applied sciences

MDPI

Article

Decision Making with STPA through Markov Decision Process, a Theoretic Framework for Safe Human-Robot Collaboration

Angeliki Zacharaki [1,*], Ioannis Kostavelis [2] and Ioannis Dokas [1]

[1] Department of Civil Engineering, Democritus University of Thrace, 671 00 Komotini, Greece; idokas@civil.duth.gr
[2] Department of Production and Management Engineering, Democritus University of Thrace, 671 00 Komotini, Greece; gkostave@pme.duth.gr
* Correspondence: azachara@civil.duth.gr

Abstract: During the last decades, collaborative robots capable of operating out of their cages are widely used in industry to assist humans in mundane and harsh manufacturing tasks. Although such robots are inherently safe by design, they are commonly accompanied by external sensors and other cyber-physical systems, to facilitate close cooperation with humans, which frequently render the collaborative ecosystem unsafe and prone to hazards. We introduce a method that capitalizes on partially observable Markov decision processes (POMDP) to amalgamate nominal actions of the system along with unsafe control actions posed by the System Theoretic Process Analysis (STPA). A decision-making mechanism that constantly prompts the system into a safer state is realized by providing situation awareness about the safety levels of the collaborative ecosystem by associating the system safety awareness with specific groups of selected actions. POMDP compensates the partial observability and uncertainty of the current state of the collaborative environment and creates safety screening policies that tend to make decisions that balance the system from unsafe to safe states in real time during the operational phase. The theoretical framework is assessed on a simulated human–robot collaborative scenario and proved capable of identifying loss and success scenarios.

Keywords: safe collaborative robots; partially observable markov decision processes; system theoretic process analysis; safety awareness levels; prompting system

Citation: Zacharaki, A.; Kostavelis, I.; Dokas, I. Decision Making with STPA through Markov Decision Process, a Theoretic Framework for Safe Human-Robot Collaboration. *Appl. Sci.* **2021**, *11*, 5212. https://doi.org/10.3390/app11115212

Academic Editor: Anton Civit

Received: 19 May 2021
Accepted: 2 June 2021
Published: 4 June 2021

1. Introduction

The immense need of contemporary industries to meet the technological requirements of the factories of the future as imposed by Industry 4.0 brought significant technological breakthroughs in the robotics domain, leading the robots out of their cages to work in close collaboration with humans, aiming to increase productivity, flexibility and autonomy in production [1]. Although the fact that such collaborative robots are designed and built with inherent safety mechanisms [2], their deployment in unconstrained environments typically necessitates their integration with other technological tools, such as visual and non-visual sensors along with custom algorithm solutions and other human machines interaction (HMI) interfaces, to realize true collaboration with human co-workers. The synergy of all these components formulate a collaborative ecosystem that, combined with the human interactive actions, could provoke hazardous events that conditionally render human–robot collaboration (HRC) unsafe [3].

This new type of collaborative environment imposed the in-depth re-consideration of safety mechanisms that take into account both the new technological advancements and the human factor, which holds an important role in Industry 4.0, to enable the workflow's execution as smoothly as possible [4]. Towards the design of such a system of human–robot collaboration, the discussions with experts and company workers is also needed [5,6]. On the one hand, the safety community brought into action specific International Organization for Standardization (ISO) standards [7,8], custom tailored to identify a series of important

hazards in HRC applications including potential sources of harm for the operator, their likely origins, and safety regulations for guiding the design and deployment of robotic solutions [9]. In particular, ISO standard 12100 [10] has been designed to identity potential hazards stemming from neglecting or intentional errors of the human operators; and to prevent their consequences, which are measured in terms of quantified risk values [11]. However, these standardized safety control mechanisms study the issues unilateral focusing either on the robots' features or on the human operator behavioral regulations and, thus, in some cases fail to model or predict hazardous events that stem from the true collaboration. On the other hand, research endeavors in safety analysis realized powerful tools that enable mainly hardware-related fault forecasting, which aims at estimating the cause–consequence chain of fault occurrence [12]. Such methods can be either *Bottom-Up* where a fault effect on the system is estimated in terms of cause–consequence, severity and probability, e.g., Failure Mode, Effects & Criticality Analysis (FMECA) [13], or *Top-Down*, where the determination of faults also induces identified unwanted effects, e.g., Fault Tree Analysis (FTA) [14], Hazard and Operability Analysis (HAZOP) [15]). Another recently introduced, yet powerful hazard analysis technique, namely System-Theoretic Process Analysis (STPA) [16], is based on the Systems Theoretic Accident Model and Process (STAMP) accident causation model and builds a model of the control structure of the system to identify control-related flaws. The main advantage of STPA, when compared to other hazard analysis methods, is that it forces a holistic study of the system to identify unsafe control actions that may lead to hazardous situations [17]. When it comes to HRC applications, with STPA, one can study simultaneously both the hardware as well as the human factor as a unified ecosystem that incorporates each other parameter and possible action related to any component of the examined system [18].

In this work, we introduce a novel method that studies the STPA as a decision making tool, based on a variation of the Markov belief propagation model, that associates un/safe control actions with un/safe states during collaborative tasks by incorporating the uncertainty of the observations obtained upon a selected action. We associate groups of states with safety-levels and map these states with context-dependent un/safe control actions, analyzed during the STPA analysis. We tackle the system's partial awareness regarding the collaborative ecosystem by modeling the finite number of expected observations with a probability density function. We prove that Partially Observable Markov Decision Processes (POMDPs) can act complementary to STPA analysis in order to create an active safety-related prompting tool that defines the loss scenarios during the system's operational phase. The system is prompted through a sequence of selected actions to transit into states of higher safety through a reward mechanism. The outcome POMDP modeling procedure is the formulation of a policy graph that can operate in real-time with the evolution of the collaborative task to continuously balance the system into safe states, as graphically illustrated in Figure 1. The main contributions of this work are summarized as follows:

- The introduction of the system's state and action space partitioning based on group of safe, unsafe and recovery states resulting from their association with the respective actions, that outline the system's current safety level.
- The interpretation of the STPA control structure as a decision making tool (based on POMDP models) to be utilized in the operational phase in order to decide on the most appropriate next action that will bring the system into a safer state.
- The application of STPA on a classic HRC topology and the evaluation of our decision making method with the formulated policy graph.

The rest of the paper is organized as follows: Section 2 discusses the relevant literature to the studied domain. The theoretical background and the problem formulation is presented in Section 3.1, while the proposed methodology is thoroughly analyzed in Section 4, where the STPA-related POMDP formulation is presented along with the prompting mechanisms on different safety levels. Section 5 appends a case study with the HRC example, interpreted with the safety prompting method, and conclusions are drawn in Section 6.

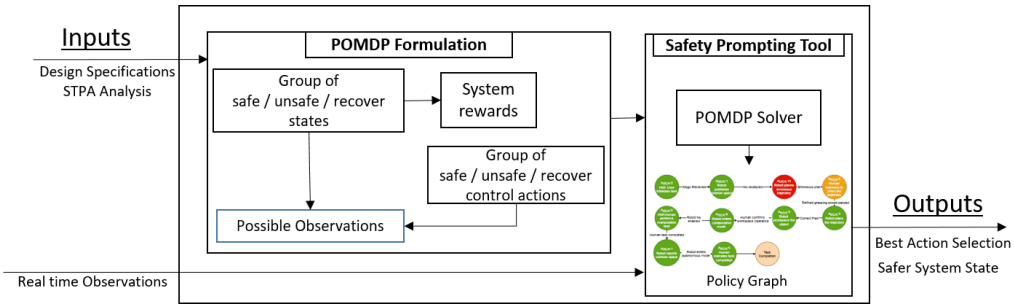


Figure 1. The flowchart of the proposed methodology.

2. Related Work

The systematic safety analysis and the systemic modeling of the hazardous situations in collaborative robots with humans has not yet been widely researched [19]. More focus has been applied on the identification of hazards on existing robotic systems to measure their dependability by employing system theory methods [11]. Yet, it has been proven that classic hazard identification approaches such as FTA and FMECA [13] are not well-suited for human–robot collaboration applications, since they cannot deal with unpredictable human–robot interactions (HRI). In another approach presented in [15], the authors systematized the pairing of HAZOP and Unified Modeling Language (UML), and demonstrated results also for collaborative scenarios, excluding the formal point of view and focusing on an informal solution, by gathering as much information as possible from brainstorming and meetings.

Authors in [18,20] studied the possibility of utilization of multi-robots with human workers in a laboratory environment by exploiting STPA and fault tree analysis to identify the potential safety hazard scenarios, their causal factors, and concluded with a set of recommendations. The approach proposed in [21] examined a case related to multi-robot systems considering the coordination, cooperation and collaboration aspects in accordance to which a risk identification study is performed by employing STPA to extract a set of risk scenarios related to different types of hierarchical coordination architectures in addition to their factors, and afterwards, the Bowtie method is utilized for the evaluation of the obtained scenarios. The method applied on a collaborative environment with multiple mobile robots and a great number of hazardous scenarios has been identified. From a different perspective, the authors in [22] introduced the SAFER-HRC methodology, which concerns an exhaustive exploration of all possible execution workflows of the application, the automatic identification of hazards within the workflows and the determination of their associated risk with the respective ISO standards, while the system also introduces provisions for risky hazards and verifies their effectiveness by computing residual risks. The method has been tested on an actual collaborative scenario and proved efficient in identifying hazards, yet the performance of it significantly depends on the accuracy of the model and the modeling of the probabilistic distribution of human's unintended behavior or errors.

As we study deeper, the research endeavors conducted for the systematic identification of hazards and risks, it becomes apparent that the identification of risks in real time, i.e., during the operation phase of the system, especially for HRC applications, is an imperative yet challenging task. Of such applications can be considered the current trend of automation and data exchange in manufacturing technologies of Industry 4.0 [23], as well as applications related to personal assistive robots [24], where also true and even physical interaction is required. In such applications, there is the need to predict the next accident more efficiently, yet the challenge of measuring what the actual safety level of a system is at a certain moment in time at a given context [25] is still under investigation, regardless of the introduction of new accident models and the view of safety as an emergent system property. The authors in [26] proposed a method to measure the gap between system design and operation as a metric for safety and introduced the Risk SituatiOn Awareness

Provision (RiskSOAP) indicator based on which one can compare the safety constraints of existing systems compared to their ideal set of safety constraints that derive from their STPA analysis, and calculate a situational awareness indicator for the system under study. In a more recent work presented in [27] a mathematical model for systems' safety level determination and its dynamic calculation based on the STAMP accident model has been introduced. The model employed the outcomes of STPA hazard analyses and transformed them into graph diagrams the nodes of which depicted an STPA based safety constraint, and each path of the acyclic diagram corresponded to a possible scenario of safety constraints violations that can lead to accidents.

Our work aims to provide such a real time tool, which capitalizes on the determined unsafe control actions extracted from the STPA, in order to prompt the system into the selection of an action, based on the current and past observations and actions, that will transit it to a next state that belongs to a higher safety level. We employ an abstraction for the clustering of the plethora of the normal and unsafe states of the system, and we formulate the problem as a partial observable Makrov decision process in order to determine a policy graph that operates in real time and deduces the safe and unsafe workflows of the system.

3. Theoretical Background and Problem Formulation

3.1. STPA Principals

STPA is one of the new hazard analysis techniques based on the STAMP causality model. In order to better understand the principles of STPA, an outline of the STAMP model basics is essential. In STAMP-related techniques, the systems are examined as a dynamic balanced system orchestrated by feedback and control loops [28]. The interaction between the subordinate components of the system and the human operator is modeled as a closed control schema obeying the classic control rule, where the controlled process is formulated through: (i) the received/sent control actions by the controller and (ii) the feedback received/observed by the controller upon the executed action. STPA is the hazard analysis technique of STAMP, where, based on the designed control structure, potentially hazardous control actions are identified. Upon the definition of purpose analysis of examined system, the basic steps of STPA are as follows [16]:

- Identify the losses considering all the stakeholders of the system;
- Identify system level hazards and link them to losses;
- Define system level constraints ;
- Model the general control structure;
- Identify unsafe control actions;
- Identify causes of unsafe control.

The outcome of this procedure is the definition of several system states that condition to the appearance of any tracked unsafe control action can lead to hazardous situations. This renders STPA an appropriate tool for the design phase of any system by contributing in the architecture definition, the creation of safety requirements, the identification of hazardous conditions and the planning of mitigation actions and other critical preparatory steps during the development of the system. Ergo, STPA contributes mainly in the design phase of a safe system. There are yet no reports in the literature describing how the safety specifications of STPA have been used to support decision-making during the operational phase of systems, since the aim of the method is to model unsafe scenarios to prevent hazardous situations. This may be related to the fact that STPA emphasizes on the states of the system that are linked to one or more unsafe control actions and does not take into consideration normal execution of control actions related to nominal system operation states. Control actions in STPA are typically analyzed in terms four different aspects in order to extract the respective UCAs, namely (i) *not providing causes hazard*, (ii) *providing causes hazard*, (iii) *providing too early, too late or out of order* and (iv) *stopped too soon or applied too long*.

3.2. STPA as Component of a Decision Making Tool

In our approach, we consider as unsafe control actions (UCAs) the actions that lead to a hazardous situation, and have been defined during the STPA analysis, while we consider as safe control actions (CAs) the nominal actions that occur when the system operation is in normal and safe conditions. We incorporate UCAs and resulting hazardous system states into the operational phase by holistically interpreting all the possible states to which the system can transit with respect to the executed control actions—either safe or unsafe ones. Thus, we handle the safety control structure as a self-organized decision-making problem upon which the system can select automatically the next best control action that will accommodate the transition from an unsafe to a safe state. This extends the usability of the STPA analysis since it turns it from a tool utilized during the system design phase into an active tool suitable for the operational phase, that can help controllers to infer selected actions to prompt the system into balanced safety levels. This allows us to establish an efficient method in reflecting dynamics features of system behavior in real-time.

Towards this direction, we seek to transform the STPA control structure into a policy-driven action selection mechanism able to bring the system into a safety equilibrium. When we cope with a decision, there are several different actions that we could select. Choosing the best action requires reasoning about more than just the immediate effects of this action. The immediate effects are often easy to see, but the long term effects are less obvious [29]. In safety control structure, we are interested in selecting an action that will lead the system from an unsafe to a safe state. However, the future impact of an action currently appears as a wise selection could lead to hazard. Therefore, a trade-off between the immediate rewards and the future gains is required to yield the best possible solution.

3.2.1. Markov Decision Processes

An HRC system in the operational phase can be considered as a dynamic scheme of states led by specific actions. Thus, we can conveniently model our problem as a Markov Decision Process (MDP). Following the MDPs formulation, we define a set of system states, a set of actions, the effects of the actions and the immediate value of each action. The state is the way the ecosystem (i.e., the environment, the human and the robot) currently exists, and an action will have the effect of changing the state of this ecosystem. The objective is to select the best action for a particular state, since alternative actions have different outcomes and may transit the system to a different state. The automation of decision-making in MPDs is achieved by associating selected actions with a value. The transition is modeled by specifying the resulting next state for each starting state and action. However, there is a certain limitation in classical MDPs, where the main assumption is that the next state depends only upon the current state [30]. Although this assumption can be valid for certain applications, in safety system scenarios, where time evolution of the facts also plays a major role, a sequential combination of CA and UCAs advocate in the system's transition from safe related states to hazard states and eventually in states associated with losses [27]. However, based on the Markov assumption, the next state is solely determined by the current state (and current action) and, thus, the MDP model is not capable of modeling these situations directly.

3.2.2. State Uncertainty in Markov Decision Processes

Classic MDPs assume that the current state is completely known and observable. However, in real-life conditions, the discrete state of each system state examined by system–theory applications cannot be fully modeled, since it is not trivial to measure and observe all the parameters that could potentially influence the ecosystem, e.g., the human mood while performing a collaborative task. To this end, each state is considered as partially known. Partial observability of the state is opposed to the MDP principals. To tackle this limitation, we incorporate partially observable Markov decision processes (POMDP) [31], where instead of directly observing the current state, the state is linked to an observation that indicates a clue regarding the state it belongs to. The observations are considered as

the outcome of the selected actions—either CA or UCA—and can also be probabilistic. Therefore, there is a need to specify an observation model that explains the probability of each observation for each state in the system given a specific applied action. It turns out that the maintenance of a probability distribution over all of the states boils down to provide the same information as if we maintained the complete history of the states [32,33]. In particular, when a selected action is performed and this action provides an observation, the distribution is updated accordingly. POMDPs constitute a convenient representation for tracking the different states of the system and provide selected UC and UCA that produce observations in a probabilistic manner. By associating each selected action with a system state of a particular safety level, we can formalize the STPA products as POMDP problem.

4. Methodology

4.1. STPA-Related POMDP Formulation

In our approach, we interpret the STPA outcomes and link them with specific ecosystem states and actions, either CAs or UCAs. For the proposed problem formulation, we read the generic POMDP design theory [34] in a safety-related manner, considering the explicit constraint and unconstrained dynamics where the problem domain comprises in our case study the environment, the operator, the robot and the peripheral monitoring components. A discrete POMDP is designed as a tuple $P = \{S, A, \Omega, R\, O, T, b_0\}$ where:

- $S = \{s_1, s_2, ..., s_n\}$ denotes the **States** space that determines the condition of the environment, the operator, the robot and the peripheral monitoring components at each time t.
- $A = \{a_1, a_2, ..., a_n\}$ denotes the **Actions** space that comprises all the system actions (including nominal CAs and UCAs) that the ecosystem agents (i.e., robot and operator) are able to perform so as to complete the collaborative task.
- $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$ denotes the **Observations** space that comprises the outcomes of each above mentioned action, but under the assumption that an observation ω partially describes the state of the previous entities.
- $R = (A, S)$ comprises a **Reward** function that determines the restrictions imposed by penalizing or favoring specific selected actions (A) during the HRC (S).

The measurement of the probability distribution of the initial state (b_0), the states transitions (T) and the observations (O), constitute the fundamental part during the design of POMDPs. This is a challenging task, and the initialization of the probability values over these parameters is typically applied empirically or by assuming a distribution function based on the nature of the problem.

- The probability distribution of the initial state comprises the likelihood about the environment, the robot, the operator and the subordinate components to be in specific state s at the time $t = 0$ such as:

$$b_0(s) = P(s_0 = s) \tag{1}$$

- The probability distribution of the *state transition* comprises the probability of propagating to state s' given that the domain is in state s and the system selects an action a and its respective expression is provided as:

$$T(s, a, s') = P(s_t = s' | s_{t-1} = s, a_{t-1} = a) \tag{2}$$

- The probability distribution of the *observations* comprises the uncertainty of the model to receive a specific observation ω, e.g., feedback, considering that the ecosystem is in state s and the prompting model has selected to perform the action a, also expressed as:

$$O(s, a, \omega) = P(\omega_t = \omega | s_{t-1} = s, a_{t-1} = a) \tag{3}$$

- The probability distribution about the *current state* of the ecosystem to be in s, being partially observable through observation ω. Since it is not possible to define the current state with complete certainty, a belief distribution is maintained to express the history of the selected actions and state transitions of the domain such as at time t, the operator, the robot, the environment and the subordinate monitoring components are at state s considering the sequence of past combination of actions and observations as follows:

$$b_t(s) = P(s_t = s|\omega_t, a_{t-1}, \omega_{t-1}, ..., a_0, b_0) \quad (4)$$

The explicit definition of the aforementioned probabilities indicate the design of a well-formed POMDP model, the solution of which can be achieved through the existing solvers [35]. The outcome of this solution is an action selection policy π that maximizes the sum of the expected future reward up to a specific time. This policy comprises a mapping from the current state belief probability to the action space A. Given the computed policy, the model can select an optimal action by computing its belief state based on the following update rule:

$$b'(s') = \frac{O(s', a, \omega) \sum_{s \in S} T(s, a, s') b(s)}{P(\omega|a, b)} \quad (5)$$

where b' is the updated belief, b is the given belief at the previous time step and (a, ω) is the latest combination of model selected action and observation. This policy constitutes the real time "map", which, based on the current observation probability ω_i at time t, the model selects a control action, initially defined in the STPA, that prompts the system to transit in the next state s_{t+1}.

4.2. Prompting on Different Safety Levels

In our safety-related decision making method, we consider as "safe states" the normal operation behavior of the system, and as safe "control actions" (CAs) the nominal actions under which the system operates without provoking any hazards. Moreover, we consider as "unsafe states" the hazardous situations of the system, which occur under the execution of an "unsafe control action" (UCA). During the STPA analysis, the unsafe states of the system along with the UCA of the studied controllers are extracted during the definition of the circumstances under which a loss scenario will occur. We employ the UCA and hazardous states and the normal operation states with their respective nominal CAs to determine the complete state space that define a collaborative workflow along their respective observations. These parameters can express an optimal policy π, through a POMDP model, which can be utilized as a safety prompting model to continuously monitor the ongoing task.

We interpret the collaborative tasks as a POMDP problem, the objective of which is to decide on each state the optimal action, in order to resolve the defined task with safety. We select to read the state and action space based-on the safety levels as dictated in the STPA analysis. This will provide our safety monitoring system with the equivalent loss scenarios. Since the state space is partially observable, it is conceptually grouped by defining blocks of states that correspond to discrete safety levels $\overline{S} = \{S_U, S_R, S_S\}$. Herein, the state space is partitioned in three safety levels, namely *Unsafe*, *Recover* and *Safe*. The states that belong to the S_U safety level group correspond to instances of the collaborative workflow that the system is completely unsafe and may lead to hazard situations. The S_R safety levels define the group of states within the collaborative task in which the robot/operator have already engaged in recover/corrective action and the level of hazard has been allayed. The states that may belong to the S_S safety level group dictate the instances in the collaborative workflow that the system is completely safe and the task propagates normally. The design principle behind the proposed POMDP formulation mechanism is schematically exhibited in Figure 2.

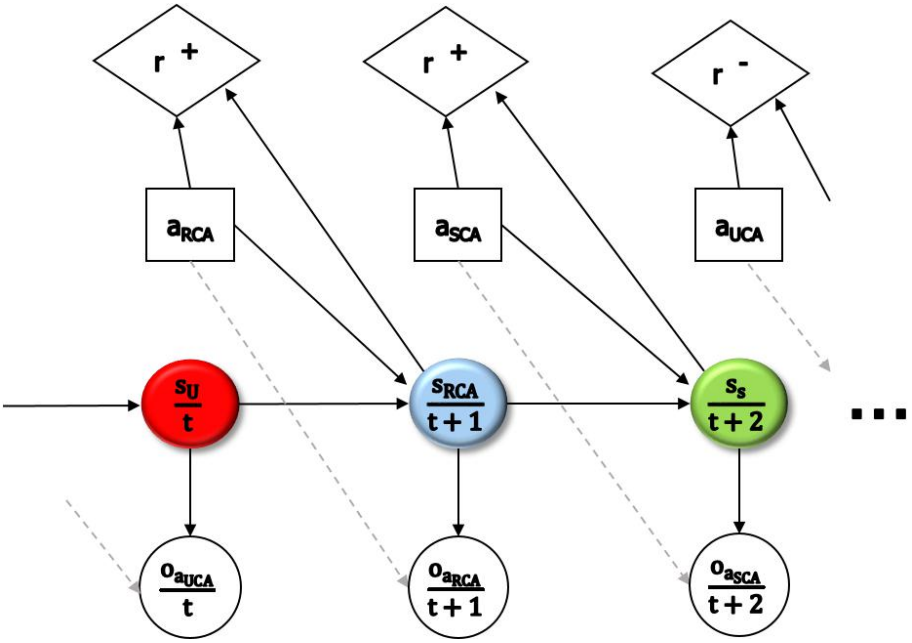


Figure 2. A state transition automaton for the proposed Markov decision process formulation.

The main advantage of the state space partitioning is that it indirectly defines groups of system actions, either performed by the robot or the operator, the context of which is related to the type of the intervention required for the completion of the collaborative task, given the current system safety level. Following the same strategy, the action space is partitioned as $\overline{A} = \{A_{UCA}, A_{RCA}, A_{SCA}\}$. The A_{UCA} set corresponds to completely unsafe control actions, as defined from the STPA, that may lead the system to a hazardous situation; the A_{RCA} set reflects the set of recovery control actions applied from the robot or the operator in order to bring the system to a safer state or to avert an unsafe control action. The A_{SCA} set involves the safe control actions performed by the system to complete the collaborative task. In more detail, the A_{UCA} set involves all the faulty, erroneous and unsafe control actions identified during the STPA analysis and classified as potential fact for the violation of the system level constraints over the identified control loops in the collaborative workflow, such as inaccurate reporting of collision space, failure of the proximity mechanism, insufficient grasping, failure of human–machine interaction interfaces, malfunction of the task planner, communication drop in augmented reality engine, collapse of hardware sensors, etc. The selection and execution of such actions can lead to hazardous situations and eventually can provoke losses. In order to transit the system from an unsafe state s_U to a safer state s_S, each s_U should be associated with mitigation actions that could prevent hazard situations and will bring the system to the nominal control loop. Such actions are considered those that belong to the A_{RCA} group, which is related to the moderation processes nested within collaborative tasks to ensure that the ongoing procedures are performed with safety. This group of actions involves, among others, the communication modalities supporting activities such as notifications in the human machine interaction interfaces, messages exchanged among robot and operator and vice versa, task assignment control commands, displays, warnings and interaction with the augmented reality modules, etc. Lastly, A_S corresponds to the group of safe control actions that are linked to the nominal operation of the controllers foreseen in the collaborative workflow of the ecosystem, that respect the initially identified system level constraints posed during the STPA analysis.

The objective of the designed model is to propagate the system to S_S set of states, by selecting the corresponding set of actions. This is further regulated by wisely assigning the values at the reward function, thus endorsing the system such as, a positive reward value is assigned to the model when the selected action transits the system from unsafe

state S_U to a safe state S_S, while a negative reward value is assigned to the model when the selected action tends to bring the system from a safe state S_S to an unsafe state S_U or from recovery state S_{RAC} to an unsafe state S_U. In situations where the system transits from a state S_{RAC} to another state S_{RAC}, i.e., the same safety level related to recovery, a uniform distribution is applied in the rewards function. It is revealed that the proposed system is designed with emphasis to increase safety levels as a subsidiary mechanism that suggests the most suitable action to be executed in order to avoid hazardous situations. Last, the probability distribution of the observations $O(s, a, w)$ tends to bring the system in less safe state, while the selected actions $\overline{A}$ attempts to stabilize the system in states associated with safe normal operation. Table 1 summarizes the mathematical notations utilized for the theoretical introduction of the STPA-related POMDP formulation.

Table 1. Table with mathematical notations.

Symbol	Description
S	The set of different states of the environment
s	A specific system state
A	The set of different actions that can take place
a	A specific action
Ω	The set of all observations appear upon the action space
ω	A unique observation based on an action a
b_0	The probability distribution of the initial state s
T	The probability distribution of the transition among states
O	The probability distribution of the observations
π	Optimal policy
$\overline{S}$	The safety related partitioned state space
S_U	A specific unsafe state
S_R	A specific recovery state
S_S	A specific safe state
$\overline{A}$	The safety related action space
A_{UCA}	A specific unsafe control action
A_{RCA}	A specific recovery control action
A_{SCA}	A specific safe control action

5. Application of STPA with POMDP in Collaborative Tasks

We apply the SPTA analysis in a human–robot collaborative manufacturing task in order to identify the safe and unsafe states along with the respective control actions. Then, we deploy the proposed POMDP interpretation methodology to extract a policy graph to be utilized as a supervision tool during the operational phase.

As shown in Figure 3, a human worker and a collaborative arm manipulator in a fenceless manufacturing cell is considered for our application scenario. To this end, we apply the STPA hazard analysis technique to a Safe Robot Manipulator (SRM) in an unconstrained cell with a human operator. To keep our analysis as straightforward as possible, yet with realistic parameters, the SRM has specific inherent safety features such as force/torque control, proximity sensing, adaptive operation speed and vision-based monitoring. The human operator employs specific tools for the collaboration with the robot such as gestures, AR-wearable interfaces and hand-held devices. These safety tools are considered as standard technologies in the contemporary collaborative robots, where the means to established the safe collaboration with human are also well studied [3]. We identified two aspects in collaboration for our application:

- **Sequential tasks**, in which the robot performs one task and after its completion the human performs the next task (e.g., the robot places the components on the workstation and, later, the human handles these components).

- **Simultaneous collaboration tasks**, in which the robot and the operator work together on the same component in order to complete a task (e.g., the robot holds a heavy component tight while the human processes its subordinate parts).

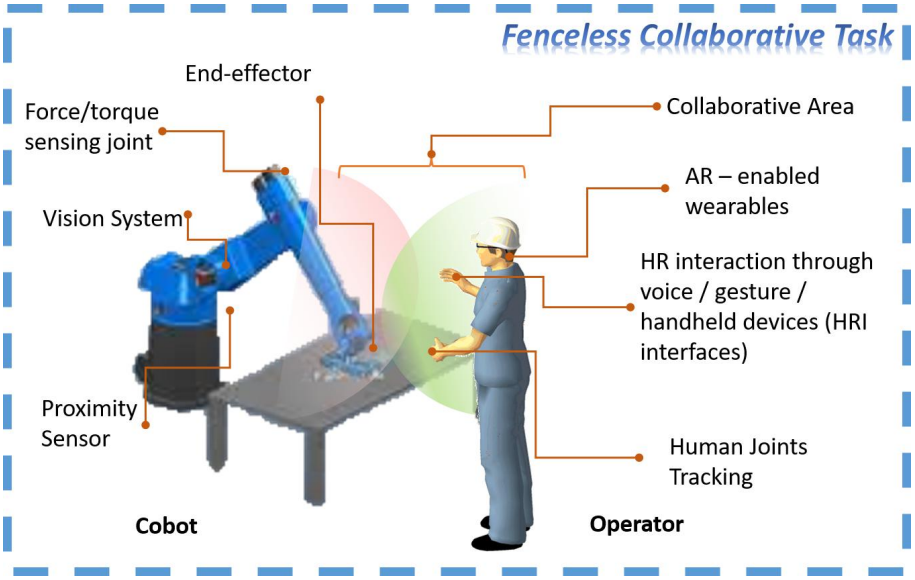


Figure 3. A graphical illustration of a fenceless collaborative topology including an operator and a collaborative robot along with the safety mechanisms.

We follow the standard formalization of STPA steps to perform our analysis, which are summarized below for the sake of completeness of the present study:

1. Investigate the system structure, goals, components, requirements, functions and components interaction.
2. Elicit the system requirements and safety constraints.
3. Identify the potential accidents and unacceptable losses and the hazards at system level.
4. Draw the functional control diagram of the system.
5. Apply step 2 and step 3 of STPA to the control structure.
6. Refine the safety requirements and constraints.

5.1. Process Analysis

We seek to define the possible accidents and system-level hazards in the human–robot collaborative manufacturing task. It is apparent that the potential accidents are related with the injury of the human while working with the robot in the collaboration area. This type of accidents can be occurred on specific phases of the collaboration procedure:

- **Accident 1**: The robot injures the human while a task is performed in the common shared workspace.
- **Accident 2**: The robot injures the human while they work in separate workspaces.

Before the beginning of the analysis, the STPA demands the identification of the stakeholders as well as the determination of the losses on which this analysis should be focused. Thus, in a human–robot collaborative task that takes place in an industrial environment, the stakeholders with the associated losses could be:

- **SK1: Manufacturing Company**: where idle and unusable robots hinder the return on investment and lead to the reduction of productivity. The identified losses could be:
  - L-1: Loss of/or damage to robot,
  - L-2: Loss of profit,
  - L-3: Loss of/or damage to objects outside/inside the collaborative area (e.g., infrastructure damages).

- **SK2: Human operator**: where the human life can be threatened from injuries, while also the user experience from the involvement in robotic tasks can be reduced. The identified losses for this stakeholder group could be:
  - L-4: Loss of human life or injury to people,
  - L-5: Loss of trust in automation.

5.2. Identification of System-Level Hazards

The examined system comprises a fenceless collaborative workstation consisted of an SRM, a human operator and the complementary sensors and communication tools, as illustrated in Figure 3. The boundary of this system is the topological area that restricts the aforementioned components to allow clear identification of the hazard causality on the examined case. Thus, any other parts of the industrial area on which the manufacturing task is also referred, e.g., ERP and any other cyberphysical systems, is opted out from our analysis. Upon the definition of the system and the system's boundary, the next step concerns the definition of the system-level hazards by determining system states or conditions that will lead to a loss in worst-case operational and environment circumstances. In order to ensure stability in the analysis and to avoid any redundancy, each identified hazard is trusted to be linked to one or more losses. Based on the SPTA principles, hazards are states or conditions to be avoided, hence we determine the system safety constraints, which will later lead to the determination of the unsafe and safe control states. Then, we iterate again on the hazard analysis and we further classify the hazards in accordance to the components of the system. The conducted analysis led to the identification of seven major group of hazards (H-), which are further analyzed in accordance to the group they belong (H- x,x):

Hazard Analysis:

1. **H-1**: Robot exceeds maximum set collaborative speed while human inserts its workspace (L-4, L-5).
   - Workspace monitoring
     - H-1.1 Workspace monitoring does not report accurately the human position relative to the robot.
2. **H-2**: Robot violates operator's personal space while not being in collaborative mode (L-3, L-4, L-5).
   - Workspace monitoring
     - H-2.1 Occlusions in proximity sensor hinder human position.
3. **H-3**: Robot does not complete the grasping task (L-2, L-3, L-5).
   - Grasping points inference
     - H-3.1 Selected grasping points is out of robot's reach;
     - H-3.2 Grasped object slips from robot's end effector.
4. **H-4**: Robot does not complete the manipulation task (L-2, L-3, L-5).
   - Trajectory plan
     - H-4.1 Robot cannot compute an inverse kinematics solution to reach the grasping goal;
     - H-4.2 Trajectory planning time out due to cluttered environment.
5. **H-5**: Robot keeps moving when an un-intentional contact occurs (L-1, L-2, L-3, L-4, L-5).
   - Hardware sensing
     - H-5.1 Force/torque sensor does not measure the excessive forces.
6. **H-6**: Robot performs unexpected and abrupt movements that are not foreseen in the collaboration flaw (L-2, L-3, L-4, L-5).
   - Task planning

- H-6.1 Robot state does not resemble the monitoring feedback.

7. **H-7**: Human performs unexpected movements that are not foreseen in the collaboration flaw (L-1, L-2, L-3, L-4).
   - Communication
     - H-7.1 HMI reports outdated manufacturing state;
     - H-7.2 AR visualizes outdated robot state.

After the definition of the potential system hazards, the STPA analysis necessitates the study of the system-level constraints, which specify system conditions or behaviors that need to be satisfied to prevent hazards. In our research, we link the identified group of hazards with the respective system-level constraints (SLC) as follows:

System-Level Constraints:

- **SLC-H-1:** Robot must satisfy collaborative standards while operating in collaborative mode.
  - SC-H-1.1 Excessive speeds should be detected to prevent unintentional contact with human.
- **SLC-H-2:** Robot must satisfy minimum separation distances from operator while not being in collaborative mode.
  - SC-H-2.1 The human personal space should be monitored to prevent unintentional contacts.
- **SLC-H-3:** Robot grasping should be maintained within object handling affordances.
  - SC-H-3.1 Grasp stability should be monitored during object transfer.
- **SLC-H-4:** Robot manipulation should be applied within manipulation affordances.
  - SC-H-4.1 Human–robot workspace and robot collision area should be continuously monitored and updated.
- **SLC-H-5:** Robot must differentiate and respond among intentional and unintentional collisions.
  - SC-H-5.1 Real time switch among collaborative and non-collaborative mode.
- **SLC-H-6:** Robot should follow a planned sequence of tasks.
  - SC-H-6.1 Robot should move to the next task upon confirmation of completion of the current one.
- **SLC-H-7:** Human should follow a planned sequence of tasks.
  - SC-H-7.1 The planned sequence of tasks should be mutually apprehended by robot and human.

Upon definition of the system hazards and the system-level constraints, we can determine the control structure which the system's model that is composed of feedback control loops and enforces constraints on the behavior of the overall system. The considered control structure is graphically illustrated in Figure 4, where control algorithms and process models, along with inputs/outputs, are grouped and exhibited with respect to the actions undertaken from each collaborative source in the designed scenario.

The study of the system control structure ends with the identification of thirty four (34) unsafe control actions $\mathbf{A}_{UCA}$- linked to the respective unsafe control states S_U, two (2) recover control actions $\mathbf{A}_{RCA}$- linked to the recover control S_R and six (6) safe control actions $\mathbf{A}_{SCA}$- associated to the normal control states of the collaborative task (S_S). The conducted analysis is summarized in Table 2.

We relied on the STPA approach to identify the unsafe control actions, which stem from abnormal behavior on the expected safe and recover control states, in accordance to which, there are four ways that a control action can be unsafe, referred to in Section 3.1. The determined safe, recover and unsafe control actions are associated with the three determined controllers in our analysis (operator, monitoring and robot controller), along with their subordinate processes and algorithms.

Table 2. Safe ($\mathbf{A}_{SCA}$-), Recover ($\mathbf{A}_{RCA}$-) and Unsafe ($\mathbf{A}_{UCA}$-) control actions for the collaborative robotic task.

Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Too Early, Too Late, Out of Order	Stopped Too Soon, Applied Too Long
$\mathbf{A}_{SCA}$**-1:** Report the collision space [**Cobot Controller**]	$\mathbf{A}_{UCA}$**-1:** Collision space during HRI is not reported due to scene registration errors [H-1.1]	$\mathbf{A}_{UCA}$**-2:** Erroneous collision space is reported when human pose estimation is inaccurate [H-1.1]	$\mathbf{A}_{UCA}$**-3:** Collision space is reported too late due to ROS communication interruptions [H-1.1] $\mathbf{A}_{UCA}$**-4:** Outdated collision space is reported due to communication bandwidth [H-6.1]	-
$\mathbf{A}_{RCA}$**-1:** Report the proximity [**Monitoring Controller**]	$\mathbf{A}_{UCA}$**-5:** Monitoring does not report human's insertion into robot's workspace due to visual occlusions [H-2.1,H-6.1]	$\mathbf{A}_{UCA}$**-6:** Monitoring reports noisy proximity measurements in presence of reflections to the optical sensors [H-1.1, H-2.1, H-6.1]	$\mathbf{A}_{UCA}$**-7:** Monitoring reports obsolete proximity measurements due to ROS bandwidth [H-1.1, H-2.1, H-6.1]	-
$\mathbf{A}_{SCA}$**-2:** Object Detection [**Monitoring Controller**]	$\mathbf{A}_{UCA}$**-8:** Vision system does not provide a target goal to the end-effector when the object classifier fails [H-3]	$\mathbf{A}_{UCA}$**-9:** Vision system provides a target goal to the end-effector with offset due to estimation errors [H-3]	$\mathbf{A}_{UCA}$**-10:** Vision system provides un-ordered sequence of grasping points [H-3.1, H-3.2]	-
$\mathbf{A}_{SCA}$**-3:** Grasping and Manipulation [**Cobot Controller**]	$\mathbf{A}_{UCA}$**-11:** Grasp planner fails to provide a grasping strategy when the current robot state is unknown [H-3, H-4.1, H4.2]	$\mathbf{A}_{UCA}$**-12:** Manipulation planner infers a grasping strategy with outdated grasping points when object has been moved in the scene [H-3.1, H-4]	$\mathbf{A}_{UCA}$**-13:** Trajectory planner infers a sequence of grasping points before the end-effector reach a pre-grasping pose due to ROS latency [H-3.1, H-3.2]	$\mathbf{A}_{UCA}$**-14:** Grasp stops early due to erroneous estimation of release surface [H-3.2, H-4.2]
$\mathbf{A}_{SCA}$**-4:** Task planning [**Cell Controller**]	$\mathbf{A}_{UCA}$**-15:** Task planner does not provide a task ID when unregistered robot state occurs [H-6.1] $\mathbf{A}_{UCA}$**-16:** Task planner does not provide a robot task execution msg when vision system fails to return a pose observations [H-4]	$\mathbf{A}_{UCA}$**-17:** Task planner infers a wrong task ID when human bypasses the scenario flaw with HMI [H-7.1, H-7.2] $\mathbf{A}_{UCA}$**-18:** Task planner provides a wrong robot task execution msgs when vision system infers erroneous poses [H-6, H-6.1] $\mathbf{A}_{UCA}$**-19:** Tasks planner halts when receiving unexpected monitoring msgs during collaborative mode	$\mathbf{A}_{UCA}$**-20:** Task planner provides shuffled robot task execution msgs due to inaccurate robot/environment state [H-7.1, H-7.2]	$\mathbf{A}_{UCA}$**-21:** Task planner terminates the task too late by skipping specific operation states [H-6.1, H-7.1, H-7.2]

Table 2. *Cont.*

Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Too Early, Too Late, Out of Order	Stopped Too Soon, Applied Too Long
$\mathbf{A}_{SCA}$**-5:** HMI interface [**Operator and Monitoring Controller**]	$\mathbf{A}_{UCA}$**-22:** HMI interface does not visualize the current task state when ROS communication is disturbed [H-6.1,H-7.1] $\mathbf{A}_{UCA}$**-23:** HMI interface does not provide error diagnostic msgs when robot states are not reported [H-5,H-6.1, H-7.1]	$\mathbf{A}_{UCA}$**-24:** HMI interface provides access to low-level control,interrupts robot's trajectory [H-4,H-5.1]	$\mathbf{A}_{UCA}$**-25:** HMI interface allows interruption before robot completes a task when collaborative mode is not enabled[H-1, H-2, H-5]	-
$\mathbf{A}_{RCA}$**-2:** AR engine [**AR Controller**]	$\mathbf{A}_{UCA}$**-26:** AR engine does not provide robot/workspace registered information in presence of severe occlusions [H-7.1,H-7.2]	$\mathbf{A}_{UCA}$**-27:** AR engine provides wrong environment/robot state when localization error occurs [H-7.1,H-7.2]	$\mathbf{A}_{UCA}$**-28:** AR engine visualizes robot/environment state with time shift due to ROS communication latency [H-7.1,H-7.2, H-6.1]	-
$\mathbf{A}_{SCA}$**-6:** Robot HW sensors [**Vision and HW Controller**]	$\mathbf{A}_{UCA}$**-29:** Vision system does not provide frames due to ROS bandwidth issues [H-2.1] $\mathbf{A}_{UCA}$**-30:** Robot does not provide force/torque measurements due to ROS bandwidth issues [H-5.1]	$\mathbf{A}_{UCA}$**-31:** vision system reports erroneous depth measurements due to unexpected reflections [H-3,H-4, H-6] $\mathbf{A}_{UCA}$**-32:** Robot reports erroneous force excess due to sensor calibration drift [H-5.1]	$\mathbf{A}_{UCA}$**-33:** Robot vision system provides input with delay, due to ROS synchronization issues [H-3,H-4, H-6] $\mathbf{A}_{UCA}$**-34:** Robot force sensor reports force/torque measurements with smaller frequency than the low-level controller due to payload excess [H-5.1]	-

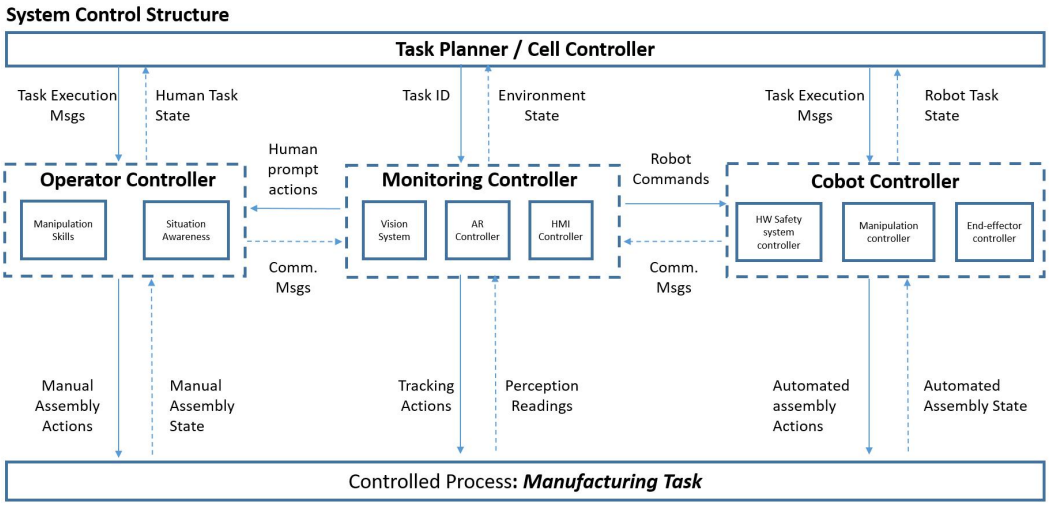


Figure 4. Control structure for the collaborative task.

Upon the definition of the state and action space—conceptually grouped in the three safety levels—the next step of POMDP formulation concerns the identification of the observations. Each action, regardless to its type, is associated with a respective observation. The probability density of the observations for each controller is uniformly distributed among the number of the identified actions for each controller in the system, i.e., each row in Table 2. For example, the probability to obtain any observation related to specific actions while the system "reports the collision space" is 1/5, and the probability to obtain any of the observations related to the specific actions while the system performs the "task planning" is 1/8. After the definition of the observations, the modeling of the safety system operation with the POMDP is complete. We employed the Cassandras's solver [36] with the open source software provided in [37] for the calculation of the police graph π.

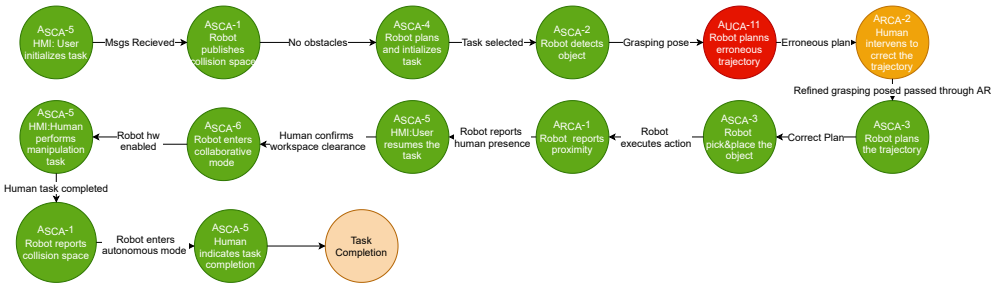
5.3. Feasibility Study

The calculated policy graph π has been employed to assess the defined STPA-related safety model. The main objective of the assessment process is to prove that the resulting policy graph is capable of monitoring the ongoing collaborative task while inferring either successful flows of the scenario or by extracting the loss scenarios during human–robot collaboration.

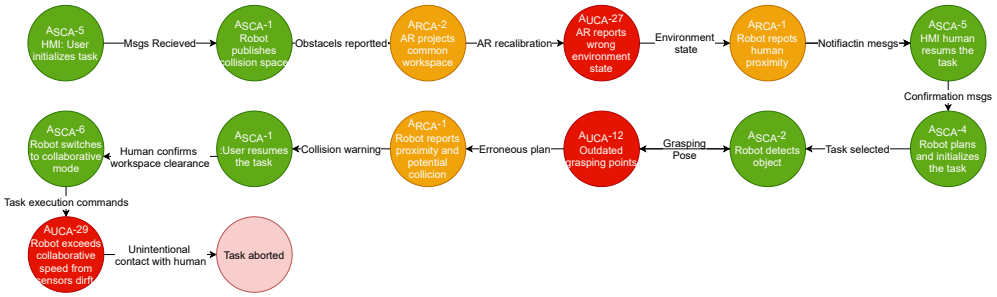
The validation of the policy graph π has been conducted with simulation experiments upon which the collaborative scenario has been executed for 100 epochs. We assumed that, in each epoch, the system initiates from a fixed state, which indicates the start of the collaborative task. For the simulation, we employed the Matlab software and created a sequence of actions and their respective observations with logical connections that can eventually resolve the collaborative task. The policy graph π (calculated through the Cassandras's solver) was utilized as a transition map, in accordance to which, for each selected action, a random selected observation that corresponded to the specific controller was generated through the simulation script. The observation probability function followed the uniform distribution based on the control process that was referred to. Given a selected observation and the past sequence of actions–observations, the transition probability was calculated and based on the indexing of the policy graph π the system transited from one state to the other, populating, respectively, the simulation script to be transited to the next state. In our experiment, each simulation epoch was terminated either when the system has selected the specific action linked to the final state of the collaborative task or when the system has reached an action that indicated that the system cannot recover any further, the collaboration task has not been completed and a sequence of unsafe control actions has occurred. Considering the fact that a loss scenario describes the causal factors that can lead to the unsafe control actions and to hazards, and given the complexity of the studied

system, our POMPD-modeled STPA can track and define loss scenarios associated with the control path. When such a type of situation occurred, the selected—through the POMDP model—sequence of actions constitutes the loss scenarios in accordance to which when specific type of observations occur in each system state, it prompts the system to select the respective actions, the sequence of which end up with a loss scenario.

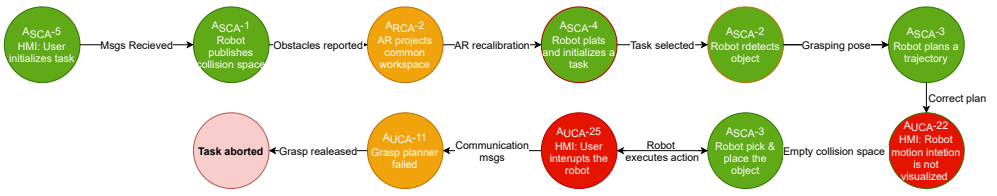
The duration of each simulation epoch was approximately 2 s $\pm$ 0.02 s, measured on an Intel Core i7 CPU at 2.8 GHz. The 60 iterations corresponded to the selection of a sequence of actions that completed successfully the collaborative tasks. The 25 epochs resulted in sequences of actions corresponding to loss scenarios, while the result of the other 15 simulation epochs corresponded to an already appeared loss scenario. Each identified loss scenario has been further examined for its logical flow and found to be valid, i.e., based on specific sequence of safe, unsafe and recovery actions, the identical loss scenario can occur. The evaluation of the simulation scenarios has been performed manually by comparing the sequence of actions with the logical states defined in the initial script. In Figure 5, three selected scenarios are graphically illustrated. On top, the first sequence of selected actions that corresponds to a successful collaborative task is presented. In the middle and the last row of Figure 5, two loss scenarios are outlined. Note that the existence of an unsafe control action A_{UCA} does not indicate that the system will directly pass to a loss scenario. Our model also supports recover states S_R that through to A_{RCA} prompt the system to select actions that will transit the system into safer states S_S.



(a) A successful collaborative scenario.



(b) An indicative loss scenario with multiple selected recovery actions.



(c) An indicative loss scenario with multiple selected usafe control actions.

Figure 5. Example of state diagrams corresponding to three different different simulation executions. In (**a**), the collaborative task has been completed normally with only one unsafe control action, in (**b**)

a loss scenario has been generated and the task has been aborted due to the fact that three unsafe control actions appeared, the final one of which the system could not compensate with a recovery action, and (**c**) another loss scenario appeared early during the task since an unsafe control action interrupted the robot during autonomous mode. Note that the recovery actions typically appeared after the unsafe control actions and tend to transit the system into a safer state. Each circle corresponds to a different state with the outlined selected actions based on the obtained observations indicated as arrows between circles.

6. Conclusions

The presented work contributed to the formulation of a decision-making method based on STPA hazard analysis and Markov decision processes in order to provide a hazard assessment monitoring tool during the operational phase of the system. We interpreted the STPA outcomes into a partial observable Markov decision process by associating a group of system's states with the respective control actions conceptually clustered based on their context. The partial observability compensates system uncertainties stemming from noisy sensors measurements and stochastic human behavior errors. A policy graph is computed based on which, given the current observation, the system probabilistically selects the next action that brings it to a safer state level. We performed STPA in a human–robot collaborative task and associated the safe, unsafe and recover states with the respective group of actions to formulate the respective POMDP model, which has been assessed in a simulation environment and proved adequate to identify loss and success scenarios given the provided observations. Contrary to the current state of art methods that employ STPA analysis as a passive tool that can identify the potential loss scenarios in a system, we introduced an STPA-based decision-making tool that can be utilized in the operational phase in order to prompt the system to select an action that will bring it to a safer level, thus avoiding hazardous situations. An obvious limitation of our method is the need to completely define the state space of the examined system and manually associate each state with any safe or unsafe control action. Moreover, even if the method aims to turn SPTA into a decision making tool, so as to monitor the studied system, the complete STPA analysis should be firstly conducted in order to define the unsafe control actions.

Author Contributions: Conceptualization, A.Z. and I.K.; methodology, A.Z., I.K. and I.D.; formal analysis, A.Z.; writing—original draft preparation, A.Z.; writing—review and editing, A.Z., I.K., I.D.; visualization, A.Z.; supervision, I.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bragança, S.; Costa, E.; Castellucci, I.; Arezes, P.M. A brief overview of the use of collaborative robots in industry 4.0: Human role and safety. In *Occupational and Environmental Safety and Health*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 641–650.
2. Vysocky, A.; Novak, P. Human-Robot collaboration in industry. *Sci. J.* **2016**, *9*, 903–906. [CrossRef]
3. Zacharaki, A.; Kostavelis, I.; Gasteratos, A.; Dokas, I. Safety bounds in human robot interaction: A survey. *Saf. Sci.* **2020**, *127*, 104667. [CrossRef]
4. Munoz, L. Ergonomics in the Industry 4.0: Collaborative robots. *J. Ergon.* **2017**, *7*, e173. [CrossRef]
5. Patalas-Maliszewska, J.; Krebs, I. A model of the tacit knowledge transfer support tool: CKnow-board. In *International Conference on Information and Software Technologies*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 30–41.
6. Ballestar, M.T.; Díaz-Chao, Á.; Sainz, J.; Torrent-Sellens, J. Knowledge, robots and productivity in SMEs: Explaining the second digital wave. *J. Bus. Res.* **2020**, *108*, 119–131. [CrossRef]
7. ISO. *Robots and Robotic Devices–Safety Requirements for Industrial Robots–Part 2: Robot Systems and Integration*; International Organization for Standardization: Geneva, Switzerland, 2011.

8. BSI Group. *Robots and Robotic Devices—Collaborative Robots (ISO/TS 15066: 2016)*; BSI Standards Publication: London, UK, 2016 .
9. Zacharaki, A.; Kostavelis, I. Dependability Levels on Autonomous Systems: The Case Study of a Crisis Management Robot. In *Robotic Systems: Concepts, Methodologies, Tools, and Applications*; IGI Global: Xanthi, Greece, 2020; pp. 1377–1390.
10. International Organization for Standardization (ISO). *Safety of Machinery—General Principles for Design—Risk Assessment and Risk Reduction*; International Organization for Standardization: Geneva, Switzerland, 2010.
11. Askarpour, M.; Mandrioli, D.; Rossi, M.; Vicentini, F. SAFER-HRC: Safety analysis through formal verification in human–robot collaboration. In *International Conference on Computer Safety, Reliability, and Security*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 283–295.
12. Guiochet, J.; Machin, M.; Waeselynck, H. Safety-critical advanced robots: A survey. *Robot. Auton. Syst.* **2017**, *94*, 43–52. [CrossRef]
13. Dhillon, B.S.; Fashandi, A. Safety and reliability assessment techniques in robotics. *Robotica* **1997**, *15*, 701–708. [CrossRef]
14. Zhuo-Hua, D.; Zi-xing, C.; Jin-xia, Y. Fault diagnosis and fault tolerant control for wheeled mobile robots under unknown environments: A survey. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 3428–3433.
15. Guiochet, J. Hazard analysis of human–robot interactions with HAZOP–UML. *Saf. Sci.* **2016**, *84*, 225–237. [CrossRef]
16. Ishimatsu, T.; Leveson, N.G.; Thomas, J.; Katahira, M.; Miyamoto, Y.; Nakao, H. Modeling and hazard analysis using STPA. In Proceedings of the 4th IAASS Conference, Making Safety Matter, Huntsville, AL, USA, 19–21 May 2010.
17. Sulaman, S.M.; Beer, A.; Felderer, M.; Höst, M. Comparison of the FMEA and STPA safety analysis methods—A case study. *Softw. Qual. J.* **2019**, *27*, 349–387. [CrossRef]
18. Bensaci, C.; Zennir, Y.; Pomorski, D. A Comparative Study of STPA Hierarchical Structures in Risk Analysis: The Case of a Complex Multi-Robot Mobile System. In Proceedings of the 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, 20–22 December 2018; pp. 400–405.
19. Gleirscher, M.; Johnson, N.; Karachristou, P.; Calinescu, R.; Law, J.; Clark, J. Challenges in the Safety-Security Co-Assurance of Collaborative Industrial Robots. *arXiv* **2020**, arXiv:2007.11099.
20. Bensaci, C.; Zennir, Y.; Pomorski, D. A New Approach to System Safety of human-multi-robot mobile system control with STPA and FTA. *Alger. J. Signals Syst.* **2020**, *5*, 79–85. [CrossRef]
21. Bensaci, C.; Zennir, Y.; Pomorski, D.; Innal, F.; Liu, Y.; Tolba, C. STPA and Bowtie risk analysis study for centralized and hierarchical control architectures comparison. *Alex. Eng. J.* **2020**, *59*, 3799–3816. [CrossRef]
22. Vicentini, F.; Askarpour, M.; Rossi, M.G.; Mandrioli, D. Safety assessment of collaborative robotics through automated formal verification. *IEEE Trans. Robot.* **2019**, *36*, 42–61. [CrossRef]
23. Villani, V.; Pini, F.; Leali, F.; Secchi, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* **2018**, *55*, 248–266. [CrossRef]
24. Kostavelis, I.; Vasileiadis, M.; Skartados, E.; Kargakos, A.; Giakoumis, D.; Bouganis, C.S.; Tzovaras, D. Understanding of human behavior with a robotic agent through daily activity analysis. *Int. J. Soc. Robot.* **2019**, *11*, 437–462. [CrossRef]
25. Knegtering, B.; Pasman, H. The safety barometer: How safe is my plant today? Is instantaneously measuring safety level utopia or realizable? *J. Loss Prev. Process. Ind.* **2013**, *26*, 821–829. [CrossRef]
26. Chatzimichailidou, M.M.; Karanikas, N.; Dokas, I. Measuring safety through the distance between system states with the RiskSOAP indicator. *J. Saf. Stud.* **2016**, *2*, 5–17. [CrossRef]
27. Zeleskidis, A.; Dokas, I.M.; Papadopoulos, B. A novel real-time safety level calculation approach based on STPA. MATEC Web of Conferences. *Edp Sci.* **2020**, *314*, 01001.
28. Alemzadeh, H.; Chen, D.; Lewis, A.; Kalbarczyk, Z.; Raman, J.; Leveson, N.; Iyer, R. Systems-theoretic safety assessment of robotic telesurgical systems. In *International Conference on Computer Safety, Reliability, and Security*; Springer: Cham, Switzerland, 2014; pp. 213–227.
29. Geist, M.; Scherrer, B.; Pietquin, O. A theory of regularized markov decision processes. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 2160–2169.
30. Alizadeh, S.; Sriramula, S. Reliability modelling of redundant safety systems without automatic diagnostics incorporating common cause failures and process demand. *ISA Trans.* **2017**, *71*, 599–614. [CrossRef]
31. Cassandra, A.R. A survey of POMDP applications. In *Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes*; 1998; Volume 1724. Available online: http://www.cassandra.org/arc/papers/applications.pdf (accessed on 1 January 2021).
32. Spaan, M.T. Partially observable Markov decision processes. In *Reinforcement Learning*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 387–414.
33. Kostavelis, I.; Giakoumis, D.; Malassiotis, S.; Tzovaras, D. A pomdp design framework for decision making in assistive robots. In Proceedings of the International Conference on Human-Computer Interaction, Vancouver, BC, Canada, 9–14 July 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 467–479.
34. Littman, M.L. A tutorial on partially observable Markov decision processes. *J. Math. Psychol.* **2009**, *53*, 119–125. [CrossRef]
35. Shani, G.; Pineau, J.; Kaplow, R. A survey of point-based POMDP solvers. *Auton. Agents-Multi-Agent Syst.* **2013**, *27*, 1–51. [CrossRef]
36. Meuleau, N.; Kim, K.E.; Kaelbling, L.P.; Cassandra, A.R. Solving POMDPs by searching the space of finite policies. *arXiv* **2013**, arXiv:1301.6720.
37. Cassandra, A.R. Pomdp.Org. 2003–2021. Available online: https://www.pomdp.org/code// (accessed on 1 January 2021).