Proceedings of the Estonian Academy of Sciences,
2021, **70**, 4, 383–391
https://doi.org/10.3176/proc.2021.4.03
Available online at www.eap.ee/proceedings

MANUFACTURING
ENGINEERING

Multi-layer cyber-physical control method for mobile robot safety systems

Heiko Pikner[a*], Raivo Sell[b,a], Jüri Majak[a] and Kristo Karjust[a]

[a] Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Tallinn, Estonia
[b] FinEst Twins Smart City Center of Excellence, Tallinn University of Technology, Tallinn, Estonia

Received 17 June 2021, accepted 19 July 2021, available online 1 November 2021

Abstract. Self-driving vehicles and mobile robots are used more and more in public transportation and industrial companies. Multiple experimental platforms, which can be operated in an urban or industrial environment, have been developed recently. The key development for robust and safe control of the robot's operation relies on the low-level cyber-physical system (CPS). CPS is composed of a collection of tightly integrated computational (cyber) units that are communicating with the physical world. CPS integrates computation and communication aspects with control and monitoring techniques. In this paper, the scientific goals underscore the analysis of the existing state-of-the-art solutions to increase the security, safety, and reliability of the multiple experimental platforms. Extra attention is paid to risk evaluation.

Key words: cyber-physical system, mobile robot, safety controller, crash detection, risk evaluation model.

1. INTRODUCTION

Digitalization and robotization are a constant challenge for the industry to make production more effective, robust, and reliable. The fourth industrial revolution, called Industry 4.0, has brought smart technology into focus where robots, production units, and services are interconnected. Despite the interconnection of hardware, the integration of the robots with the manufacturing engineering systems should be developed and tested [1]. Thus, the standard automated guided vehicles (AGV) cannot fulfil more complex tasks and must be upgraded to more flexible mobile robots that can reroute on-demand and intercommunicate with other machines. Several types of research have already investigated how to integrate autonomous robots into the Industry 4.0 environment [2]. Modularization in mobile robotics is essential to offer flexible reconfiguration, efficient design, and reduce the development and implementation time. The proper architecture and modular concept must be taken into account already in the early design stage and proper methodologies are proposed for mechatronic system design. Christophe et. al. has proposed OPAS [3] to synthesize conceptual design solutions in early design stages [4]. Early design and a simulation toolkit for mobile robot platforms [5] relying on these methodologies help to design modular universal mobile robots for the industrial logistic environment, inside the manufacturing area. From the development stage and later from the implementation side, the inside manufacturing area adds additional limitation parameters for the development stage such as communication limitations, localization, shorter distances and space, limitations of safety, etc.

The lower-level cyber-physical solution is a critical part of controlling modular mobile robots using modules that interact with the physical world. These modules are mainly divided into sensors, actuators, and computation units. In addition, inside cyber-physical solutions the modules are usually distributed, and therefore communication is an important part of the system. The separate constituent parts (sensors, actuators, software, etc.) of the cyber-physical solution collaborate to create some global

* Corresponding author, heiko.pikner@taltech.ee

behaviour [6]. Nowadays it is common to get real-time data from the production area and from the mobile robot to make real-time analysis and management tasks [7].

Mobile cyber-physical systems have significant computational resources to maintain localization, obstacle detection, safety functions, and path following. Computational resources can be divided into two different categories such as artificial intelligence (AI) based on high-level decision-making and lower-level control logic. AI and high-level decision making are based on some special computers to run robotic operating systems (ROS). In many cases, it is a regular PC. The low-level control logic is near or inside the actuator or sensor modules. It handles a regulation for actuators and performs the first information processing for the information received from sensors. Moreover, it controls and forwards information between the modules.

The communication inside the mobile cyber-physical solutions is usually based on the CAN (Controller Area Network) and the Ethernet networks [8]. CAN is an existing multi-master broadcast serial bus communication protocol for connecting embedded electronic control units (ECUs) in automotive applications. The ECU's functionality ranges from small tasks, such as changing a light, to more advanced functionality such as steering and drivetrain systems.

FlexRay is a newly introduced communication protocol for an automotive control system developed to fulfil the increasing demands for higher safety and data rate [9]. Furthermore, multiple wireless communication mechanisms, such as WiFi, 4G, EDGE, Bluetooth, used to interconnect several devices, connect robots or connect robots to the Internet. Currently, new developments are going to use and test 5G in similar conditions.

A mobile cyber-physical system may have multiple sensory input devices. The most common vision and 3D imaging systems' sensors are lidars, radars, and cameras [10]. There may be sensors for localization, for example, GPS and inertial sensors, also sensors that can detect the presence of nearby objects without any physical contact. Such sensors are ultrasound or infrared distance sensors. Different sensors may be measuring internal parameters – for example, battery current and motor speed. Common actuators are propulsion and steering systems for wheeled robots. There may be lifting mechanisms for cargo, robot arms, and other moving devices. Some robots have legs with electrical or hydraulic drive mechanisms. All these modules are interconnected over a communication network and controlled computational resources.

Public transportation and logistics inside industrial companies include different types of robots, drones, and autonomous vehicles. Smaller robots that can be used inside warehouses and factory floors are under consideration. Different robots are presented in [6] and [11], according to the producers, and are selected to analyse the onboard cyber-physical system development. Robots are chosen based on the kind of lifting or goods carrying mechanism, which allows them to move the payload from one location to another. These robots usually have good navigation capabilities inside the rooms, which is crucial in narrow corridors and in cases where limited space is available.

The robots introduced in previous works have quite different software, sensors, and control systems. There is a trend to use open-source solutions for different kinds of mobile robot control software. A popular choice is a robot operating system with a specific software stack or add-on modules [12]. There are few such robots, for example, BoxBot, Robotnik, MiR, and Freight. Others are using proprietary software, which may be a disadvantage, because it may be difficult to add new functionality of changing tasks on an ongoing basis. High-quality sensors such as 3D lidar (light detection and ranging) are not very common. Most robots use 2D lidars, cameras, and ultrasound sensors. This set and functionality of sensors may limit the robot's performance in new or more complex environments and in the manufacturing environments where the changes very often take place.

The long-term goal is to create a situation, where multiple robots work together for one specific task or serve the manufacturing process. For example, logistics robots serve the production stations by supplying the raw material and bringing production results into the warehouse. This kind of multi-robot environment requires an additional layer on top of the system, which controls the high-level goals and guides a single robot's individual goal. Such a robot swarm can handle very complex tasks but needs complex control algorithms. Individual robots can also be partly remotely controlled by humans [13], which makes the initial implementation easier. Swarm robotics is trying to follow biological systems and theoretical developments have been conducted to apply a bio-inspired approach for robot swarm in smart factories [14].

Despite emerging developments in the area of mobile robot systems, fully autonomous driving systems, as a rule, are not yet allowed, at least in public traffic [15]. Safety is a key issue of any fully or partially autonomous driving system due to the need to consider/understand several complex factors such as environment, traffic, hardware and software systems' reliability, information availability, cyber hacking, etc.

This paper focuses on a practical approach and implementation of a cyber-physical system on mobile robots and autonomous vehicles. The safety issues are studied in the context of the considered problems. The risks and their evaluation criteria are developed for a particular class of problems.

2. CONCEPT OF MULTI-LAYER CYBER-PHYSICAL ARCHITECTURE

Numerical modelling and design optimization of systems, networks, and devices are essential parts to increase the safety of mobile cyber-physical systems. At a low level, it is difficult to avoid situations where the artificial intelligence based on the high-level decision-making layer has decided to make the wrong manoeuvres or crash the robot. The lower-level control logic can detect when a crash is happening and execute commands to stop the robot as fast as possible. Security and safety-critical electronic control unit designs depend on several factors. ECUs can be classified based on safety-security characteristics [16]. The ECU's components should comply with international automotive application standards, for example, AEC-Q100 – a standard for packaged integrated circuits. Automotive microcontrollers also offer additional safety mechanisms.

Mainly it is important to keep the microcontroller executing the program and send an alert if a fault occurs. As a technical solution, a watchdog or co-microcontroller should be monitoring the main microcontroller. If the main microcontroller is not running properly, then restart may be one of the simplest options. Also, if the main microcontroller has a status message sent in the communication network in a specified interval and if this message disappears, another ECU can execute commands to stop the robot. A more complicated solution is to use a separate communication network for safety-related co-controllers, located inside the ECU. In this way, when the main communication network is damaged, safety-related controllers can still communicate with each other and stop the robot. The solution is intended for use cases when stopping the robot requires the cooperation of several ECUs. An error alert should also be sent when the main microcontroller inside the ECU was unable to achieve a required actuator correction according to the feedback in the given time interval. In this case, there may be a fault in the mechanics, a fault in the power unit, or a fault in the actuator motor. It is not very reasonable to measure the PWM or other direct control signals generated by the main microcontroller with the co-controllers, because if the power unit or motor is already burned out, the PWM signal is measured as good but the system will not work as intended. Thus, it is rather more reasonable to read and compare the values of the feedback sensors with the setpoints.

Modern vehicles are equipped with an occupant restraint system which is controlled by the dedicated airbag control module [17]. Self-driving robots may not have people inside at all or they are shuttle-type minibuses where airbags are not required. In this case, a standard airbag control module is not required, but crash detection may be a good idea to stop the robot immediately. In this case, a separate security ECU is a location for control electronics for collision sensors and crash battery disconnect fuse or pyro fuse. It is advisable to disconnect the battery and also the traction battery if it exists because the electrical wiring may be damaged as a result of the collision. Moreover, a separate safety controller can handle communication for safety buttons via a radio link or safety buttons over the Internet meant to be used inside a remote control centre. If there is a separate data communication network for the security co-controllers, then the security ECU can analyse information and coordinate the stop of the robot.

In some cases, ECUs can be made in such a way that the robot must be drivable for at least sufficient time to stop itself safely if a fault occurs. The microcontroller, the power unit, and even the actuators are duplicated (two independent motors on one shaft) or made in such a way that their reliability is guaranteed. In addition to the ECUs, the communication network can also be duplicated. If one network goes down, for example, due to a short circuit, the other network will continue to work.

One criterion for choosing a solution may be the speed of the robot. If the robot speed is up to 20 km/h, the strategy is to brake as fast as possible:

1. Critical microcontrollers should be kept operational, and their faults monitored.
2. As regards design, vital ECUs can be supplemented with a coprocessor that monitors when something happens to the main processor. A data network for coprocessors may be added. If the basic data network is unusable, an additional CAN network is used to coordinate the robot stop.

If the robot speed is higher than 20 km/h, the strategy is that if something happens, the robot must be drivable:

1. One of the critical controllers must be kept operational in the event of any fault.
2. In the event of a fault, the robot must be fully controllable to drive and stop in a safe place.
3. As regards design, vital systems should be duplicated, or reliability is guaranteed in another way.

We have developed a low-level control architecture of the CPS first for a self-driving last-mile bus called Taltech iseAuto. Then based on the same architecture, a new small-scale logistic robot (BoxBot) has been developed to move boxes in rather tight spaces. During the tests, it was determined that the robot is capable of transporting packages without human interference. It is not trivial to implement robot-based logistics as many specific issues need to be solved and addressed. Corridors and transportation spaces are usually designed for humans and the robot must navigate between obstacles that may change at any time. The tasks of the robot may also change.

TalTech iseAuto has been designed to be a minibus that is going to operate primarily on the territory of the university campus, therefore the speed of the robot was limited to 20 km/h. The architecture of the robot's CPS is divided into layers as described in Fig. 1. The AI and high-level decision-making layer make autonomous driving decisions based on the sensor's input layer. The robot speed and direction commands are sent to the actuators layer that has a mission-critical functionality to take care of the robot's actual control. Control logic is divided into two layers – the master controller layer and the drive controller layer. The main task of the master controller is to act as a central gateway between all the nodes. The most sophisticated low-level functionality is integrated into the drive controller which controls the robot movement and steering. For safety reasons, a separate safety controller has been developed to stop the robot when some fault is detected. The communication layer runs on separate CAN networks and Ethernet [18].

The small-scale logistic robot BoxBot, developed in our research group, was designed to operate inside warehouses as well as to transport materials from and to the production units and empty boxes between washing stations and production units. The control architecture of the logistic robot is similar to iseAuto and is divided into the same layers as described in Fig. 2. In addition, it is adapted to fit a much smaller space. The upper layer provides input to the ROS high-level control system. The AI and drive algorithm layer is based on the NVIDIA Jetson AGX Xavier developer kit. The logistic robot commands are sent to the low-level control layer that has a

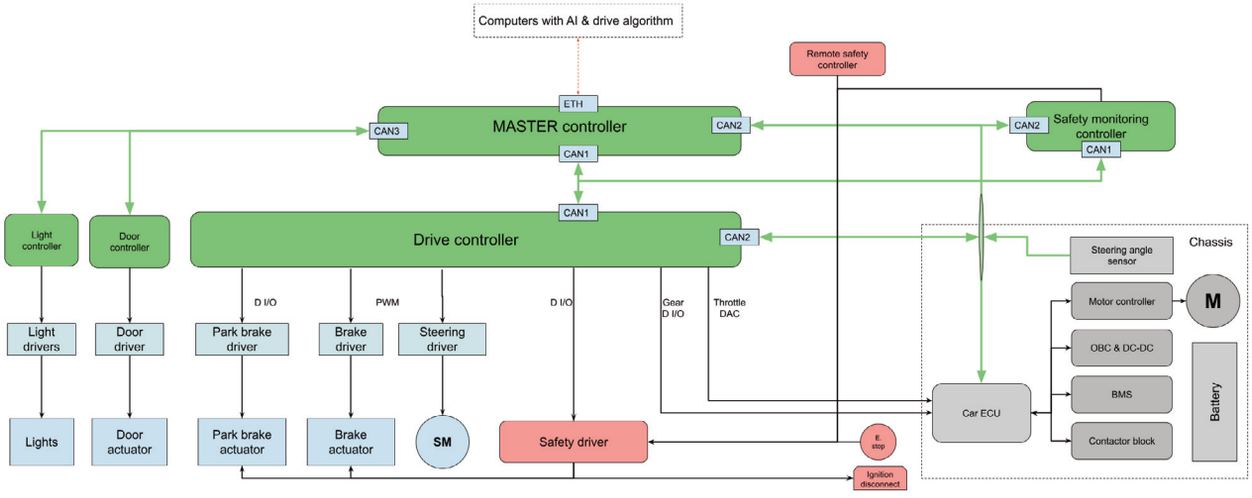

Fig. 1. Low-level control solution for TalTech iseAuto. ETH – Ethernet; CAN – Controller Area Network; D I/O – digital input/output; PWM – pulse-width modulation; DAC – digital-to-analog converter; OBC – onboard charger; DC-DC – direct current to direct current converter; SM – steering motor; E. stop – emergency stop; ECU – electric control unit; BMS – battery management system.

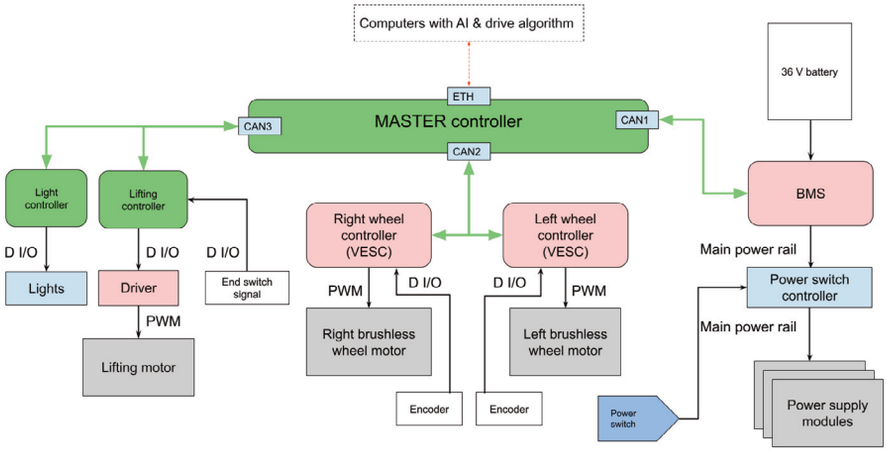

Fig. 2. Low-level control solution for TalTech BoxBot.

mission-critical functionality to take care of the robot's movement control. The central unit for this layer is the master controller. The difference is that the drive controller which is originally meant to control the iseAuto platform and steering is not needed and replaced with a simpler motor controller for two-wheel hub motors. For lifting mechanism and other systems, dedicated controllers are included.

Both CPSs have three layers of control: high-level running AI and drive algorithms, mid-level master controller mediating and prioritizing the messages, and low-level control for actuators. A separate safety controller has been developed for the iseAuto platform, but it is not suitable for other applications. Therefore, new safety functions are built into the system and tightly integrated with all layers of CPS.

TalTech BoxBot has new power management integrated with safety functions as shown in Fig. 3. The power management board checks over the communication interface whether there are any problems with the system. Furthermore, the power management board has interfaces for physical emergency switches and a radio interface for remote emergency switches. If a fault occurs, the power board can turn off all or some power buses. Also, if upon starting the robot there is a short circuit in some of the power buses, then the launching is stopped and the error triggered.

3. RISK EVALUATION MODEL

Mobile robots and automated driving systems obviously have various risks due to multiple complex issues that need to be covered. The public expectancy for self-driving vehicles is zero traffic accidents, although the technology and other factors are still under the heavy development stage. Thus, risk analysis is one of the key factors in the development of safe mobile robots and self-driving vehicles.

The risk evaluation model proposed is based on the combined use of the fuzzy analytic hierarchy process (FAHP) and similarity to the ideal solution (TOPSIS). First, the criteria and risks are introduced for particular mobile robot types that are considered. Based on literature review [7,19,20] and five-member expert group opinion (industry + academia), the following six criteria are proposed:

Mission (C1): Criterion refers to the reliability of the system. Situations, where the robot is unable to perform the tasks assigned to it, may lead to cessation of production or interruption of the transportation of passengers and goods.

Cybersecurity (C2): Criterion refers to all kinds of hacking of automated systems. Remote-control attacks are one of the prioritized security threats. Moreover, autonomous passenger transport carries the risk of the passenger

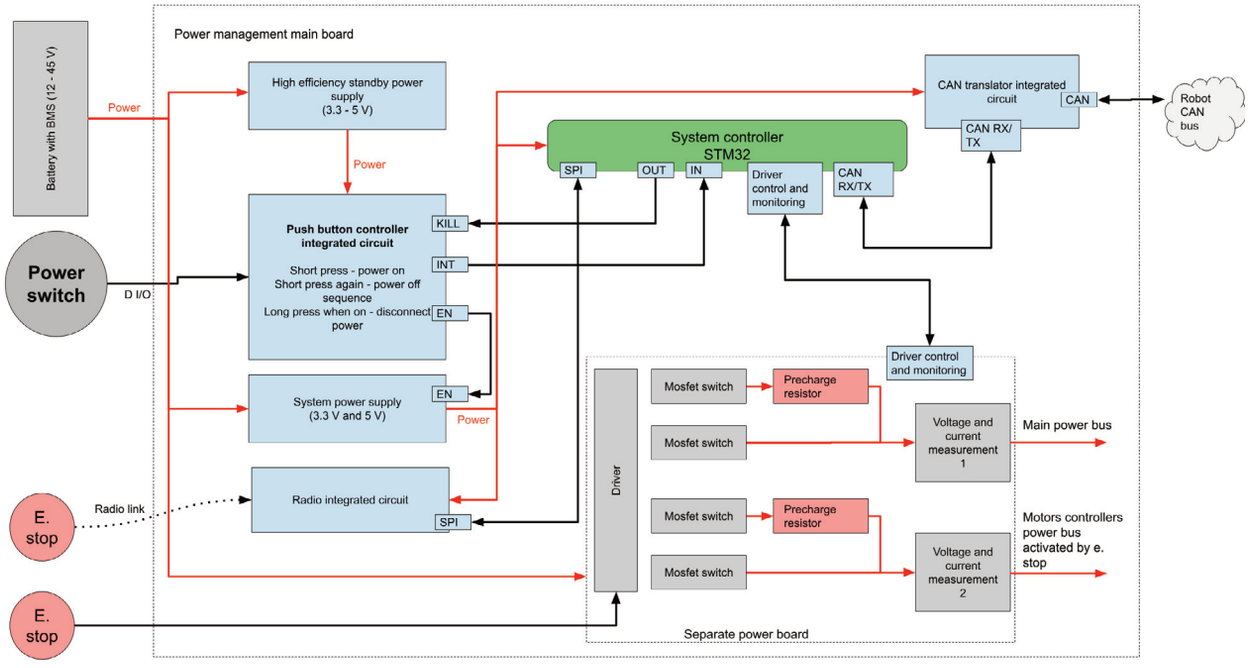

Fig. 3. New power management and safety solution for TalTech BoxBot. SPI – serial peripheral interface; INT – interrupt; EN – enable; RX – receive; TX – transmit.

gaining access to the robot's internal network, or computer viruses find their way into the system.

Malfunction of AV component (C3): The components of an autonomous vehicle or robot may also fail, which is a risk of accidents and further damage as well.

The sensor system (C4): Criterion refers to the reliability of the sensors. The sensors may stop working due to mechanical breakage or electrical failure. The operation of the sensors can also maliciously interfere with lasers, radio jammers, and other devices.

The communication link (C5): Criterion refers to the reliability of the communication links. The components of the communication link may fail due to hardware or software issues, as well as hacking. In addition, loss of communication may lead to accidents.

Environmental factors (C6): Criterion refers to the driving environment factors including weather conditions and other essential factors for prioritizing the risk in a driverless vehicle.

The risks involved in cyber-physical architecture are identified by the same expert group and literature analysis as follows:

Mechanical failure risk (A1): Risk refers to the failure of the mechanical components due to normal wear and tear, manufacturing or design errors, corrosion, vandalism, mishandling, or an accident.

Electrical failure (A2): Risk refers to the failure of the electrical components. Electrical components can be divided roughly into ECUs, wiring harness, batteries, sensors, and electromechanical actuators. Failure may happen due to manufacturing or design errors, corrosion, short circuit, overheating or firmware failure. These types of faults can lead to greater damage, such as fire or fatal accidents.

Information shortage (A3): Risk refers to the failure of lost communication. As the vehicle or robot should operate autonomously, this type of error does not directly cause major damage. However, if an attempt is made to stop or drive the vehicle due to a previous malfunction, the information shortage may result in an accident.

Autonomous driving software failure (A4): Risk refers to the failure of autonomous driving software. This is one of the most prioritized security threats which could lead to an accident. This type of failure is difficult to detect and correct from the lower side and requires urgent intervention by the remote-control centre.

Low-level software failure (A5): Risk refers to the low-level software failure due to mainly programming or design errors. In addition, cyber hacking is a possible cause. Risk is controllable by the right design choices of cyber-physical architecture. However, low-level software failure occurrence is dangerous as the actuators can move unpredictably, and the vehicle or robot may accelerate, causing the crash. The actuators and the electrical system may also get damaged due to overload or due to signals in the wrong order.

Communication bandwidth shortage (A6): Risk refers to the communication bandwidth shortage. As the vehicle or robot should operate autonomously, this type of error does not directly cause major damage. However, if an attempt is made to stop or drive the vehicle due to a previous malfunction, the communication bandwidth shortage may result in an accident. The risk also indicates that the remote-control centre may lose the vehicle overview info and the remote-control option. An accident or vandalism may occur if bad circumstances coincide. In addition, a robot or vehicle downtime may occur if it does not respond to the order to move to a new location to collect products or pick up passengers.

Cyber hacking (A7): Risk involved with deliberate exploitation of automated vehicle systems by unauthorized entities. The target of the attack can vary, ranging from the attack on software to manage the system to even the physical attack on the vehicle's hardware. Remote-control attacks are one of the prioritized security threats, which could be considered the most dangerous type of attack.

Interruption of uplink (A8): Risk involved with interruption of uplink. As the vehicle or robot should operate autonomously, this type of error does not directly cause major damage, but the remote-control centre may lose the vehicle overview info and the remote-control option. An accident or vandalism may happen if bad circumstances coincide. In addition, a robot or vehicle downtime may also occur if it does not respond to the order to move to a new location to collect products or pick up passengers.

A drastic change of environment (A9): Risk involved with a drastic change of environment. In addition, snow may accumulate on the sensor's surfaces, and heavy rain or snow may disturb the operation of the sensors. An inside environment may contain dust, food, and other substances which may cover sensors or block mechanical actuators. An accident may occur if bad circumstances coincide. A large drop in temperature may cause an electrical system failure.

Loss of localization (A10): Risk refers to loss of localization. In this case, the robot or vehicle does not know where it is located. An accident may occur if a robot or vehicle tries to move. With appropriate design choices for autonomous driving software, this risk should be minimized. In addition, if the robot is unable to restore its localization, the remote-control centre should take control.

Based on the above-defined criteria and risks, the decision hierarchy tree for the considered mobile robot systems can be established as Fig. 4.

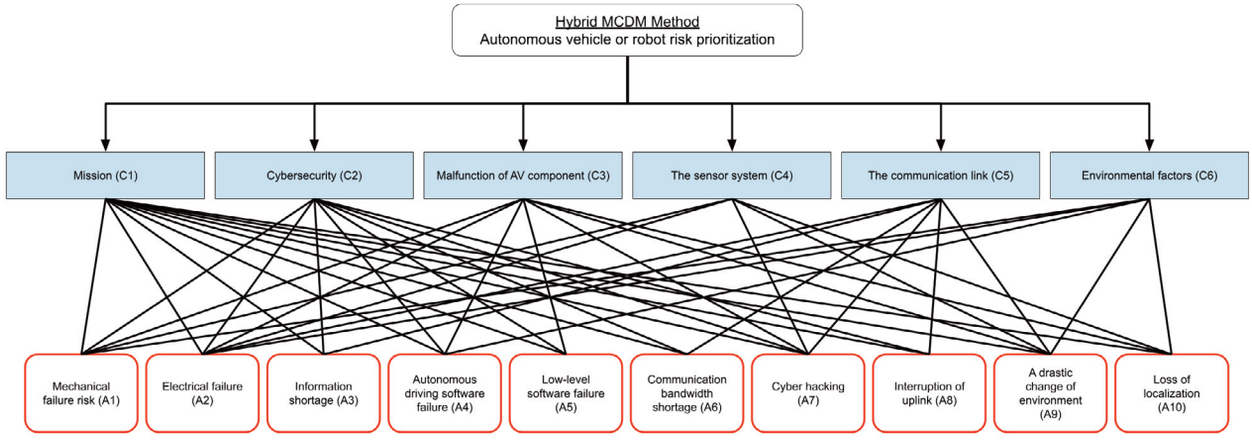

Fig. 4. Decision hierarchy of the risk evaluation problem for mobile robot systems.

Since the current study is focused on the development of the risk evaluation model, the following implementation tasks are described briefly.

3.1. Criteria prioritization using Fuzzy AHP

Fuzzy analytical hierarchy process (AHP) combines traditional AHP methods with the fuzzy set theory. Herein, the triangular fuzzy number (TFN) is utilized for the fuzzy AHP and TOPSIS. The basic steps of the Fuzzy AHP can be summarized as:

Step 1. Data collection. The pair-wise comparison matrix criteria vs criteria are filled by decision-makers. Most commonly the linguistic variables are provided for decision-makers to simplify their evaluation of the importance of the criteria.

Step 2. The linguistic scales are transferred to triangular fuzzy numbers.

Step 3. The individual evaluation matrices are aggregated by applying a fuzzy geometric mean.

Step 4. The aggregated comparison values are computed for each row of the evaluation matrix.

Step 5. The triangular fuzzy weights are calculated.

Step 6. The triangular fuzzy weights can be defuzzified to a crisp number.

Step 7. The crisp weights are normalized.

Step 8. The criteria are prioritized based on normalized crisp weights.

Step 9. The consistency ratio (CR) of the defuzzified matrix is calculated and validated (should be <0.1).

As a result, one obtains the ranked criteria.

3.2. Risks prioritization using Fuzzy TOPSIS

In the current study, the Fuzzy TOPSIS is used for ranking the risks. The basic steps of the Fuzzy TOPSIS have a certain similarity with the above-listed steps for the Fuzzy AHP and can be pointed out as:

Step 1. Data collection. The pair-wise comparison risk vs criteria are filled by decision-makers.

Step 2. The linguistic scales are transferred to triangular fuzzy numbers.

Step 3. The individual evaluation matrices are aggregated by applying a fuzzy arithmetic mean.

Step 4. The aggregated fuzzy decision matrix is normalized.

Step 5. The weights of the criteria obtained by applying the Fuzzy AHP are utilized to compute the weighted normalized decision matrix.

Step 6. The Fuzzy positive and negative ideal solutions (PIS and NIS) are determined.

Step 7. The distances of each risk to positive and negative ideal solutions are computed.

Step 8. The similarities to an ideal solution can be computed.

Step 9. The risks are ranked based on the values of the similarities.

The ranked risks provide useful information to the developers of mobile robot systems.

4. DISCUSSION

The first objective was to establish a multi-model cyber-physical architecture for a new industrial mobile robot BoxBot 2, which can be used later to build different types of smaller or bigger scale autonomous robots. The long-term goal is to create a situation where multiple robots work together for one specific task or serve the manufacturing process. Based on the workgroup's long-time experience in the area of optimization and monitoring [21–23], the development of less platform-specific

real-time tracking and monitoring systems has been foreseen.

5. CONCLUSIONS

This paper has analysed the existing state-of-the-art solutions of cyber-physical systems of mobile robots and concludes that many existing mobile robot platforms are non-modular and not compatible with extensions. The previously developed low-level control architecture of the CPS of a self-driving last-mile bus called TalTech iseAuto and the mobile industrial robot BoxBot was analysed in more detail and compared. By using the same architecture as a full-scale self-driving vehicle, a new small-scale logistic robot BoxBot 2 was developed for industrial indoor logistics. The target is to move materials in rather tight spaces in industrial areas and corridors. The concept derived from self-driving vehicles has been converted to a multi-model cyber-physical architecture for an industrial autonomous mobile robot where the long-term goal is to create a situation where multiple robots work together for one specific task or serve the manufacturing process. The risk evaluation model is proposed by introducing the criteria and risks featured in particular mobile robot systems.

ACKNOWLEDGEMENTS

This research received funding from two grants: the European Union's Horizon 2020 Research and Innovation Programme, under the grant agreement No. 856602; and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No. 2014-2020.4.01.20-0289. The publication costs of this article were covered by the Estonian Academy of Sciences and Tallinn University of Technology.

REFERENCES

1. Riives, J., Karjust, K., Küttner, R., Lemmik, R., Koov, K. and Lavin, J. Software development platform for integrated manufacturing engineering system. In *Proceedings of the 8th International DAAAM Baltic Conference "Industrial Engineering", Tallinn, Estonia, April 19–21, 2012*. Tallinn University of Technology, 555–560.
2. Sell. R., Rassõlkin, A., Wang, R. and Otto, T. Integration of autonomous vehicles and Industry 4.0. *Proc. Est. Acad. Sci.*, 2019, **68**(4), 389–394.
3. Christophe, F., Sell, R., Bernard, A. and Coatanéa, E. OPAS: Ontology processing for assisted synthesis of conceptual design solutions. In *Proceedings of the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 249–260.
4. Sell, R., Coatanéa, E. and Christophe, F. Important aspects of early design in mechatronic. In *Proceedings of the 6th International Conference of DAAAM Baltic Industrial Engineering*, *Tallinn, Estonia, April 24–26, 2008*. Tallinn University of Technology, 177–182.
5. Sell, R. and Petritsenko, A. Early design and simulation toolkit for mobile robot platforms. *Int. J. Prod. Dev.*, 2013, **18**(2), 168–192.
6. Pikner, H. and Karjust, K. Multi-layer cyber-physical low-level control solution for mobile robots. *IOP Conf. Ser.: Mater. Sci. Eng.*, 2021, **1140**, 012048.
7. Snatkin, A., Eiskop, T., Karjust, K. and Majak, K. Production monitoring system development and modification. *Proc. Est. Acad. Sci.*, 2015, **64**(4S), 567–580.
8. Sawant, A., Lenina, S. and Joshi, D. CAN, FlexRay, MOST versus ethernet for vehicular networks. *Int. J. Innov. Adv. Comput. Sci.*, 2018, **7**(4), 336–339.
9. Pikner, H. Overview of cyber-physical control systems for self-driving vehicles. In *Proceedings of the 19th International Symposium 'Topical problems in the Field of Electrical and Power Engineering. Doctoral School of Energy and Geotechnology. III', Tartu, Estonia, January 14–17, 2020*. Tallinn University of Technology, 105–106.
10. Arnold, E., Al-Jarrah, O. Y., Dianati, M., Fallah, S., Oxtoby, D. and Mouzakitis, A. A survey on 3D object detection methods for autonomous driving applications. *IEEE Trans. Intell. Transp. Syst.*, 2019, **20**(10), 3782–3795.
11. Pikner, H., Sell, R., Karjust, K., Malayjerdi. E. and Velsker, T. Cyber-physical control system for autonomous logistic robot. In *Proceedings of the 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, *Gliwice, Poland, April 25–29, 2021*, 699–704.
12. Hellmund, A., Wirges, S., Taş, Ö. Ş., Bandera, C. and Salschneider, N. O. Robot operating system: A modular software framework for automated driving. In *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, November 1–4, 2016*. IEEE, 1564–1570.
13. Sell, R. and Otto, T. Remotely controlled multi robot environment. In *Proceedings of the 2008 19th EAEEIE Annual Conference, Tallinn, Estonia, June 29–July 2, 2008*. IEEE, 20–25.
14. Rohrich, R. F., Teixeira, M. A. S., Piardi, L. and de Oliveira, A. S. A bio-inspired approach for robot swarm in smart factories. In *Advances in Intelligent Systems and Computing*. Springer, Cham, 2020, **1093**, 303–314.
15. Ziyan, C. and Shiguo, L. China's self-driving car legislation study. *Comput. Law Secur. Rev.*, 2021, **41**, 105555.
16. Nilsson, D. K., Phung, P. H. and Larson, U. E. Vehicle ECU classification based on safety-security characteristics. In *Proceedings of the IET Road Transport Information and Control Conference and the ITS United Kingdom Members' Conference, Manchester, UK, May 20–22, 2008*, 102.
17. Chan, C.-Y. Trends in crash detection and occupant restraint technology. *Proc. IEEE*, 2007, **95**(2), 388–396.
18. Rassõlkin, A., Sell, R. and Leier, M. Development case study of the first estonian self-driving car, iseauto. *Electr. Control Commun. Eng.*, 2018, **14**(1), 81–88.
19. Raval, V. and Dentlinger, M. J. Risk landscape of autonomous cars. *EDPACS*, 2017, **56**(3), 1–18.

20. Lima, A., Rocha, F., Völp, M. and Esteves-Veríssimo, P. Towards safe and secure autonomous and cooperative vehicle ecosystems. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy – CPS-SPC, Vienna, Austria, October 28, 2016*. ACM, 59–70.
21. Kaganski, S., Majak, J. and Karjust, K. Fuzzy AHP as a tool for prioritization of key performance indicators. *Procedia CIRP*, 2018, **72**, 1227–1232.
22. Paavel, M., Karjust, K. and Majak, J. PLM maturity model development and implementation in SME. *Procedia CIRP*, 2017, **63**, 651–657.
23. Paavel, M., Karjust, K. and Majak, J. Development of a product lifecycle management model based on the fuzzy analytic hierarchy process. *Proc. Est. Acad. Sci.*, 2017, **66**(3), 279–286.

Mitmekihiline küberfüüsikaline juhtimismeetod mobiilsete robotite turvasüsteemide jaoks

Heiko Pikner, Raivo Sell, Jüri Majak ja Kristo Karjust

On käsitletud isejuhtivate sõidukite ja mobiilsete robotite madala taseme küberfüüsikalise süsteemi turvalisust ning mitmesuguseid riskihinnanguid. Isejuhtivaid sõidukeid ja mobiilseid roboteid kasutatakse ühistranspordilahendustes ning tööstuses järjest rohkem. Töö autorid on loonud mitmesuguseid eksperimentaalseid robotplatvorme, mida saab kasutada nii linna- kui ka tööstuskeskkonnas. Selliste platvormide ohutu töö põhineb madala taseme küberfüüsikalisel süsteemil. Viimane koosneb tihedalt integreeritud arvutuslike (küber-) üksuste kogumist, mis suhtleb füüsilise maailmaga ja integreerib arvutus- ja kommunikatsiooniaspektid juhtimis- ning jälgimistehnikaga. Olemasolevate katseplatvormide analüüs võimaldab tuletada meetodeid nende turvalisuse, ohutuse ja usaldusväärsuse tõstmiseks.