COBOT Safety Awareness: A RealTSL Demonstration in a Simulated System

Apostolos Zeleskidis
Democritus University of Thrace
azeleski@civil.duth.gr

Stavroula Chalarampidou
Democritus University of Thrace
stcharal@civil.duth.gr

Ioannis M. Dokas
Democritus University of Thrace
idokas@civil.duth.gr

Frantsi Torra
Democritus University of Thrace
frantorr@civil.duth.gr

ABSTRACT

This work aims to propose the RealTSL methodology to empower collaborative robotic systems with self-safety awareness capability and address the methodology's limitation in determining time ranges for the unsafe system state transitions, which are inputs of the methodology. The COBOT system used in this paper to demonstrate RealTSL is an automated scissor lift robot to be used by first responders for "work at height," simulated in Simulink™. The demonstration begins by 1) applying STPA to the system, 2) applying Early Warning Sign Analysis based on STAMP (EWaSAP), 3) creating an acyclic diagram that depicts system state transitions towards unsafe states, 4) incorporating the appropriate sensory equipment in the simulation, 5) simulating the system's operation for different scenarios using fault injection and finally 6) use information from the simulations to complete the RealTSL analysis and calculate the safety level of the system in real-time during its simulated operation.

Keywords

COBOT, Safety awareness, STAMP, Simulation, Safety level,

INTRODUCTION

Due to the emerging needs of manufacturing and other industries, robots are being called to interact with humans and other robots much more closely than before. This has led to the development of a new type of robots called COBOTS or collaborative robots. For example, due to the high risk involved in many crisis management operations (i.e., firefighting, search and rescue SAR missions, work during a pandemic, etc.) COBOTS are being used to substitute for humans that otherwise would be subjected to high-risk environments. For instance, COBOTS were used to reduce the potential safety risk for workers during the COVID-19 pandemic (Doyle-Kent and Kopacek, 2022). As Wilk-Jakubowski et al. (2022) noted after reviewing 74 scientific papers on the use of robotics in crisis management "*the limitations of humans in operating in dull, dirty, and dangerous environments make it necessary for the use of robots in their place.*". They also identified that there is a "*need for technical solutions that would enable more effective robot implementation in crisis management*".

Since humans and robots are called to collaborate directly, new responsibilities (i.e., avoiding collisions or real-time course correction, being aware of their environment, and adapting quickly to changes, etc.) have been assigned to these robotic systems on top of the responsibilities they held prior (i.e., repeatable, precise movements without delays, etc.). The result is the exponential increase in the complexity of these COBOT systems compared to their "isolated" predecessors. The resulting complexity has made ensuring their safe operation a problem that scientists and engineers are trying to address.

The literature review conducted by Zacharaki et al. (2020) presents different approaches attempted by the COBOT community to ensure the safe interaction between humans and robots. Some of these approaches include but are not limited to: The enhancement of the perception, cognition, and hardware safety of these systems, vision-based methodologies to monitor working areas and avoid collisions, compliant control, flexible and soft materials to decrease the chance of injuries, force, and torque sensing systems to keep robot movements in check and the use of risk assessment and hazard analysis techniques to design competent safety systems.

One of the gaps in the literature that Zacharaki et al. (2020) identified is that "*robust and generalizable safety methods are required to enable safe incorporation of robots into homes, offices, factories or any other setting*" and noted that the way to achieve this is by "*active vision mechanisms which should be incorporated within the future robots to convert them from passive environment observers to active explorers*".

This gap can be minimized by enhancing COBOT robotic systems with self-safety awareness capability. One can define self-safety awareness capability as the inherent capability of a system allowing it to be aware of its safety level at any given time. In man-made systems, this capability is dependent, among other things, on system structure, component characteristics, and interactions between them. This capability should, ideally, be paired with a dynamic decision-making process enabling such systems to be aware of their safety level and course correct to a safer future state when necessary.

On the part of self-safety awareness capability, Chatzimichailidou and Dokas (2016) proposed the Risk Situational Awareness Provision methodology (RiskSOAP) which utilizes an indicator that measures the distance of a system from its ideal theoretic state, meaning a safe state. This ideal theoretic state is determined using an STPA (Systems Theoretic Process Analysis) applied to the system in question. This indicator is meant to provide safety awareness capability in the RiskSOAP methodology.

Another generic methodology that can provide self-safety awareness capability called RealTSL (Real-Time Safety Level) was proposed by Zeleskidis et al. (2022). This real-time safety level determination methodology is based on the STAMP (Systems theoretic accident model and process) accident model. The methodology allows for: a) the definition of a system state transitions acyclic diagram that can lead to accidents, b) the definition of the appropriate sensors and data needed to monitor if these system states have occurred in real-time, and c) the calculation of the safety level of the system in real-time. One of the main limitations of the RealTSL methodology is the fact that the methodology requires time data for the transitions of the system under study through different states. One solution is for domain experts to assess these time data empirically.

This paper, on the contrary, proposes the use of system simulations in fault injection scenarios to determine these time range values and applies this sort of analysis in a simulated COBOT environment to demonstrate the potential impact self-safety awareness could bring to complex systems. This paper refers to fault injection scenarios as the purposeful simulation of scenarios that will lead to losses to observe and record how the system will respond.

The rest of the paper is organized as follows: Section 2 will briefly introduce the methodologies used and how the STAMP accident model, STPA, EWaSAP (Early Warning Sign Analysis based on STAMP), and RealTSL work. In Section 3, the COBOT system will be defined in detail, and the analyses will be presented along with the simulation results of the system's simulation, as well as the calculation of the safety level by RealTSL in tandem with the simulation. Section 4 presents the results and conclusions of this paper and discusses possible future works in this research effort.

METHODOLOGIES

RealTSL utilizes in novel manner specific methods and tools from the domain of safety science. These methods are briefly described in the following sections.

STAMP

The STAMP accident model was created by Leveson (2004) and was intended to connect systems thinking principles to safety science. This approach has shown its effectiveness over other accident models (i.e. Domino model (Heinrich, 1931), Swiss cheese model (Reason, 1990), etc.) when applied to high-complexity sociotechnical systems. According to the STAMP accident model, safety should be viewed as an emergent property of the system as a whole. Emergent properties arise from the direct and indirect interactions of system components. STAMP provides the toolsets to define safety constraints that determine potential hazardous states and interactions in given system contexts that ideally should not occur during its operation phase.

According to STAMP, safety is a control problem and not a reliability problem. The goal then becomes the empowerment of the system's safety and not necessarily the prevention of component failures. Based on STAMP one must model the system under study using feedback control loops. A model of a feedback control loop is depicted in Figure 1. It is comprised of a controlled process that receives inputs and produces outputs, a controller who, by utilizing a control algorithm and mental/process model, is in charge of maintaining the safe and productive operation of the controlled process, actuators that enforce the actions taken by the controller to affect the control process and finally sensors that provide the information needed by the controller to maintain the controlled process to acceptable levels effectively.

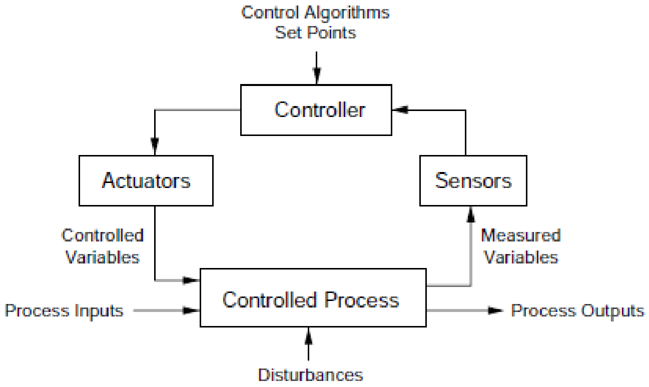


Figure 1. A standard system feedback control loop (Leveson, 2004).

STPA & EWaSAP analyses

STPA

The hazard analysis approach of STAMP is called Systems Theoretic Process Analysis (STPA) (Leveson and Thomas, 2018). The analyst can identify with STPA, among other things, under which operational circumstances and contexts the controllers in the system could enforce unsafe control actions. The term "unsafe" here refers to the occurrence of losses at a system level, including loss of life, injuries, financial losses, productivity losses, etc.

STPA is comprised of the following steps:

(1) This step aims at identifying the system boundaries, the events that constitute system losses, and the system states called hazards, which, when paired with the most detrimental to environmental safety conditions, will lead to those losses, and lastly, the system-level safety constraints that are generated from those hazards.

(2) The aim then is to create a model of the system using feedback control loops in a comprehensive, hierarchical safety control structure with all the appropriate controllers, control algorithms, process models, actuators, sensors, and controlled processes.

(3) The identification of Unsafe Control Actions (UCAs) follows. In this step, the analyst identifies system states or operational contexts in which control actions that may be enforced, not enforced provided too early/late, out of order, stopped too soon, or applied too long by a controller may lead to hazards.

(4) The identification of the loss scenarios that would lead to control actions being improperly executed or unsafe control actions occurring.

EWaSAP

In the context of self-safety awareness capability, the concept of early warning signs is of crucial importance. According to Leveson (2014), there are always signs before major accidents that indicate their future occurrence. However, due to the inefficient capacity of systems to comprehend and manage them properly and promptly, these signs are often "perceived only as noise".

EWaSAP (Dokas et al., 2013) is an STPA extension for identifying early warning signs based on the STAMP accident model that indicates the violation of the safety constraints (Leveson, 2014). It is focused, among other things, on the identification of the appropriate sensors that should be installed in the system, together with the data (i.e., signs) that should be collected by its controllers to comprehend which system states identified by STPA hazard analysis occurred or not in a given time moment. The steps of the EWaSAP analysis are shown below:

(1) The first step of EWaSAP takes place during the definition of system boundaries by STPA, and it aims at identifying entities outside of the system boundaries (i.e., local authorities, first responders, company managers, etc.), who should be informed about the occurrence or progression of hazardous events in the system. This step is most important when dealing with safety critical systems where accidents significantly affect society (i.e., Seveso establishments, power plants, airports, etc.). This step helps to minimize the potential consequences of those hazardous events by lowering response times and delays due to information distribution.

(2) The second step of EWaSAP consists of identifying the data (i.e., the warning signs) that the controllers need to perceive to be able to comprehend the occurrence of unsafe system states described by STPA (losses, hazards, UCA's, loss scenarios), the violation of safety constraints derived by those states and the violation of the assumptions made during the analysis. Next, it identifies the sensory equipment (i.e., proximity sensors, surveillance systems, etc.) that the system must have to collect that data.

(3) The third step of EWaSAP has the goal of employing "awareness actions," namely the transmission of appropriately coded warnings and alert messages, which must be perceived by controllers at a higher level of system hierarchy or by other controllers outside of the boundaries of the system under study, to make them aware of the migration of the system under study from a safe to an unsafe state. The "Issue a Warning" awareness action is enforced by a controller when it comprehends the occurrence of loss scenarios, UCA's, and the violation of constraints tied to them. They are usually provided internally and to agents in charge of the system's operation at that point in time. The "Issue an Alert" awareness actions, on the other hand, are provided to indicate the occurrence of system-level hazards as well as losses. They are usually provided to internal controllers at a higher level of the hierarchy and the external entities identified in the first step of the analysis.

RealTSL methodology

RealTSL is a methodology that combines the above analyses (STPA, EWaSAP) under the STAMP accident model and, together with a non-stochastic mathematical model, aims at providing a real-time assessment of the safety level of control systems. The safety level of a system is defined in RealTSL as "the ordered set of the most detrimental to safety sequences of unsafe system states that result in an accident and are ordered according to the severity of their resulting accident" (Zeleskidis et al., 2022). To calculate this the two phases of RealTSL should be executed. The first is the preparation phase, where the inputs of the methodology are collected, and the mathematical model is calibrated based on them. The second phase is the execution phase, where real-time data from the system are used in conjunction with the calibrated mathematical model to calculate its safety level.

The inputs of the preparation phase and their use are shown in Table 1:

Table 1. RealTSL Inputs at the preparation phase.

Input	Description	Use in RealTSL
Losses, Hazards, Unsafe Control Actions, and Loss Scenarios	System states that are causally connected via the STPA analysis.	They are used to form an acyclic diagram (see Figure 2) that is the basis for the RealTSL mathematical model. The RealTSL acyclic diagram is, in essence, a transformation of the STPA hazard analysis results into a form that the mathematics of RealTSL can use.
Time ranges	Indicate the possible range of time (minimum to maximum) that it would take for the system to transition from one unsafe system state to another, as described in the STPA analysis.	They are used to populate the RealTSL acyclic diagram with time information for every connection between two states, as described in the STPA analysis.

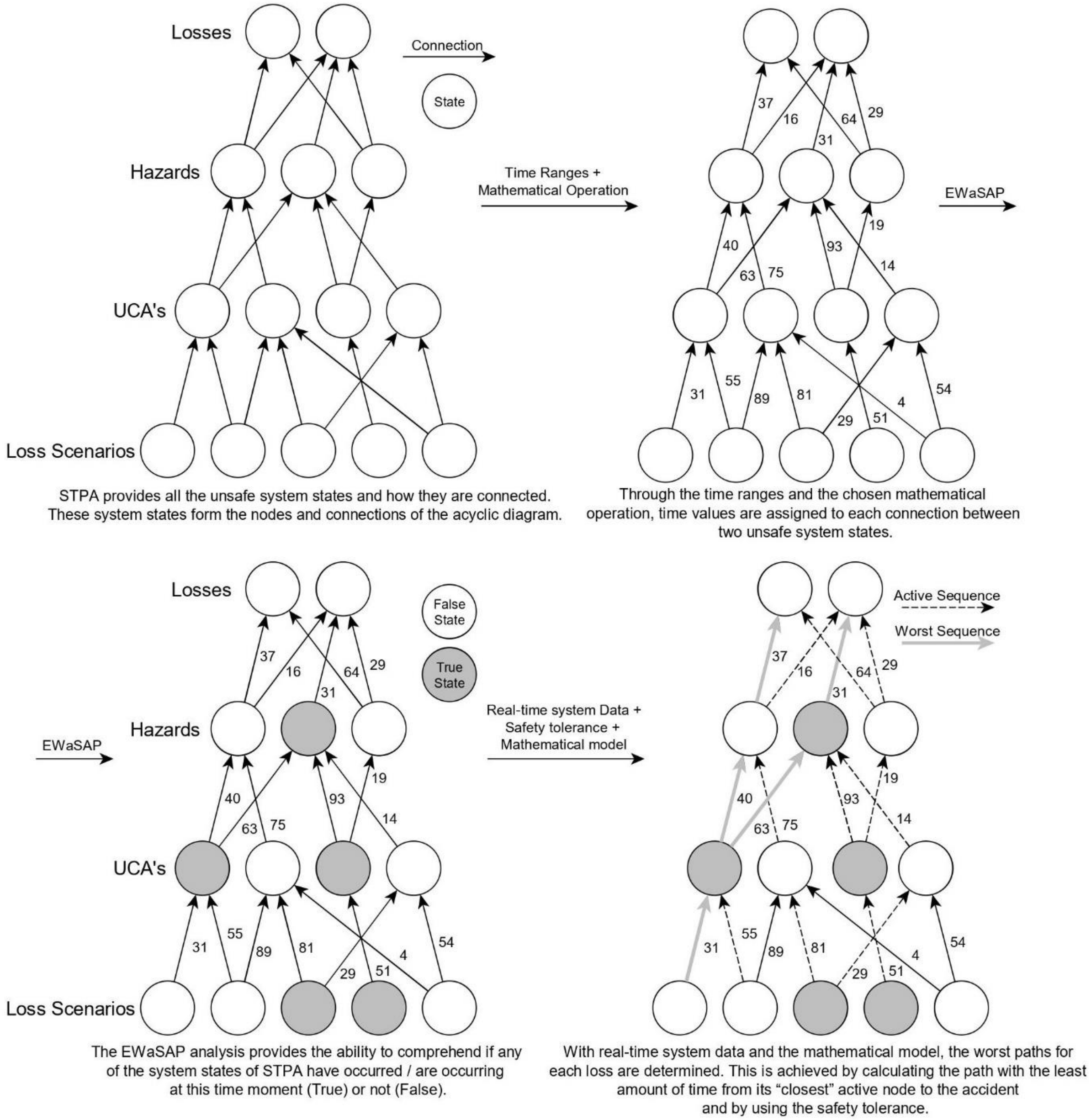
Mathematical operation	Mathematical operations are, for example, minimum (MIN), maximum (MAX), mean-average (MEAN), etc. Any linear transformation that is in accordance with the following rule can be used: $F: [x, y]^2 \rightarrow \mathbb{R}^+, where\ x, y \in \mathbb{R}^+$ This rule states that the linear transformations that have ranges of positive real numbers (range of time values) as their domain and a single positive real number (single time value) as their co-domain.	The mathematical operation aims at narrowing the time ranges to single values for every connection between system states. The operation is selected by those responsible for applying RealTSL in the system under study.
EWaSAP outputs	The data that must be perceived during the operation phase by the controllers in the system to comprehend whether the system states identified by the STPA analysis have occurred or are occurring.	The early warning signs defined by EWaSAP provide the set of data that must be perceived by the sensors of the systems to be aware of the existence of unsafe states, according to STPA.
Safety tolerance	The safety tolerance is a parameter of the mathematical model of RealTSL that is a single-time value selected by those responsible for the effective application of RealTSL in the system under study.	This parameter determines the minimum amount of time considered more detrimental to safety than the progression of the hazardous sequence of system states that will eventually lead to an accident.

The mathematical operation and safety tolerance inputs described in Table 1 allow RealTSL to be flexible to system characteristics and operational contexts. In essence, these input parameters enhance the general use of RealTSL. On the other hand, these parameters should be carefully chosen by those responsible for applying RealTSL because they are essential for the accuracy and reliability of the estimated safety level.

During the execution phase of RealTSL, the system operates as designed. The sensors transmit data indicating if and when a system state identified by the STPA analysis has occurred. RealTSL utilizes the sensor data to comprehend to what extent the system is in a hazardous state. To do this, RealTSL is using the acyclic diagram and its parameters to calculate the most detrimental to safety sequences of unsafe system state transitions according to: a) the time left until the accident occurs, b) the progress of the sequence in terms of system state hierarchy (losses > hazards > UCAs > loss scenarios) and c) the severity of the resulting accidents.

For more details about the methodology and an in-depth analysis of its mathematical model, the reader is referred to the work by Zeleskidis et al. (2022).

Figure 2 depicts a generic representation of how the inputs described in Table 1 are combined to define the RealTSL acyclic diagram during the preparation phase. In addition, it shows how the RealTSL acyclic diagram is utilized by the mathematics of RealTSL during the execution phase.



STPA provides all the unsafe system states and how they are connected. These system states form the nodes and connections of the acyclic diagram.

Through the time ranges and the chosen mathematical operation, time values are assigned to each connection between two unsafe system states.

The EWaSAP analysis provides the ability to comprehend if any of the system states of STPA have occurred / are occurring at this time moment (True) or not (False).

With real-time system data and the mathematical model, the worst paths for each loss are determined. This is achieved by calculating the path with the least amount of time from its “closest” active node to the accident and by using the safety tolerance.

Figure 2. The RealTSL acyclic diagram. The first three figures indicate its creation, and the last shows its use in calculating the safety level of the system (From left to right, top to bottom).

CASE STUDY

In the following section, the results of each step of the RealTSL methodology will be illustrated on a critical subsystem of a COBOT scissor lift. The set of possible scenarios that can lead to an accident in this type of COBOT system is quite extensive such as unanticipated movement of the lift when work is underway or violations of minimum safe distance from obstacles. The case study in this paper will focus on one such scenario, namely the accumulation of weight on the lift's platform in cases such as the evacuation of a large number of people or the provision of heavy equipment and materials to the emergency response team.

Firstly, the scissor lift system will be presented. Then, a description of the critical subsystem where RealTSL will be applied will be given. Following that, RealTSL methodology steps and results will be presented.

COBOT scissor lift system

The simulated COBOT is based on the scissor lift model developed by Miller (2022). It is comprised of a raisable platform with protective rails for workers, a base with wheels for movement through the work site, the cylinders that use expansion or compression from a hydraulic mechanism can lift or lower the platform, and the scissor legs that can constrict and extend. It was used as a basis to showcase a potential tool that can be used in crisis management situations, such as rescue operations executed by the fire department, when "work at height" is needed.

The model developed by Miller (2022) was expanded into a COBOT by incorporating an "Automatic Scissor Lift Controller". This controller has three modules: the self-driving module, the voice command module, and the height controller module. The self-driving module consists of proximity sensors (i.e. LiDAR, ultrasonic sensors, etc.) and a navigation system with the motor of the COBOT acting as an actuator. Its goal is to navigate the scissor lift to the specified work location in a safe manner. The components of the voice command module are microphones that act as sensors, positioned in parts of the scissor lift, and embedded into the equipment of the operators, to be able to capture any spoken commands addressed to the COBOT and a voice translator that can convert the spoken words to commands that the controller can address. The voice command module does not have designated actuators, since it is working as a human to COBOT interface. Finally, the module that this demonstration is focused on is the height controller module, which is used to lift or lower the platform of the scissor lift and is fitted with sensors that monitor position, speed, movement direction, weight, and height. The voice command aspect of the system was incorporated to "free the hands" of the working firefighters when they encounter complex tasks in adverse conditions.

Figure 3 shows a high-level control structure for the COBOT system. The critical subsystem used for the case study is highlighted in gray.

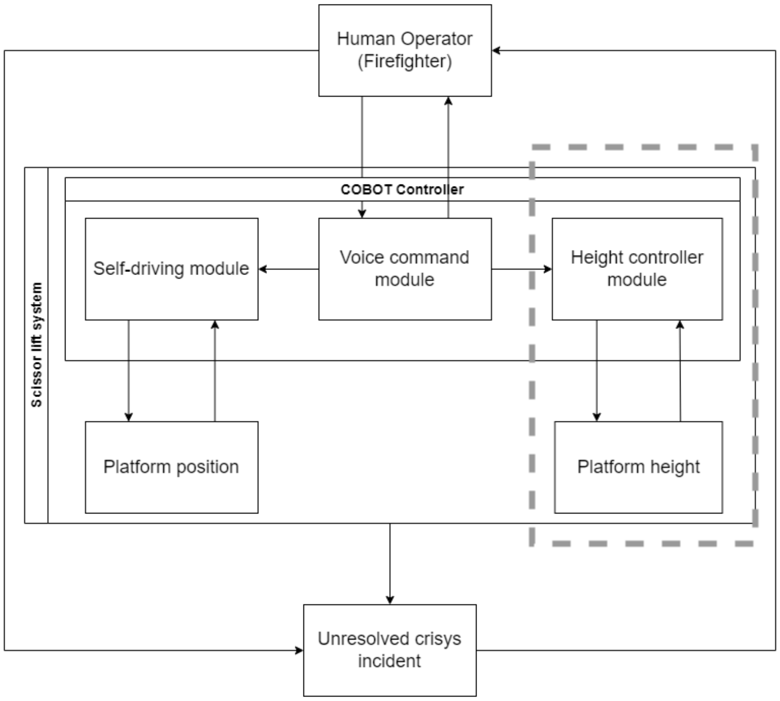


Figure 3. The control structure of the Scissor lift COBOT system.

Height controller module critical subsystem description

The system in this study comprises the scissor lift described in the previous section and the "Automated scissor lift height controller." The automated height controller is in charge of controlling the height of the working platform by managing the hydraulic pressure inside the cylinders. Figure 4 shows the simulated physical model of the scissor lift used in this analysis.

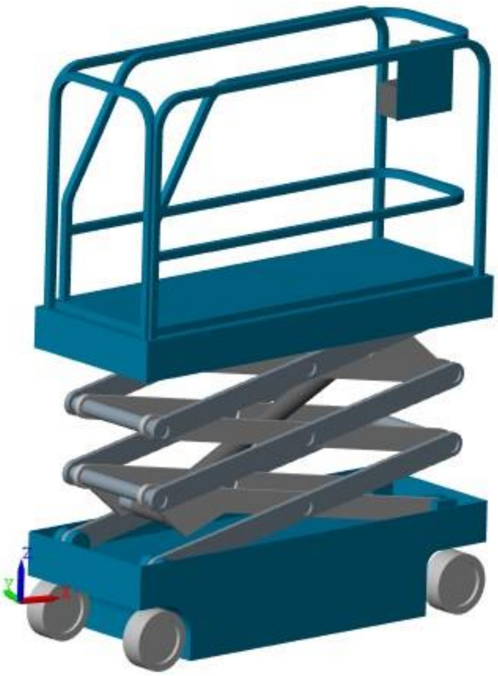


Figure 4. Simulated physical model of the Scissor lift COBOT system.

The height controller operates using the voice commands of the scissor lift operator. The operator or worker can give the commands "ascend", "descend," and "stop", which then, after being translated into inputs from the voice command module the automated height controller will use as the control actions of the system. The automatic height controller also uses a series of feedback, namely the platform's position, speed, movement direction, weight, and height, to have the necessary knowledge of the system state at all times. The system's goals are the ascension and descension of the working platform without delays and in a safe manner for the working personnel and the machinery.

Simulation Tools

To simulate the system in question, a suite of software solutions was used, namely, Simulink® and Simscape™ Multibody™. Simulink® is a multi-domain modeling and simulation environment developed by MathWorks, that can be used to design and simulate systems with multi-domain models. It can be combined with other MathWorks products, like MATLAB®, and includes customizable block libraries.

Simscape™ Multibody™ is a software that provides a multibody simulation environment for 3D mechanical systems, using blocks to model the systems. The systems then can be imported into Simulink® to continue their development or progress their parameterization using MATLAB® and develop control systems and test system-level performance.

The fault injection part of the simulation is conducted by implementing additional parameters in the simulation. For example, to simulate the accumulation of waste work materials on the platform of the scissor lift, a variable was implemented that would raise its value when the platform was positioned at the defined working height and work was being conducted as well as lower its value when the platform was positioned at its lowest position to simulate the "emptying" of the platform from this waste.

The Simulink® setup for this simulation is shown in Figure 5. To record the appropriate information as per the EWaSAP methodology, block switches were used as listeners to check the changes in the parameters of the simulation. The information transmitted from the block switches was then recorded to be used in the RealTSL analysis. In Figure 4 the circled parameters are used for EWaSAP and their use will be described in detail in the following sections.

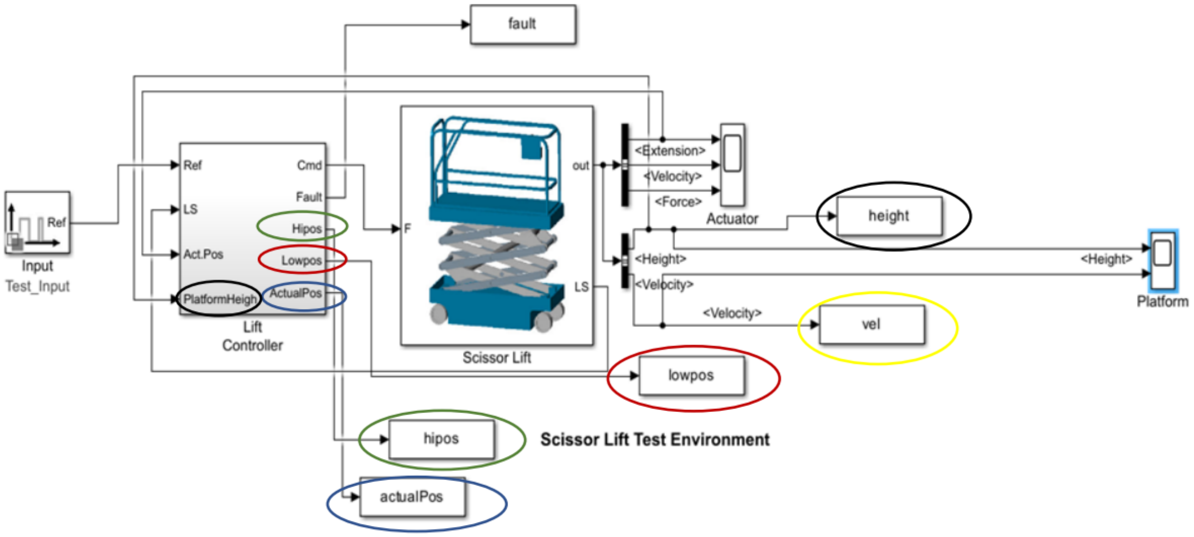


Figure 5. Simulink® setup of the simulated height controller critical subsystem module.

Preparation Phase of RealTSL

STPA analysis

The application of the STPA began with Step 1 of the analysis, where the definition of the losses and system level hazards of the COBOT system was conducted. Examples of losses include loss of life or injuries, property damage, and loss of time. An indicative loss and its connected hazards are presented in Table 2. One potential loss of the system is presented in the first row of the table and the hazards are on the second row. The symbolism in the hazard row identifies if the system state is a loss (L) or hazard (H) as well as any connections this system state has with other system states in brackets [].

Table 2. Losses and system level hazards identified for the Scissor lift COBOT system.

Losses		Hazards	
L-1	Loss of time	H-1	The scissor lift does not stop at the specified working height. [L-1]
		H-2	The scissor lift descends instead of ascending and vice versa. [L-1]

Taking into account the fact that the safety constraints generated in the different steps of the STPA analysis are not utilized by the RealTSL methodology, they will not be presented here.

Next, as per Step 2 of the STPA analysis the control structure of the system is demonstrated, with the visual representation of the control structure shown in Figure 6. The structure is composed of the following elements: the necessary inputs for the system are the voice commands from the scissor lift operator and the time ranges in which the ascend and descend of the platform must be completed, the controller of the system is the "Automated scissor lift height controller," and its three control actions towards the "Platform height" controlled process are: 1) Ascend, 2) Descend, and 3) Stop. These three control actions are given through the "Hydraulic lifting system," which works as an actuator. These control actions are given continuously, meaning that when given by the controller, they are continually provided. For example, the "Ascend" control action is provided until the platform reaches its highest possible height, and then it stops being provided for the platform to stop. The "Stop" control action can override the other control actions. For example, if the "Ascend" control action is being given and the "Stop" control action is given while the "Ascend" control action has not stopped being given, the "Stop" control action will override it and stop the platform's movement.

Some of the feedback needed for the controller to achieve the goals noted above is the platform's position in space, the height of the platform, etc. This feedback is provided through various sensors in the system. The complete Safety control structure of the height controller module of the Scissor lift COBOT system is shown in Figure 6.

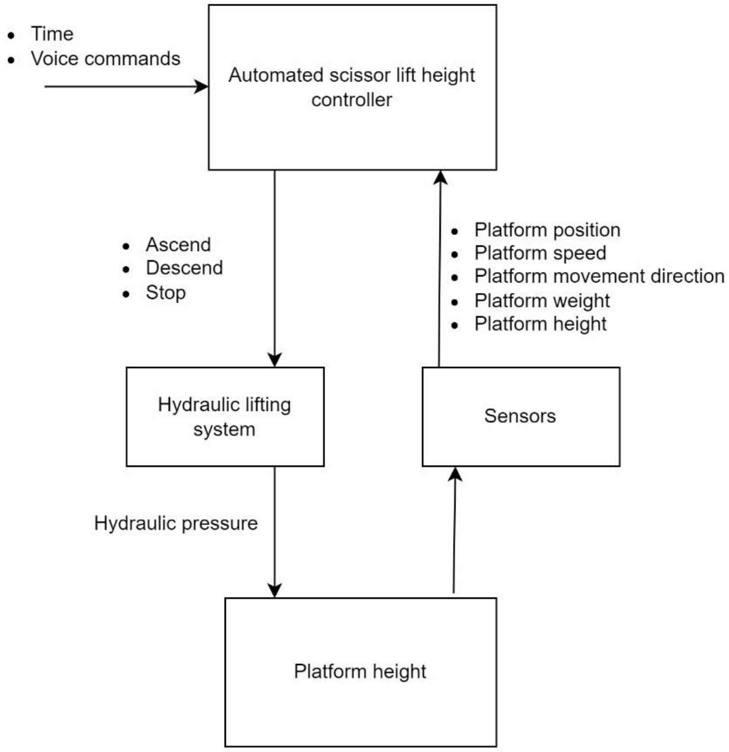


Figure 6. Safety Control Structure for the height controller module.

An indicative set of UCAs (Unsafe Control Actions) that were generated during the application of step 3 of the STPA analysis for the control actions "ascend," "descend," and "stop" are presented in Table 3. The control actions are shown in the first row of the table, the different situations in which control actions might be given and be identified as unsafe are shown in the second to fifth rows. The environmental conditions needed for the UCA to occur are shown in the columns with the symbolism that identifies them as Unsafe control action (UCA) and is numbered accordingly. Any connections with the hazards are shown in brackets [].

Table 3. Indicative Unsafe Control Actions (UCA's) for the Scissor lift COBOT system.

Control Action	Not Providing CA Causes Hazard	Providing CA Causes Hazard	Wrong Timing/Order of CA Causes Hazard	CA Stopped Too Soon/Applied Too Long
Ascend	UCA-1: The automated scissor lift height controller does not give the control action "Ascend" while the platform must move. [H-1]	UCA-2: The automated scissor lift height controller gives the control action "Ascend" while the platform is at the required height. [H-1]	UCA-3: The automated scissor lift height controller gives the control action "Ascend" before the workers and the required equipment get on the platform. [H-1]	-
Descend	UCA-4: The automated scissor lift height controller does not give the control action "Descend" while the platform must move. [H-1]	UCA-5: The automated scissor lift height controller gives the control action "Descend" while the platform is at the required height. [H-1, H-2]	UCA-6: The automated scissor lift height controller gives the control action "Descend" while work is being performed on the platform. [H-1]	-
Stop	-	UCA-6: The automated scissor lift height controller gives the control action "STOP" while the platform is moving to the desired height. [H-1]	-	UCA-7: The automated scissor lift height controller stops giving the control action "STOP" while work is being done on the platform. [H-1]

The final step of the application of the STPA analysis is the generation of the loss scenarios. Some scenarios, with their corresponding UCAs, are presented below.

UCA-7: The automated scissor lift height controller stops giving the control action "STOP" while work is being done on the platform. [H-1]

Scenario 1 from UCA-7: The workers have no way of removing the weight from the platform during work operations, which will lead to increased weight on it and the trigger of the "failsafe mode", to protect the scissor lift structure.

UCA-6: The automated scissor lift height controller gives the control action "Descend" while work is being performed on the platform. [H-1]

Scenario 2: The workers have no way of removing the construction waste from the platform, which will lead to increased weight on it and the trigger of the "failsafe mode", to protect the scissor lift structure.

Scenario 3: The area around the scissor lift was too noisy and the scissor lift's microphone was affected by this.

Scenario 4: A worker operating a scissor lift used the command "Descend" and the microphone of a different scissor lift registered the sound and the command.

Scenario 5: A worker during a talk with other workers, used the word "Descend" and the scissor lift registered the word as the command "Descend".

EWaSAP analysis

The EWaSAP analysis was then conducted on the system states identified by STPA. Table 4 depicts specific EWaSAP outputs, namely the sensory equipment and identifying signs needed for the STPA results shown in the above section. Specifically, each row in Table 4 represents a system state determined by STPA. The first column shows the symbolism that identifies if the system state is a loss (L), hazard (H), Unsafe control action (UCA), or loss scenario (LS) and its identifying number. The second column shows the description of the analyzed system state as well as any connections this system state has with other system states in brackets []. The third column shows the sensory equipment needed to be present in the system to identify the occurrence of the system state. The fourth and final column presents the signs that would indicate the occurrence of the system state by the sensory equipment of the last column.

Table 4. EWaSAP applied to the Scissor lift COBOT system.

Symbols	Description	Sensory System	Identifying Signs
L-1	Loss of time.	The human operator task state signal (via voice command or computer interface). Stopwatch that measures the amount of time operations take.	Status of operations ≠ completed (via the human operator task signal). Operations time exceeded the anticipated time by a large margin (via the stopwatch).
H-1	The scissor lift does not stop at the specified working height. [L-1]	Platform height sensor that records the precise height of the platform in real-time. Platform accelerometer sensor that records the precise speed of the platform in real-time.	The platform is stopped, and its height is not as anticipated for the work that was planned.
H-2	The scissor lift descends instead of ascending and vice versa. [L-1]	The human operator task state signal (via voice command or computer interface). Platform accelerometer sensor that records the direction of the movement of the platform in real-time.	The platforms movement direction is ascending while the operations have concluded, or the platforms movement direction is descending while the operations are still in progress.
UCA-7	The automated scissor lift height controller	The human operator task state signal (via voice command or computer interface).	The platform is moving in any direction while

	stops giving the control action "STOP" while work is being done on the platform. [H-1]	Platform accelerometer sensor that records the precise speed of the platform in real-time. Control action listener that provides a signal whenever a control action is enforced and stopped by the controllers.	operations are taking place on the platform.
LS-1	The workers have no way of removing the weight from the platform during work operations, which will lead to increased weight on it and the trigger of the "failsafe mode", to protect the scissor lift structure. [UCA-7]	Platform weight sensor that records the weight of the platform in real-time.	The platform has an added weight that exceeds the load it can carry safely.

Time values determination & Safety tolerance

The scissor lift simulation was utilized to identify the time values and safety tolerance inputs of the case study. To achieve this, additional sensory equipment was placed in the COBOT system simulation based on the EWaSAP results (see Table 4). The system then was simulated in a manner to produce all sequences of unsafe system state transitions that resulted in an accident. These simulations were reproduced a number of times with different environmental parameters to identify the minimum and maximum amounts of time it would take to transition the system between those system states. These two values then constitute the time ranges.

One of the sequences of unsafe system state transitions that were simulated was the following:

(1) LS-1: The workers have no way of removing the weight from the platform during work operations, which will lead to increased weight on it and the trigger of the "failsafe mode", to protect the scissor lift structure. [UCA-7]
(2) UCA-7: The automated scissor lift height controller stops giving the control action "STOP" while work is being done on the platform. [H-1]
(3) H-1: The scissor lift does not stop at the specified working height. [L-1]
(4) H-2: The scissor lift descends instead of ascending and vice versa. [L-1]
(5) L-1: Loss of time.

The time range values were achieved by finding the difference between the time when the data about the occurrence of one of these states is provided through the sensors and the time the data about the occurrence of a connected state to that is provided. For example, when the simulation of the system starts, a virtual time-recording device starts counting seconds. At 90 seconds the platform weight sensor indicates that 350 kilograms of weight have been deposited to the working platform which indicates the occurrence of the LS-1 system state as shown in Table 4. At 94 seconds the human operator task state signal indicates that work is underway at the platform, at the same time as the Platform speed sensor indicates that the platform is not stopped. This according to Table 4 indicates that the UCA-7 system state has occurred and since LS-1 and UCA-7 are connected, 4 (90-94=4) seconds are one observation about the time range for that connection. This process is repeated with various possible simulation conditions for that connection to determine the time range. The mathematical operation used in this demonstration is the minimum (MIN). This means that for example in the case of a connection having a time range be (4,10), the final time value of the connection is 4 seconds.

The resulting time ranges and time values are shown in Table 5. Each row represents one transition from one system state to the next as defined by STPA. The first column presents the starting system state meaning the state before the transition and the second column the ending system state meaning the state after the transition. These two states are not the only states the system goes through during these transitions. The time ranges that contain the minimum and maximum (min, max) time values it took the system to transition between these states (in the simulations) are in the third column. The last column provides the results of applying the chosen mathematical operation (in this case the minimum) to the time ranges of the third column.

Table 5. Time ranges produced through the system simulation and the resulting time values.

Starting system state	Ending system state	Time range	Time value MIN (Time range)
LS-1	UCA-7	(4,10)	4
UCA-7	H-1	(8.4,15)	8.4
H-1	L-1	(30,300)	30
H-2	L-1	(30,300)	30

The safety tolerance for this demonstration is set at 10 seconds. This means that when there are two potential sequences of system state transitions that will potentially lead to an accident and are at this point in time at the same safety hierarchical level (loss, hazard, UCA, loss scenario), one will be considered most detrimental to safety if it is set to lead to the accident more than 10 seconds faster than the other one.

Execution phase of RealTSL

Real-time safety assessment

To calculate the safety level of the system in real-time, the simulated system was enhanced with the mathematical model of RealTSL to mimic the execution phase of RealTSL in a simulated environment. The scenario presented at the beginning of the Case study section is used here to demonstrate the calculations of the method to assess the safety level of the system. The scenario of the operation unfolds as follows. The COBOT scissor lift begins with no added weight, a horizontal distance of 30 meters and a vertical distance of 3 meters from the needed work area, which is an area that needs to be cleared by firefighter professionals. The platform is loaded with the appropriate equipment for the removal of the debris, the firefighters get on top of the platform and the navigation module of the COBOT transports the scissor lift to the appropriate area. When the COBOT reaches the correct coordinates, the voice command is given to raise the platform to the appropriate height. The time moment that the voice command is given is counted as the t=0. When the platform reaches the appropriate height, the removal of debris begins. During the removal, weight from the debris is accumulated on the platform. Before the needed amount of debris has been cleared, the accumulated weight and the weight of the firefighters and their equipment exceeds the designed weight capacity of the lift and the platform starts to descend. When on the ground, the firefighters remove the vegetation from the platform and give the voice command "ascend", and when the platform reaches the appropriate weight, the removal operations continue. The scenario with corresponding time moments is shown in Table 6.

Table 6. The scenario of the operation sequence of events with corresponding time moments.

Time moment	Description
0	The COBOT scissor lift with the added weight of the workers and the equipment (total weight=250kg), is in the correct area. Voice command "ascend" is given, and the platform starts moving.
10.25	The platform reaches the predetermined height of 3m and removal work begins.
22.5	The platform reaches the maximum weight capacity of 500kg and begins the descent to the minimum height of 1m.
30.9	The platform reaches the minimum height and the removal of the extra weight begins.
35.9	The extra weight is removed and the voice command "ascend" is given.
45	The platform reaches the height of 3m and the work continues.

The calculation of the safety level begins in time moment 22.5 when the system transitions to the unsafe system state LS-1: The workers have no way of removing the weight from the platform during work operations, which will lead to increased weight on it and the trigger of the "failsafe mode", to protect the scissor lift structure. In the RealTSL acyclic diagram depicted in Figure 7, only the sequences of unsafe system states that lead to a loss connected through LS-1 come to focus. In Figure 7, the focused sequences are shown with solid lines, and dashed lines are possible sequences that were identified by the STPA analysis.

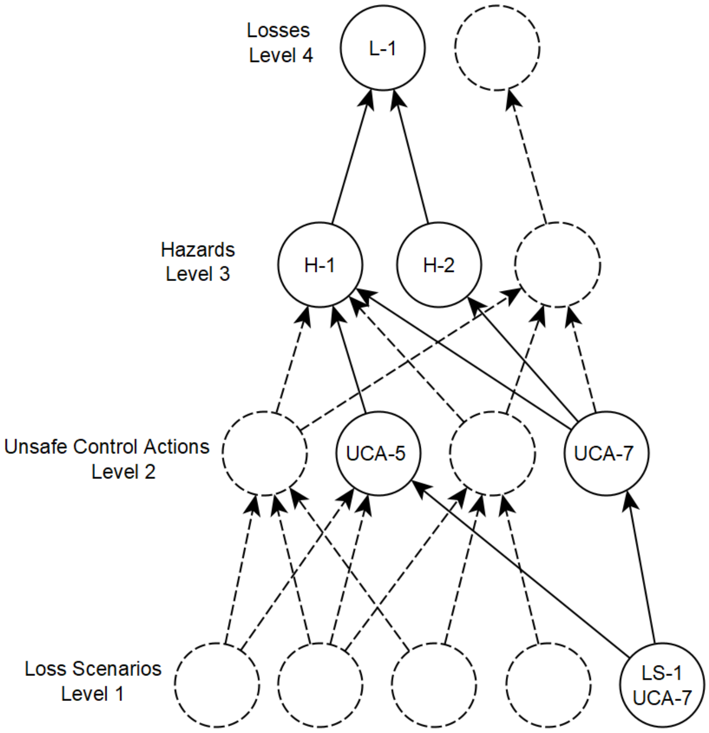


Figure 7. System level RealTSL acyclic diagram and the focused sequences.

The three focused sequences of unsafe system states are shown in Figure 8. For simplicity's sake, only sequence 1 will be calculated for the safety level. The focused sequences are presented with different line types: sequence 1 is shown with solid lines, sequence 2 dashed lines, and sequence 3 with bold solid lines.

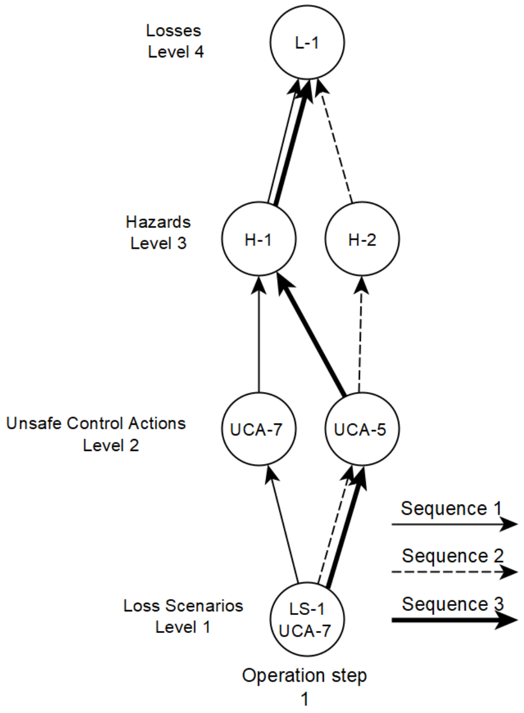


Figure 8. RealTSL acyclic diagram focused sequences.

The progression of sequence 1 is shown through 5 RealTSL acyclic diagrams each for one of the time moments of the scenario presented in Table 6 and are shown in Figure 9. The time remaining until the accident is calculated for every time moment of the scenario using the acyclic diagrams of Figure 9 are presented in Table 7.

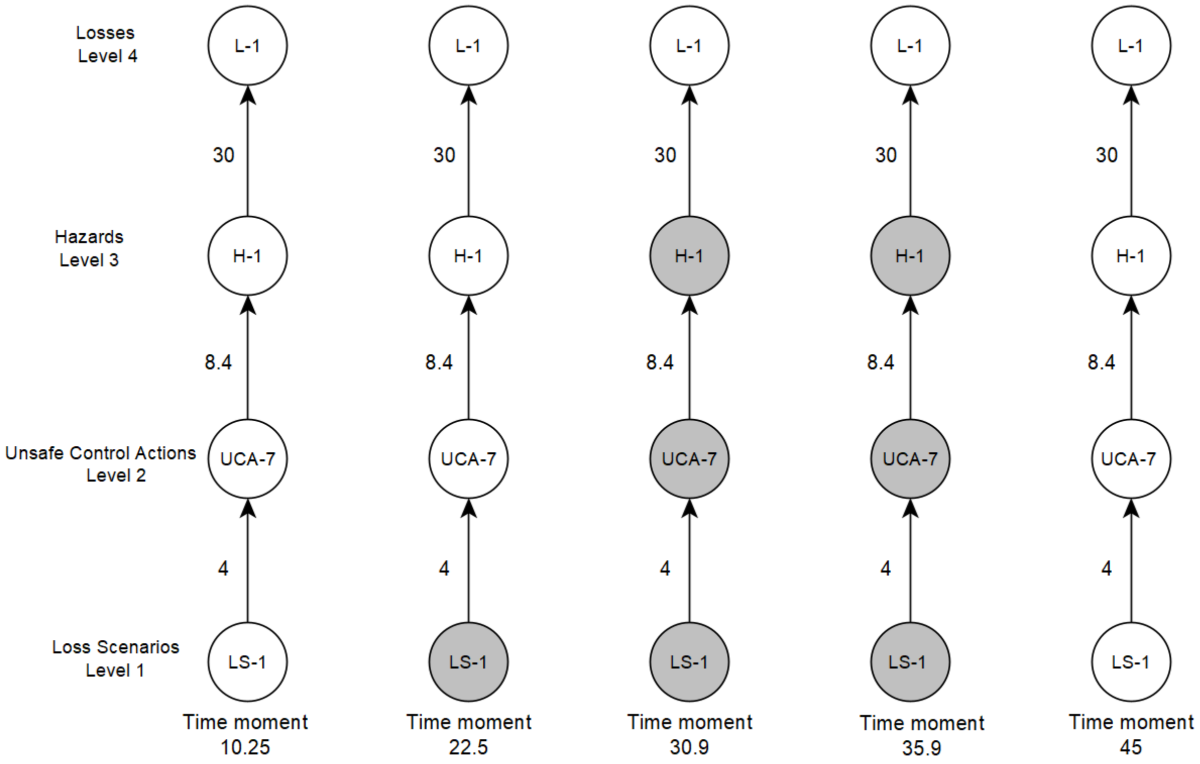


Figure 9. RealTSL acyclic diagrams for the sequence of system state transitions for the time moments of the scenario.

Table 7. Time ranges produced through the system simulation and the resulting time values.

Time moment	Time until accident (seconds)
10.25	42.4
22.5	42.4
30.9	30
35.9	30
45	42.4

The results shown in Table 7 are generated in real-time during the operation of the system. Meaning that the system controller can be aware of time left until an accident occurs (Time until accident (seconds)) at every moment, as well as what sequence of system states is the one that will cause this accident. This information can be used by controllers to decide the actions necessary to maintain the safety of their systems at acceptable levels.

CONCLUSIONS & FUTURE WORK

This paper investigates the self-safety awareness capability, which in the context of COBOT systems, could be of use to enhance their adaptive behavior. The RealTSL methodology was demonstrated in this paper showing how it is possible for a system to assess its own safety level in real-time, utilizing, among other things, its own sensory data. Based on the presented results, it is concluded that simulations can be used successfully to address one limitation of RealTSL, which is the identification of time ranges between unsafe system states. Specifically, this paper proposed an initial set of simulations where the behavior of a COBOT system was recorded in different operational scenarios defined by the STPA analysis to identify the time ranges. Then this paper proposed using these time ranges on the second set of simulations that mimic the execution phase of RealTSL. The benefits of the proposed approach are that (1) the need for domain experts is reduced to only feedback on the creation of the system simulation and the actualization of the hazard analysis, (2) one can assess the sensory equipment available to the system and how to incorporate additional sensory equipment and (3) makes it possible to evaluate the potential use of the outputs of RealTSL before the incorporation of the methodology to a system i.e., if the methodology would be better suited in a subsystem or the entire system or if the system would not be able to utilize the calculated safety level with its available processes. In addition, a novelty of this paper is that the RealTSL methodology was demonstrated not on a theoretical level but in a real system and, more precisely, applied to a critical subsystem of a larger system.

One limitation of the RealTSL methodology is the management of uncertainties that stem from: a) non-existent sensors, b) malfunctioning sensors and c) interpretation of the sensor signals from the controllers, especially when translating a simulation into an actual system. This limitation, however, is planned to be addressed in future works by incorporating fuzzy logic and the safety constraints of STPA into the RealTSL methodology. Furthermore, advances in the technologies of Industry 4.0, such as 5G, IoT and big data, can provide the necessary tools to reduce the effect of this limitation.

Also, the application of RealTSL as a KPI (Key Performance Indicator) on a human-oriented system is planned to be conducted, more specifically on a social network monitoring system for the identification of crisis events, which would have the responsibility of warning the appropriate stakeholders to begin crisis management operations.

ACKNOWLEDGMENTS

We acknowledge support of this work by the project "Risk and Resilience Assessment Center –Prefecture of East Macedonia and Thrace -Greece." (MIS 5047293) which is implemented under the Action "Reinforcement of the Research and Innovation Infrastructure", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

European Union
European Regional Development Fund

HELLENIC REPUBLIC
MINISTRY OF ECONOMY & DEVELOPMENT
SPECIAL SECRETARY FOR ERDF & CF

EPAnEK 2014-2020
OPERATIONAL PROGRAMME
COMPETITIVENESS ENTREPRENEURSHIP INNOVATION

ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Partnership Agreement 2014-2020

Co-financed by Greece and the European Union

REFERENCES

Chatzimichailidou, M.M. and Dokas, I.M., (2016) Introducing RiskSOAP to communicate the distributed situation awareness of a system about safety issues: an application to a robotic system. *Ergonomics*, 59(3), pp.409-422.

Dokas, I.M., Feehan, J. and Imran, S., (2013) EWaSAP: An early warning sign identification approach based on a systemic hazard analysis. *Safety science*, 58, pp.11-26.

Doyle-Kent, M. and Kopacek, P., (2022) Collaborative Robotics Making a Difference in the Global Pandemic. In *Digitizing Production Systems* (pp. 161-169). Springer, Cham

Heinrich, H.W., (1941) Industrial Accident Prevention. A Scientific Approach. Industrial Accident Prevention. A Scientific Approach., (Second Edition).

Leveson, N., (2015) A systems approach to risk management through leading safety indicators. *Reliability engineering & system safety*, 136, pp.17-34.

Leveson, N.G., (2016) Engineering a safer world: Systems thinking applied to safety (p. 560). *The MIT Press*.

Leveson, N.G. and Thomas, J.P., (2018) STPA handbook. Cambridge, MA, USA.

Miller S. (2022) Scissor Lift Model in Simscape Multibody (https://github.com/mathworks/Simscape-Scissor-Lift/releases/tag/22.2.3.5), *GitHub*. Retrieved December 11, 2022.

Reason, J. (1990) Human Error. *Cambridge University Press*, Cambridge. https://doi.org/10.1017/CBO9781139062367

Wilk-Jakubowski, G., Harabin, R. and Ivanov, S., (2022). Robotics in crisis management: A review. *Technology in Society*, p.101935.

Zacharaki, A., Kostavelis, I., Gasteratos, A. and Dokas, I., (2020) Safety bounds in human robot interaction: A survey. *Safety science*, 127, p.104667.

Zeleskidis, A., Dokas, I.M. and Papadopoulos, B.K., (2022) Knowing the safety level of a system in real-time: An extended mathematical model of the STAMP-based RealTSL methodology. *Safety Science*, 152, p.105739.