Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-1 (2021) 534–539

An STPA Safety Analysis Case Study of a Collaborative Robot Application

Adriaensen, A. *-, Pintelon, L. **, Costantino, F. *, Di Gravio, G. *, Patriarca, R.***

** Department of Mechanical and Aerospace Engineering, Sapienza University, Rome, Italy; (arie.adriaensen@uniroma1.it; francesco.costantino@uniroma1.it; giulio.digravio@uniroma1.it; riccardo.patriarca@uniroma1.it)*

*** Centre for Industrial Management/Traffic and Infrastructure, KU Leuven, 3001 Leuven, Belgium (liliane.pintelon@kuleuven.be)*

Abstract: The technology for collaborative robots and the way these technologies are used in current socio-technical work systems are rapidly evolving in industrial applications. In the absence of prescribed safety assessment methods from normative standards, this paper explores the capabilities of an STPA analysis for the socio-technical behaviour of collaborative robot applications. We applied the STPA to a collaborative robot with a heavy-load manipulating arm and gripper, mounted on an AGV-type mobile base. The scope of the analysis is limited to a single AGV mode controller. It explores the systems thinking capabilities of STPA for the safety analysis, from which the principles can be applied to several types of collaborative robot applications.

Keywords: Socio-technical safety analysis, STAMP, collaborative robots, cobots

1. INTRODUCTION

Collaborative robots, in short cobots, are a quickly emerging technology in today's manufacturing and assembly industries. Cobots engage in industrial applications where they operate alongside humans without the presence of a fence to physically interact with humans in a shared workspace (Hentout, Aouache, Maoudj, & Akli, 2019). Although the last decade has delivered a rapid increase in technological and academic developments, this technology has also changed the way socio-technical systems perform by new ways of sharing tasks and workspaces with machines. Several levels of collaborative robot tasks exist, expressed by increasing degrees of engagement between cobots and human in a single simultaneous task. Presently, most industrial cobot applications consist of tasks, where robots and humans still engage in independent tasks (different tasks in one workspace) or sequential tasks (consecutive collaboration on a single task) (Malik & Bilberg, 2019), whereas the full versatility from collaborative robots is to be expected when cobots are able to simultaneously engage in true collaboration on a single task with humans. Adapting to the complexity of increasingly versatile applications poses new challenges to workers' safety. It requires an analysis of joint human-robot behaviour from a systems-thinking perspective. Mutual interactions require a proper understanding of industrial cobots integration in socio-technical systems, whereby deficiencies in human-machine interaction cannot be understood as deficiencies in an absolute sense but depend on how system characteristics shape cognition and collaboration in the actual context of a particular work system (Woods, Dekker, Cook, Johannesen, & Sarter, 2017). In collaborative work systems where humans and machines simultaneously engage in collaboration on a single task, they must engage in joint behaviour through a shared mental image. Research in relation to automation has pointed to the fact that mode error and automation surprises can occur when the operator misinterprets the different meanings from automated functions resulting from multiple device mode settings (Sarter, Woods, & Billings, 1997). Likewise, scholars have described the importance of mode awareness for operators of collaborative robot operations (Gopinath & Johansen, 2019), whereas the cobot in turn should be able to interpret and predict human behaviour (Hentout et al., 2019; Lasota, Fong, & Shah, 2017).

Currently, it is still unclear how to bridge safety requirements for the emerging field of cobot operations to meet hazard and risk analysis from a systems ergonomics, and human factors perspective. Today's normative standards for cobots do not prescribe particular safety assessment methods (Chemweno, Pintelon, & Decre, 2020; Guiochet, Machin, & Waeselynck, 2017). To fill this gap, we propose the usage of the Systems Theoretic Process Analysis (STPA), a hazard analysis technique based on the Systems-Theoretic Accident Model and Processes (STAMP). In the proposed case study, the STPA has been applied for demonstration purposes to a mobile cobot.

2. CASE STUDY DESCRIPTION

The case study used in this paper, is based on an existing cobot demonstrator model that combines a heavy-load manipulating

10.1016/j.ifacol.2021.08.061

arm & gripper, mounted on an AGV-type mobile base (David, André, Kfoury, & Garrec, 2014). This case was chosen because the joint behaviour of the AGV base and manipulating arm movements provides a good example of increasingly complex behaviour from a new generation of cobot applications in addition to the joint complex behaviour of the human-machine ensemble. The STPA analysis provides an extensive understanding of the engineered system and its socio-technical application. Nevertheless, the scope of the analysis has been restricted to an individual subsystem controller, providing in this way a representative, yet manageable unit of analysis for the requirements of the conference. The case study is indeed intended to provide a clear concept on the capabilities of the applied method in an under-investigated domain.

The mobile platform in this case study is able to move fully autonomous between tasks without operator interference; Subsequently, the manipulator arm can be used to grab and dispose heavy objects or workpiece extensions, e.g. a drill workpiece extension. This is performed by a pair of handles which simultaneously act as a hand guiding device to guide the cobot manipulator arm and as an enabling device. When the operator makes positive two-hand contact, the enabling device is automatically activated, which can authorise or restrict certain cobot functions. Contrarily to fully autonomous cobot navigation to move between tasks (Mode 1), the mobile platform remains stationary when the operator handguides the manipulator arm to pick heavy objects or extensions (Mode 2) so that the arm can be manipulated and rotated. Subsequently, once objects have been picked, or workpieces have been installed, they can be transported with the help of the mobile platform (see Figure 1).

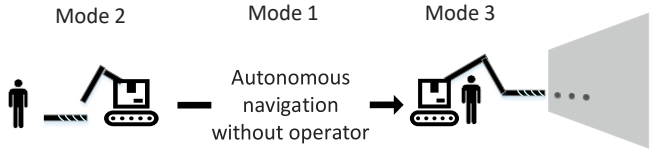

Fig. 1. Three different modes of the mobile platform navigation

While performing the drill function, the behaviour of the mobile platform changes to a third mode in which the platform follows the lateral movements of the operator (Mode 3). This enables to perform precision drilling at recurring distances in a concrete massive structure and ensures that the safety of the operator standing in between the cobot and the concrete structure is assured.

3. METHODOLOGY

The explanation of the STAMP-based methodology is based on the work of Leveson (Leveson, 2011; Leveson & Thomas, 2018). STAMP is the accident causality model underlying the STPA, and providing its theoretical foundation. In STAMP, safety is considered as a system control problem that needs to be managed at several hierarchical layers of a socio-technical system. Multiple interacting and overlapping control loops can be modelled in a hierarchical control structure, as a functional representation of control actions and feedback loops between controllers. Controllers represent agents, systems and subsystems, which are jointly needed to manage a controlled process. Besides control actions and feedback loops, environmental inputs or control inputs that originate from outside the scope of analysis can be added. Figure 1, 2, and 3 (see data and results Section) of our case study can be consulted as examples of hierarchical control structures at different abstraction levels. In STAMP, different models with increasing detail will be provided to proceed from a more conceptual to a concrete design level. The multiple-level approach permits to verify requirements related to multiple subsystems and any potential conflicts between them.

The methodology consists of a number of prescribed steps: step (i) requires describing the system boundaries in line with the scope of the analysis. After the system boundaries have been set, step (ii) defines losses (L). These involve anything of value to stakeholders which need to be prevented from being harmed or damaged. Such negative consequences can arise from failure or from undesirable system interactions. Note that loss of mission is considered as a valid loss in STPA. In the subsequent step (iii) system-level hazards are defined as system states or conditions that will lead to a loss under worst-case conditions. The system-level hazards (H) are linked to the losses for the previous step by their respective numbers (L1, L2, etc.). Thereafter in step (iv) system-level constraint (SC) need to be specified as those system conditions or behaviours that prevent hazards from happening, once they are satisfied. Once more the system-level hazards from the previous steps need to be linked to the current system-level constraint by their respective numbers (H1, H2, etc.).

Once these steps are completed, the hierarchical control structure can be modelled as a series of feedback control loops that constrain the overall system behaviour in step (v). This results in system or subsystem control models as found in figures 2 and 3 from our case study. Based on the different system control constraints, the hierarchical control structure results in its basic elements: controllers, control actions and feedbacks. Controllers define which subsystems are involved in the hierarchically ordered control of the system by enforcing control actions (arrows downwards) on the next lower level (Figure 1 and 2). Feedbacks (arrows upwards) are consequently returned from the lower levels to inform if the controller constraints that came from the higher level are satisfied. Safety control problems can occur among other things when the internal models of humans and technical controllers diverge. Therefore, the control actions and feedbacks always need to be assessed against the internal process models of the different controllers.

A number of steps are required after the hierarchical control structure has been set up. Step (vi) defines the context parameters for the process model of the controllers. The combinations of these parameters serve as the context to form unsafe control actions (UCA) for every control action from a specific controller in step (vii). The STPA requires to verify what should happen for each control action, in the case it is present, in the case it is absent and for timing and duration

problems. These UCAs should be linked to the hazards from step (iii). In the next step (viii) safety constraints (C) are derived from the UCAs, specifying controller behaviours that need to be fulfilled to prevent UCAs from happening. Step (viii) foresees the possibility to identify loss scenarios as a last step, in which the causal factors that can lead to the unsafe control actions and to hazards are described in detail. Loss scenarios basically fall apart in two clusters: (a) reasons for UCAs to occur and (b) reasons why control actions are improperly executed or not executed at all, inevitably resulting in system-level hazards from step (iii). Although the STPA provides a systematic methodology, it also foresees the possibility for iterations of previous steps when additional insights are gained during the analysis.

4. DATA AND RESULTS

In our case, the STPA has been applied to a single sub controller, i.e. the AGV Control Module, indicated by a dotted squared red line in figure 3. This limits the boundaries of this analysis (step i) to the controller that regulates the navigation behaviour of the mobile base of the cobot. The AGV Control Module is particularly interesting because its output is the result from multiple inputs and conditions such as the separation sensor signal, the GPS signal, the operator's mode selection and the safety stop signal as a last barrier defence. The different navigation mode behaviours result from the drive and steer commands. To situate the AGV control module in the socio-technical analysis, we have first provided the hierarchical control structures at higher abstraction in line with the STAMP causation model. At the first level of abstraction, the system is still depicted by one controller per agent (Figure 2), with the high definition goals for each controller.

The definition of losses (step ii) applied to the limited scope of the safe control of the cobot's mobile base and the AGV control module in particular are: 'loss of mission' [L1]; 'loss of structural integrity of cobot' [L2]; 'loss of life or injury to operator or other operators' [L3]; and 'damage to objects or workspace environment' [L4].

Hazards (step iii) related to our scope are defined as 'cobot violates separation minima to surrounding objects or operator(s) [H1]'; 'cobot moves during picking of objects/workpieces [H2]' and; 'cobot does not adapt mobile base navigation to correct operational mode [H3]'. From these system level hazards, we have derived the system level constraints (step iv) in Table 1 which shows how drive and steer commands can satisfy separation constraints in response to providing dynamic safety separation and by providing GPS position for autonomous navigation.

By hand guiding the manipulator arm, the operator automatically powers the enabling device, which notifies the cobot that a human operator is present at the manipulating arm. Human presence, sensed through the enabling device, could for example safeguard against fully autonomous navigation of mode 1, since in this scenario the human operator should keep a distance from the cobot. The enabling device could also be a requiring condition to accept the lateral movement from mode 3 in the drilling function, as this function specifically requires operator hand guiding instructions.

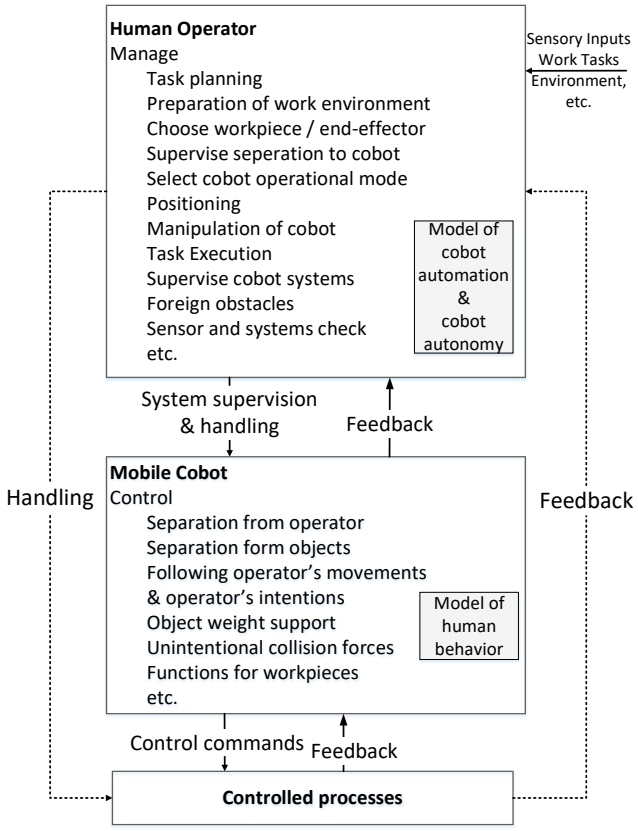

Fig. 2. Hierarchical control structure – highest level of abstraction

Table 1. System Level Constraints

#	Type of constraint	Link to H
SC1	Steer commands should ensure minimum separation with objects/operators in navigation modes 1 and 3	H1
SC2	Drive commands should ensure minimum separation with objects/operators in navigation modes 1 and 3	H2
SC3	When mode 2 is engaged, the cobot mobile platform should stay motionless	H3
SC4	When separation minima are violated, then violation must be detected and a safety stop override will be engaged	H1-H2-H3
SC5	Cobot is able to know its navigation position	H1

The controllers that are directly connected to the AGV control module are reproduced in detail in Figure 3, including the process model (step vi) that define the context parameters involved.

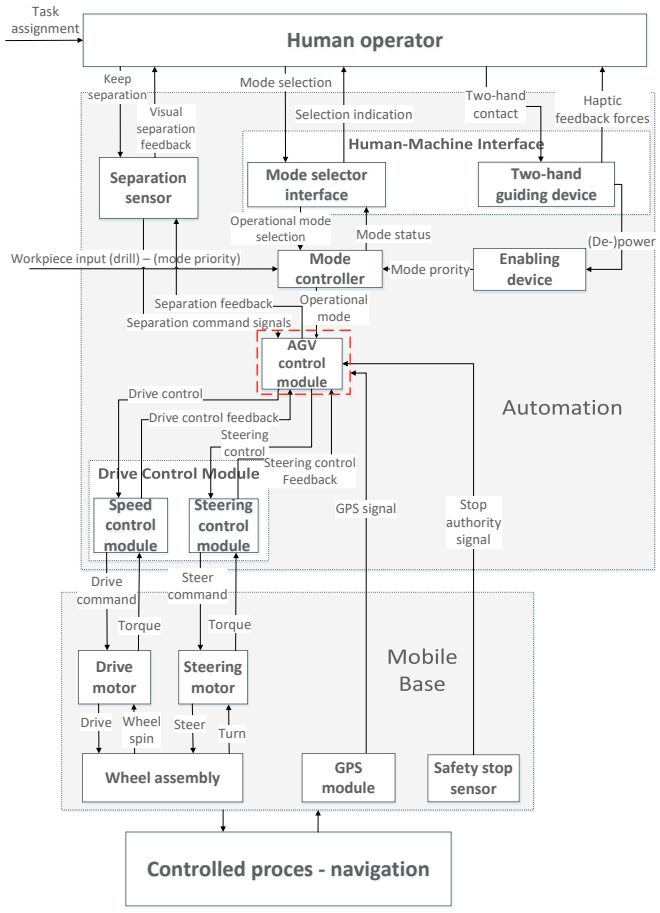

Fig. 3. Hierarchical control structure – mobile base automation structure

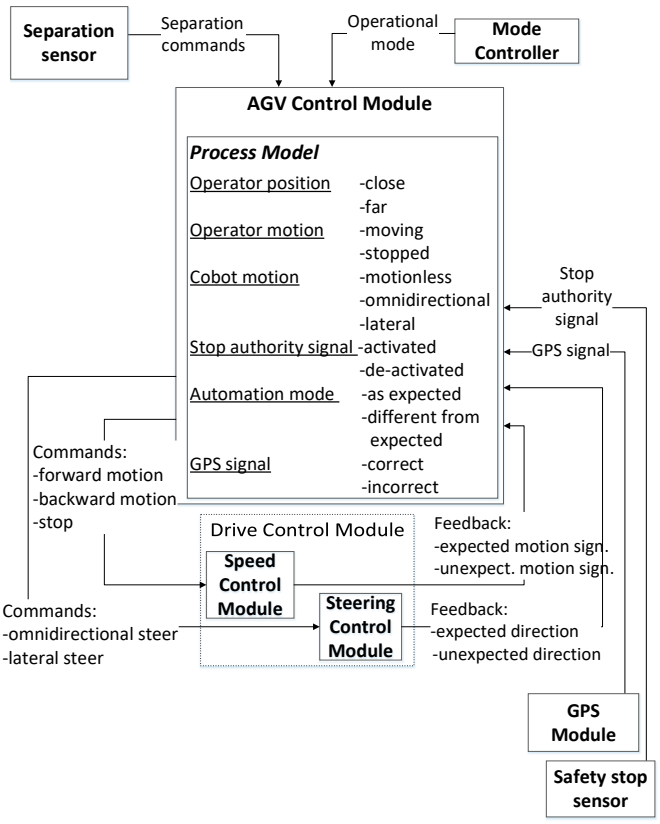

Fig. 4. Focus on AGV control module, including process model.

A number of observations can be derived from Table 2. The combination of parameters in the second line of each control action category, once provided (*), and once not provided (**) display the parameters to be expected in normal operation. In this particular instantiation (the specific configuration of process parameters), objects and operators are far from the cobot, whereby both operator and cobot are in motion, the safety stop signal is not active, the automated navigation mode behaves as expected and the GPS signal is correct. Providing the control actions does not provide a hazard (*), except for the 'STOP control action provided' (§). Although this specific combination of context parameters (§) does not provide a safety threat in terms of separation [L2-4], it creates an unnecessary loss of mission [L1], hence the safety hazard is filled in with both yes [L1] and no [L2-4], depending under the losses considered. Ultimately, the safety stop, as a last defence barrier should not be sacrificed to save the mission and productivity.

In all other cases, providing the control action is each time necessary to keep the system in a safe controlled state under the parameters provided. Contrarily, not providing the intended control action (**), for example due to a mechanical failure of the steer or drive module, creates hazards when these same control actions remain absent. This informs the designer that system failures in relation to these control actions are dependent on component mechanical failure and preferably require a feedback and/or redundancy mechanism.

Table 2. Context Parameters leading to UCAs

Object/ operator position	Operator /cobot motion	Stop sign.[a]	Auto mode[b]	GPS sign.[c]	Hazard
Omnidirectional steer – control action provided					
close	moving /not moving	+	+	+	yes
far	moving	-	+	+	no*
far/close	moving	-	-	+	yes†
far/close	moving	-	+	-	yes
Omnidirectional steer – control action not provided					
close	moving /not moving	+	+	+	no
far	moving	-	+	+	yes**
far/close	moving	-	-	+	yes
far/close	moving	-	+	-	yes
Lateral Steer – control action provided					
close	moving /not moving	+	+	+	yes
far	moving	-	+	+	no*
far/close	moving	-	-	+	yes
far/close	moving	-	+	-	yes
close/close	moving	+	-	-	yes(no)‡
Lateral Steer – control action not provided					
close	moving /not moving	+	+	+	no
far	moving	-	+	+	yes**

far/close	moving	-	-	+	yes
far/close	moving	-	+	-	yes
Forward motion – control action provided					
close	moving /not moving	+	+	+	yes
far	moving	-	+	+	no*
far/close	moving	-	-	+	yes
far/close	moving	-	+	-	yes
Forward motion – control action not provided					
close	moving /not moving	+	+	+	no
far	moving	-	+	+	yes**
far/close	moving	-	-	+	yes
far/close	moving	-	+	-	yes
STOP – control action provided					
close	moving /not moving	+	+	+	yes(no)
far	moving	-	+	+	yes(no)*§
far/close	moving	-	-	+	no ¶
far/close	moving	-	+	-	yes(no)
STOP – control action not provided					
close/close	moving /not moving	+	+	+	yes
far/far	moving	-	+	+	yes**
far/close	moving	-	-	+	yes
far/close	moving	-	+	-	yes

[a] safety stop signal activated (+); de-activated (-)
[b] automated navigat. mode as expected (+); not as expected (-)
[c] GPS signal correct (+); incorrect (-)

The difference in operational losses [L1] versus traditional safety losses [L2-4] is the reason for some fields to have a simultaneous yes-no response. In the last field from 'lateral steer control action – provided'(‡), we have intentionally created a supercritical instantiation with multiple critical parameters. Even if in this combination of parameters, the automation mode does not behave as expected and the GPS signal is incorrect, the safety stop will be activated when the separation with objects or operators nearby is violated due to undesired lateral steering from an unexpected mode. With the safety stop engaged the operator is not able to know that the GPS signal is incorrect, or the mobile base is not engaged in the expected navigation mode. Erratic operation modes or erratic GPS signals that would be apparent under other circumstances will now only be revealed when the safety stop signal is deactivated. The same dual-stop double-negation effect can be observed in the third row of the 'stop – control action provided'(¶), where the mobile base is not in the automated navigation mode which is expected, but the navigation is overridden anyway by providing a non-desired stop control action, for example in mode 1 and 3.

These two instantiations show that the safety stop signal should also be equipped with a feedback mechanism to produce safety stop mode awareness in situations where the system coincidentally 'pretends' to behave normally under erratic conditions. A simple solution would be to install a warning or alarm, but the real benefit from a systems thinking perspective would be to also align apparent behaviour from other sub controllers. To discriminate the safety stop from mode 2 behaviour (motionless mobile base platform during object/work extension picking) it could be advisable to also reflect the safety lock in the manipulator arm response, in which the arm can move freely in mode 2, but remains locked in a safety stop situation. Additionally, any visual alarm or indicator to warn about a safety lock should at least be provided at the hand guiding position where the operator is positioned in mode 2 and 3. Other issues of mode awareness could arise if omnidirectional steering commands would be erroneously allowed in mode 3 where only lateral steering is allowed. During the drilling function, the mobile platform could move unnoticed towards the operator because off-axis omnidirectional movement might be subtle and because the operator drills with her/his back to the cobot platform with the manipulator arm reaching overhead from behind the operator (see Figure 1). This specific instantiation can be found in 'omnidirectional steering – control action provided' (†), where the automation navigation mode is filed as 'not behaving as expected'. Again, a systems-thinking perspective teaches us that any mode changes and warning indication should be apparent at the operator's hand guiding position to align the safety requirements and feedback mechanisms from the human operator controller perspective. From the examination of the different context parameters, we have derived the following controller constraints (step viii) for the AGV control module (see Table 3).

Table 3. Controller constraints

	Controller constraints
C1	Cobot should interrupt omnidirectional/ lateral steer when the safety stop signal is present
C2	Cobot should only provide omnidirectional steer in mode 1, no steer in mode 2 and lateral steer in mode 3
C3	Cobot should immediately stop drive and steer when the GPS signal is incorrect
C4	Cobot should never provide forward/backward motion when the safety stop signal is present
C5	Cobot should never provide forward/backward motion in mode 2
C6	Cobot should provide forward/backward drive commands in mode 1 and 3 when separation with objects and operators is provided, safety signal is de-activated and GPS signal is correct.
C7	Unambiguous feedback signal should be provided for all modes and safety stop.

5. DISCUSSION & CONCLUSION

The STPA has previously demonstrated its capabilities to analyse the joint performance in other human-technical systems where automation is involved (Abdulkhaleq et al., 2017; Chatzimichailidou, Karanikas, & Plioutsias, 2017).

The STPA provides a step-by-step analysis to define systemic issues with the help of hierarchical control structures. We have provided an example of a single controller, which we started

to combine exploratively with the safety requirements for the manipulator arm and the human operator. Thereby the STPA provides a way of examining joint complex behaviour, by deconstructing the requirements for different sub controllers and by identifying potential conflicts. A full STPA analysis of this system would have to be performed on multiple sub controllers, to fully understand the emergent behaviour from the human operator, the cobot's manipulator and the mobile platform behaviour (Patriarca et al., 2021). Especially in the case where all context parameters should be checked for every single controller, a formal model checking tool like UPAAL would be required, and has previously been combined with the STPA method (Yang, Karashima, Okano, & Ogata, 2019). Industrial standards in relation to cobots like ISO 10218 and ISO/TS 15066 list a number of hazards and requirements, but do not offer pre-determined formal verification methods. The STPA approach allows to incorporate requirements from existing industrial standards, which can be used to cross-validate hazards and constraints in an iterative way.

An activated safety stop signal always provides a loss of mission [L1], even if just temporarily, whereas any scenario that involved an unexpected automation mode or incorrect GPS signal, created a safety hazard [L2-4] in our context scenarios, except in the example above (¶). Hence, by discriminating between the type of losses, the STPA can additionally provide a trade-off analysis between safety hazards and operational hazards. Alternatively, the requirements for the remaining sub controllers could be optimized to prevent the safety stop from engaging when all other context parameters are safe to prevent loss of mission by a safety stop engagement. The STPA analysis is thereby capable to extend the safety analysis to trade-offs between traditional safety objectives and mission efficiency.

Performing a full STPA requires substantial time and personnel resources, but we argue that this is the case for any comprehensive safety analysis. Especially in an emerging field with a lack of historical data, this method could be useful for the identification of safety issues early in the design phase. The approach has been applied to a number of other safety fields and it could be generically applied to a variety of different cobot applications. Specifically, with the observed increase in complexity of cobot technology and increased symbiotic tasks between humans and robots, the STPA provides a promising possibility to look at the joint behaviour of human-machine collaborative tasks from a systemic perspective.

6. REFERENCES

Abdulkhaleq, A., Lammering, D., Wagner, S., Röder, J., Balbierer, N., Ramsauer, L., … Boehmert, H. (2017). A Systematic Approach Based on STPA for Developing a Dependable Architecture for Fully Automated Driving Vehicles. *Procedia Engineering*, *179*, 41–51. https://doi.org/10.1016/j.proeng.2017.03.094

Chatzimichailidou, M. M., Karanikas, N., & Plioutsias, A. (2017). Application of STPA on Small Drone Operations: A Benchmarking Approach. *Procedia Engineering*, *179*, 13–22. https://doi.org/10.1016/j.proeng.2017.03.091

Chemweno, P., Pintelon, L., & Decre, W. (2020). Orienting safety assurance with outcomes of hazard analysis and risk assessment: A review of the ISO 15066 standard for collaborative robot systems. *Safety Science*, *129*. https://doi.org/10.1016/j.ssci.2020.104832

David, O., André, S., Kfoury, F., & Garrec, P. (2014). Cobomanip: A new generation of intelligent assist device. *Proceedings for the Joint Conference of ISR 2014 - 45th International Symposium on Robotics and Robotik 2014 - 8th German Conference on Robotics, ISR/ROBOTIK 2014*, 93–100.

Gopinath, V., & Johansen, K. (2019). Understanding situational and mode awareness for safe human-robot collaboration: case studies on assembly applications. *Production Engineering*, *13*(1), 1–9. https://doi.org/10.1007/s11740-018-0868-2

Guiochet, J., Machin, M., & Waeselynck, H. (2017). Safety-critical advanced robots: A survey. *Robotics and Autonomous Systems*, *94*, 43–52. https://doi.org/10.1016/j.robot.2017.04.004

Hentout, A., Aouache, M., Maoudj, A., & Akli, I. (2019). Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017. *Advanced Robotics*, *33*(15–16), 764–799. https://doi.org/10.1080/01691864.2019.1636714

Lasota, P. A., Fong, T., & Shah, J. A. (2017). A Survey of Methods for Safe Human-Robot Interaction. *Foundations and Trends in Robotics*, *5*(3), 261–349. https://doi.org/10.1561/2300000052

Leveson, N. G. (2011). *Engineering a safer world: systems thinking applied to safety*. (J. Moses, de N. R., M. Heitor, G. Morgan, E. Paté-Cornell, & W. Rouse, Eds.). Cambrdige, MA: Massachusetts Institute of Technology Press.

Leveson, N. G., & Thomas, J. P. (2018). *STPA Handbook*. https://doi.org/10.2143/JECS.64.3.2961411

Malik, A. A., & Bilberg, A. (2019). Developing a reference model for human–robot interaction. *International Journal on Interactive Design and Manufacturing*, *13*(4), 1541–1547. https://doi.org/10.1007/s12008-019-00591-6

Patriarca, R., Falegnami, A., Costantino, F., Di Gravio, G., De Nicola, A., & Villani, M. L. (2021). WAx: An integrated conceptual framework for the analysis of cyber-socio-technical systems. *Safety Science*, *136*, 105142. https://doi.org/https://doi.org/10.1016/j.ssci.2020.105142

Sarter, N. B., Woods, D. D., & Billings, C. E. (1997). Automation Surprises. In *Handbook of Human Factors & Ergonomics*.

Woods, D., Dekker, S., Cook, R., Johannesen, L., & Sarter, N. (2017). *Behind Human Error*. CRC Press.

Yang, P., Karashima, R., Okano, K., & Ogata, S. (2019). Automated inspection method for an STAMP/STPA - Fallen Barrier Trap at Railroad Crossing - Fallen B. *Procedia Computer Science*, *159*, 1165–1174. https://doi.org/10.1016/j.procs.2019.09.285