Preliminary Hazard Analysis for Autonomous Mobile Robots', *Journal of Intelligent and Robotic Systems: Theory and Applications*, 76(1), pp. 73–117. https://doi.org/10.1007/s10846-013-0020-7.

Dutuit, Y. *et al.* (1997) 'Dependability modelling and evaluation by using stochastic Petri nets: application to two test cases', *Reliability Engineering & System Safety*. Elsevier, 55(2), pp. 117–124.

Fan, T. *et al.* (2018) 'Fully distributed multi-robot collision avoidance via deep reinforcement learning for safe and efficient navigation in complex scenarios', *arXiv preprint arXiv:1808.03841*.

Fazlollahtabar, H. and Niaki, S. T. A. (2017) 'Integration of fault tree analysis, reliability block diagram and hazard decision tree for industrial robot reliability evaluation', *Industrial Robot: An International Journal*. Emerald Publishing Limited.

Fleming, C. H. *et al.* (2013) 'Safety assurance in NextGen and complex transportation systems', *Safety science*. Elsevier, 55, pp. 173–187.

Grif (2020) GRIF-Workshop, "Graphical interface for reliability forecasting software",2020. Available at: http://grif-workshop.fr/

Guiochet, J. (2016) 'Hazard analysis of human–robot interactions with HAZOP–UML', *Safety science*. Elsevier, 84, pp. 225–237.

Hardy, K. and Guarnieri, F. (2011) 'Using a systemic model of accident for improving innovative technologies: application and limitations of the STAMP model to a process for treatment of contaminated substances', in *The 15th World Multi-Conference on*

Systemics, Cybernetics and Informatics: WMSCI.

IEC 61508standard (2010) ‘Functional safety of electrical/ electronic/ programmable electronic safety-related systems- Parts 1 to 7’, International Electrotechnical Commission, Geneva, Switzerland.

Ishimatsu, T. *et al.* (2010) ‘Modeling and hazard analysis using STPA’. International Association for the Advancement of Space Safety (IAASS).

Ishimatsu, T. *et al.* (2014) ‘Hazard analysis of complex spacecraft using systems-theoretic process analysis’, *Journal of Spacecraft and Rockets*, 51(2), pp. 509–522. https://doi.org/10.2514/1.A32449.

Jian, S., Shaoping, W. and Yaoxing, S. (2008) ‘Petri-nets based availability model of fault-tolerant server system’, in *2008 IEEE conference on robotics, automation and mechatronics*. IEEE, pp. 444–449.

Jin, L. *et al.* (2019) ‘Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach’, *Automatica*. Elsevier Ltd, 100, pp. 75–81. https://doi.org/10.1016/j.automatica.2018.11.001.

Kazanzides, P. (2009) ‘Safety Design for medical robots’, *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, pp. 7208–7211. https://doi.org/10.1109/IEMBS.2009.5335275.

Khan, S., Madnick, S. and Moulton, A. (2018) ‘Cybersafety Analysis of a Central Utilities Plant (CUP) Gas Turbine using STPA-SEC’. MIT Sloan Research Paper. http://doi.org/10.2139/ssrn.3370560.

Kim, H. *et al.* (2018) ‘Application of system-theoretic process analysis to the isolation of subsea wells: Opportunities and challenges of applying STPA to subsea operations’, *Proceedings of the Annual Offshore Technology Conference*, 6, pp. 4351–4361.

Kumar, G., Jain, V. and Gandhi, O. P. (2012) ‘Reliability and availability analysis of mechanical systems using stochastic petri net modeling based on decomposition approach’, *International Journal of Reliability, Quality and Safety Engineering*. World Scientific, 19(01), p. 1250005.

Kumar, V. and Aggarwal, K. K. (1993) ‘Petri net modelling and reliability evaluation of distributed processing systems’, *Reliability Engineering & System Safety*. Elsevier, 41(2), pp. 167–176.

La, N.-T. and Kwon, G. (2018) ‘Risk Assessment for STPA with FMEA Technique’, in *International Conference on Frontier Computing*. Springer, pp. 444–455.

Lasota, P. A., Fong, T. and Shah, J. A. (2017) ‘A survey of methods for safe human-robot interaction’, *Foundations and Trends® in Robotics*. Now Publishers, Inc., 5(4), pp. 261–349.

Lee, D.-A. *et al.* (2013) ‘Application of system-theoretic process analysis to engineered safety features-component control system’, in *Proc. of the 37th Enlarged Halden Programme Group (EHPG) meeting.*

Lee, S. and Yam, Y. (2012) ‘Risk Assessment and Functional Safety Analysis to Design Safety Function of a Human-Cooperative Robot’, *Human Machine Interaction - Getting Closer*. doi: 10.5772/37821.

Leveson, N. (2004) ‘A new accident model for engineering safer systems’, *Safety science*. Elsevier, 42(4), pp. 237–270.

Leveson, N. (2011) *Engineering a safer world: Systems thinking applied to safety*. MIT press.ISBN 9780262016629, Cambridge, p.560.

Leveson, N. and Thomas, J. (2018) *STPA handbook*, Online document, vol. 3, p.188 .

Li, H. and Savkin, A. V (2018) ‘An algorithm for safe navigation of mobile robots by a sensor network in dynamic cluttered industrial environments’, *Robotics and Computer-Integrated Manufacturing*. Elsevier, 54, pp. 65–82.

Li, S. *et al.* (2012) ‘Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks’, *Neurocomputing*. Elsevier, 91, pp. 1–10.

Li, S., Kong, R. and Guo, Y. (2014) ‘Cooperative distributed source seeking by multiple robots: Algorithms and experiments’, *IEEE/ASME Transactions on mechatronics*. IEEE, 19(6), pp. 1810–1820.

Liu, C. (2017) ‘Safe Robot Navigation Among Moving and Steady Obstacles [Bookshelf]’, *IEEE Control Systems Magazine*. IEEE, 37(1), pp. 123–125.

Liu, S. B. *et al.* (2017) ‘Provably safe motion of mobile robots in human environments’, *IEEE International Conference on Intelligent Robots and Systems*, 2017-Septe, pp. 1351–1357. https://doi.org/10.1109/IROS.2017.8202313.

Machin, M. (2015) ‘Synthèse de règles de sécurité pour des systèmes autonomes critiques’. Université de Toulouse, Université Toulouse III-Paul Sabatier.

Malhotra, M. and Trivedi, K. S. (1995) ‘Dependability Modeling Using Petri-Nets’, *IEEE Transactions on Reliability*, 44(3), pp. 428–440. https://doi.org/10.1109/24.406578.

Martin-Guillerez, D. *et al.*(2010) ‘A UML-based method for risk analysis of human-robot interactions’, in *Proceedings of the 2nd International Workshop on Software Engineering for Resilient Systems*. ACM, pp. 32–41.

Mendiburu, F. J., Morais, M. R. A. and Lima, A. M. N. (2016) ‘Behavior coordination in multi-robot systems’, *2016 IEEE International Conference on Automatica, ICA-ACCA 2016*. https://doi.org/10.1109/ICA-ACCA.2016.7778506.

Milutinovic, D. and Lima, P. (2002) ‘Petri net models of robotic tasks’, *Proceedings - IEEE International Conference on Robotics and Automation*, 4(February 2002), pp. 4059–4064. https://doi.org/10.1109/robot.2002.1014376.

Nakao, H. *et al.* (2011) ‘Safety guided design of crew return vehicle in concept design phase using STAMP/STPA’, in *Fifth International Association for the Advancement of Space Safety Conference, Versailles, France, Oct*. Citeseer, pp. 17–19.

Pandey, A. and Parhi, D. R. (2016) ‘Multiple mobile robots navigation and obstacle avoidance using minimum rule based ANFIS network controller in the cluttered environment’, *International Journal of Advanced Robotics and Automation*, 1(1), pp. 1–11.

Plioutsias, A. and Karanikas, N. (2015) ‘Using STPA in the Evaluation of Fighter Pilots Training Programs’, *Procedia Engineering*. Elsevier B.V., 128, pp. 25–34. https://doi.org/10.1016/j.proeng.2015.11.501.

Rachman, A. and Ratnayake, R. M. C. (2015) ‘Implementation of system-based hazard Analysis on physical safety barrier: A case study in subsea HIPPS’, in *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, pp. 11–15.

Rodríguez, M. and Díaz, I. (2016) ‘A systematic and integral hazards analysis technique applied to the process industry’, *Journal of Loss Prevention in the Process Industries*. Elsevier, 43, pp. 721–729.

Rokseth, B., Haugen, O. I. and Utne, I. B. (2019) ‘Safety Verification for Autonomous Ships’, *MATEC Web of Conferences*, 273, p. 02002. https://doi.org/10.1051/matecconf/201927302002.

Saenz, J. *et al.* (2018) ‘Survey of methods for design of collaborative robotics applications-Why safety is a barrier to more widespread robotics uptake’, in *Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering*. ACM, pp. 95–101.

Scarinci, A. *et al.* (2019) ‘Requirement generation for highly integrated aircraft systems through STPA: An application’, *Journal of Aerospace Information Systems*, 16(1), pp. 9–21. https://doi.org/10.2514/1.I010602.

Signoret, J.-P. (2008) ‘Analyse des risques des systèmes dynamiques: réseaux de Petri-Exemples de modélisation’,*Sécurité et gestion des risques*, *Techniques de l’ingénieur.* Ref: SE4072 v1.

Signoret, J. P. (2009) ‘Dependability & safety modeling and calculation: Petri nets’, *IFAC Proceedings Volumes (IFAC-PapersOnline)*. IFAC, 2(PART 1), pp. 203–208. https://doi.org/10.3182/20090610-3-IT-4004.00040.

Signoret, J. P. *et al.*(2013) ‘Make your Petri nets understandable: Reliability block diagrams driven Petri nets’, *Reliability Engineering and System Safety*, 113(1), pp. 61–75. https://doi.org/10.1016/j.ress.2012.12.008.

Song, Y. (2012) ‘Applying system-theoretic accident model and processes (STAMP) to hazard analysis’, PhD Thesis, MC Master university, Canada. 2012.

Szatmary, B. and Richert, M. (2017) ‘Apparatus and methods for safe navigation of robotic devices’. U.S. Patent No 9,840,003, Washington.

Tang, S., Thomas, J. and Kumar, V. (2016) ‘Safe navigation of quadrotor teams to labeled goals in limited workspaces’, in *International Symposium on Experimental Robotics*. Springer, pp. 586–598.

Thomas, J., Lemos, F. and Leveson, N. (2012) ‘Evaluating the safety of digital instrumentation and control systems in nuclear power plants’, *NRC Technical Researcy Report2013*.

Uesako, D. (2016) ‘STAMP applied to Fukushima Daiichi nuclear disaster and the safety of nuclear power plants in Japan’. Massachusetts Institute of Technology.

Wang, J. (2007) ‘Charging information collection modeling and analysis of GPRS networks’, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. IEEE, 37(4), pp. 473–481.

Wang, J. (2012) ‘Chapter 15, Petri nets’, in : *Handbook of finite state based models and applications (1 st ed.)*. Computer Science, Mathematics & Statistics, Chapman and Hall/CRC, 409 pages, 2012. https://doi.org/10.1201/b13055.

Woodman, R. *et al.* (2012) ‘Building safer robots: Safety driven control’, *International Journal of Robotics Research*, 31(13), pp. 1603–1626. https://doi.org/10.1177/0278364912459665.

Wróbel, K., Montewka, J. and Kujala, P. (2018) ‘Towards the development of a system-theoretic model for safety assessment of autonomous merchant vessels’, *Reliability Engineering and System Safety*, 178, pp. 209–224. https://doi.org/10.1016/j.ress.2018.05.019.

Yang, X. *et al.* (2011) ‘Petri net model and reliability evaluation for wind turbine hydraulic variable pitch systems’, *Energies*. Molecular Diversity Preservation International, 4(6), pp. 978–997.

Yoo, S. J. and Kim, T.-H. (2015) ‘Distributed formation tracking of networked mobile robots under unknown slippage effects’, *Automatica*. Elsevier, 54, pp. 100–106.

Zhang, J. *et al.* (2019a) ‘Combining system-theoretic process analysis and availability assessment: A subsea case study’, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. SAGE Publications Sage UK: London, England, 233(4), pp. 520–536.

Zhang, J. *et al.*(2019b) ‘Zhang, J., Kim, H., Liu, Y., & Lundteigen, M. A. (2019). Combining system-theoretic process analysis and availability assessment: A subsea case study. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability. https:/’, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. https://doi.org/10.1177/1748006X18822224.