

Safe Performance of an Industrial Autonomous Ground Vehicle in the Supervisory Control Framework

Fotis N. Koumboulis ¹, Dimitrios G. Fragkoulis ^{1,*}, George E. Panagiotakis ^{1,2} and Efthimios Mavroeidis ²

¹ Robotics, Automatic Control and Cyber-Physical Systems Laboratory, Department of Digital Industry Technologies, School of Science, National and Kapodistrian University of Athens, Euripus Campus, 34400 Euboea, Greece; fkoumboulis@dind.uoa.gr (F.N.K.)

² Core Department, National and Kapodistrian University of Athens, Euripus Campus, 34400 Euboea, Greece

* Correspondence: dfragkoulis@dind.uoa.gr

Abstract: A Cyberphysical system, being an autonomous guided vehicle (AGV) and having diverse applications such as thematic parks and product transfer in manufacturing units, is modeled and controlled. The models of all subsystems of the AGV are provided in discrete event systems (DES) form following the Ramadge–Wonham (R–W) framework. The safe performance of the AGV, being the desired behavior of the system, is presented in the form of desired rules and translated into a set of regular languages. Then, the regular languages are realized as supervisory automata in the framework of Supervisory Control Theory (SCT). To ease implementation and coordination of the control architecture, the supervisors are designed to be in two-state automata forms. The controllability of the regular languages, regarding the AGV, will be proved, using the physical realizability (PR) of the synchronous product of the automata of the system and the supervisors. Also, the nonblocking property of all the controlled automata will be proven to be satisfied. Simulation of the controlled AGV will validate the proposed method.

Keywords: AGV; discrete event systems; supervisory control theory



Citation: Koumboulis, F.N.; Fragkoulis, D.G.; Panagiotakis, G.E.; Mavroeidis, E. Safe Performance of an Industrial Autonomous Ground Vehicle in the Supervisory Control Framework. *Electronics* **2023**, *12*, 5035. <https://doi.org/10.3390/electronics12245035>

Academic Editors: Andrea Bonci, Jianchao Luo and Jun Xu

Received: 16 October 2023
Revised: 13 December 2023
Accepted: 14 December 2023
Published: 17 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Manufacturing Cyberphysical Systems (CPS), (also known as cybermanufacturing systems [1], are continuing to expand, following the directions of Industry 4.0 [2–5]. This way, all the parts of a manufacturing process are appropriately interconnected to share their information, e.g., through the Industrial Internet of Things (IIOT), and they are controlled or monitored by industrial computers [3]. Clearly, the control of the coordination of such a complex system is a key factor for successful integration [6,7]. Furthermore, the safety and security demands of manufacturing CPSs make even more difficult their analysis, design, and modeling. Thus, the use of approaches based on formal methods becomes more and more a necessity [7,8]. Regular languages and linear temporal logic are two efficient tools to describe, at a high level, the desired behavior of the manufacturing process [7], where the first tool is inherently related to Supervisory Control Theory (SCT) [9,10].

SCT is based on the modeling of CPSs as a finite symbolic system. SCT often uses abstractions of the real CPS. The first aim of SCT is the synthesis of a supervisory control architecture configuration, enforcing the desired behavior of the process while being as permissive as possible. The second aim is the easy and robust implementation of the designed scheme to the real CPS. The main issue in SCT design is the size and complexity of the total symbolic mathematical model, including all CPS's devices and subsystems, as their size and complexity tend to grow exponentially with respect to the development of new technologies [11,12]. Another issue of SCT, which aims to provide formal and effective solutions, is cyberattacks at the level of the physical part of the CPS [13–15].

The use of intelligent assistants in manufacturing CPS in various areas of applications is becoming a normal trend [16–18]. Automated Guided Vehicles (AGVs) are used for

intelligent transfers, not only in manufacturing, product processing, and product storage, but also in the large infrastructures of the modern economy. In such systems, SCT is used for coordination control [6]. The use of specially designed AGVs that navigate and cooperate autonomously is met in a large class of manufacturing CPSs [6,19,20].

In [21–23], an AGV product of the ETF Group [24], called multimover, has been introduced and described. Multimover is usually used as a ride system in theme parks and museums, as well as at special events. This AGV follows a floor-integrated electrical wire, producing a magnetic field that is measured by the embedded track sensors of the AGV. Furthermore, appropriate codes are placed on the floor surface in certain positions near the wire and at points of particular interest, e.g., the starting point and any destination point. Multimover reads the floor codes using appropriate metal detectors. Using multimover, an industrial AGV has been proposed in a study [6] and is used as a forklift that transfers products between fixed stations. The architecture of this AGV is identical to that of multimover, in the sense that they have the same devices and subsystems. The AGV, proposed in [6], is equipped with a product lifting and carrying mechanism in forklift configuration. It is important to mention that the operation of the lifting and carrying mechanism is treated independently from the operation of multimover. The model of this multimover-type industrial AGV has been developed in [6] and is proposed to be used in manufacturing systems. The model is based on state- and event-based abstractions of the actual devices of the systems and their respective low-level controllers. Thus, this model describes the real behavior of the low-level controlled devices of the AGV. Based on its characteristics and applications, multimover can be used as a general-purpose AGV.

In [21,22], the mathematical description of this type of AGV is given using discrete event systems (DES). In [21], for brevity reasons, only some components are presented, while in [22], descriptions of the sensors and actuators of the vehicle are given. Furthermore, in [22], the supervisory control level is studied for the case where the desired requirements are realized using rules of logical expressions. In [21,22], as well as in [6], the controllability and nonblocking issues are not investigated.

Here, the description of the AGV for manufacturing use, proposed in [6], is expressed using 6-tuple DES in the R-W framework [9,10]. The navigation motors and controllers, the proximity and bumper sensors, and the LEDs and buttons are modeled separately. The design requirements of the system are introduced in the form of rules. The rules are translated into regular languages. This is a contribution of the present work. Following the design approaches proposed in [25–28], a general two-state supervisor that covers the desired regular languages is introduced. The controllability of the desired languages is proven with regard to the total CPS. This property is proved via the physical realizability (PR) [29] of the supervisors in the synchronous product of the system's automaton and the supervisors. The controllability property of the desired languages is a necessary and sufficient condition for the proposed supervisors to be physically realizable, see [9,10,29]. This is another contribution of the present work. An analytic proof of the nonblocking property of the controlled system is given. This is the third contribution of the paper. Finally, it is important to mention that the present control scheme is event-based, while in [6], the supervisors are state-dependent. Here, the proposed supervisory scheme responds to changes in the system through the appearance of the signal corresponding to the change. This way, a faster execution of control with respect to changes may be derived.

The first motivation of the paper is the huge number of applications of AGVs. So, the present work was directed at the analytic modeling, and supervisor design of AGVs that are clearly offered for the development of extended or new supervisor designs. The second motivation is the clear need to implement event-based and simple supervisors. The third motivation is the need to ensure satisfactory system performance by proving controllability through PR and the nonblocking characteristics of the influence of the supervisors.

The material of the paper is grouped into six sections. In Section 2, the mathematical model of the AGV, in the form of 6-tuple DES models of its subsystems, will be presented. In Section 3, the desired behavior, in the form of rules and in the form of regular languages,

will be proposed. In Section 4, a general two-state supervisor automaton will be proposed to cover all realizations of the desired regular languages. In Section 5, the implementation issues of the derived supervisors are discussed. In Section 6, indicative simulation results are presented.

2. DES Model of the AGV

2.1. A Brief Description of the AGV

The class of AGVs studied in the present paper includes autonomous driving electric vehicles with embedded batteries, controlled via computers. This class of AGVs is produced by ETF Group in the Netherlands, and it is called multimover. In [21,22], an abstraction of the actual functionality of the vehicle that incorporates the low-level controllers of the AGV is presented in the form of DES models. In [6], multimover is used for industrial manufacturing. In Figure 1, an indicative form of industrial AGV is presented. AGVs can be used in various applications, as they can be found to be installed in manufacturing or logistic facilities [6] as well as in theme parks [21]. Each AGV of the present class is equipped with two different motors, one for driving and one for steering, a controller for the riding coordination, a battery level sensor, two proximity sensors, one bumper sensor, three LEDs indicating the forward and backward movement and the mode status, and three buttons for forward, backward, and reset selection. Each of the above devices is a separate subsystem of the AGV. Additionally, the operation modes of the vehicle constitute one more subsystem, being the central subsystem of the AGV.



Figure 1. An indicative form of AGVs for product transfer.

In the following subsections, the models of the subsystems will be presented in the form of 6-tuple automata in the R-W framework. For 6-tuple forms of DES in the R-W framework, see [25–28]. The DES models of the subsystems will be derived following the respective directions in [6], where a hybrid model for the present class of AGVs has been introduced. The overall model of the AGV will be derived from the synchronous product of the models of its subsystems. Regarding the synchronous product, see [9,10].

Before presenting the models of all devices of the AGV, it is important to mention that the following assumptions, introduced in [6,21,22], are also adopted here:

1. There is no fault in the resource control of the AGV. Hence, all the given commands are executed correctly. This assumption is valid as the already embedded resource controllers of the AGV have passed various quality and functionality tests.
2. The communication delay between the AGV and the supervisors is practically equal to zero. Thus, the synchronization of the supervisors and the events of the AGV is immediate.

2.2. Operation Subsystem

The DES model of the AGV operation modes is

$$\mathbf{G}_{MM} = (\mathbb{Q}_{MM}, \mathbb{E}_{MM}, f_{MM}, \mathbb{H}_{MM}, x_{MM,0}, \mathbb{Q}_{MM,m}).$$

The state set of \mathbf{G}_{MM} is $\mathbb{Q}_{MM} = \{q_{MM,1}, q_{MM,2}, q_{MM,3}\}$. The AGV has three different operational modes: the emergency mode, the reset mode, and the active mode. The state $q_{MM,1}$ represents the emergency mode. The state $q_{MM,2}$ represents the reset mode. The state $q_{MM,3}$ represents the active mode. The alphabet of \mathbf{G}_{MM} is $\mathbb{E}_{MM} = \{e_{MM,1}, e_{MM,2}, e_{MM,3}\}$. The event $e_{MM,1}$ is the command to switch to reset mode. The event $e_{MM,2}$ is the command to switch to active mode. The event $e_{MM,3}$ is the command to switch to emergency mode. $x_{MM,0} = q_{MM,1}$ is the initial state and $\mathbb{Q}_{MM,m} = \{q_{MM,1}\}$ is the marked state set. The active events sets of \mathbf{G}_{DM} are:

$$\mathbb{H}_{MM}(q_{MM,1}) = \{e_{MM,1}\}, \mathbb{H}_{MM}(q_{MM,2}) = \{e_{MM,2}, e_{MM,3}\}, \mathbb{H}_{MM}(q_{MM,3}) = \{e_{MM,1}, e_{MM,3}\}.$$

The transitions of \mathbf{G}_{MM} are:

$$f_{MM}(q_{MM,1}, e_{MM,1}) = q_{MM,2}, f_{MM}(q_{MM,2}, e_{MM,2}) = q_{MM,3}, f_{MM}(q_{MM,2}, e_{MM,3}) = q_{MM,1}, \\ f_{MM}(q_{MM,3}, e_{MM,1}) = q_{MM,2}, f_{MM}(q_{MM,3}, e_{MM,3}) = q_{MM,2}.$$

The controllable event set is $\mathbb{E}_{MM,c} = \{e_{MM,1}, e_{MM,2}\}$ and the uncontrollable event set is $\mathbb{E}_{MM,uc} = \{e_{MM,3}\}$. The marked behavior and the closed behavior of \mathbf{G}_{MM} are:

$$\mathbb{L}_m(\mathbf{G}_{MM}) = (e_{MM,1}((e_{MM,2}e_{MM,1})^* + (e_{MM,2}e_{MM,1})^*e_{MM,2})e_{DM,3})^*, \mathbb{L}(\mathbf{G}_{MM}) = \overline{\mathbb{L}(\mathbf{G}_{MM})}.$$

Obviously, \mathbf{G}_{MM} is a nonblocking automaton [9,10]. In Figure 2, the state diagram of \mathbf{G}_{MM} is depicted.

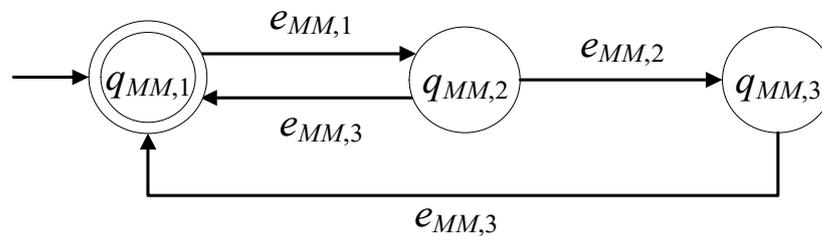


Figure 2. State diagram of \mathbf{G}_{MM} .

2.3. Drive Motor Subsystem

The DES model of the drive motor (DM) is:

$$\mathbf{G}_{DM} = (\mathbb{Q}_{DM}, \mathbb{E}_{DM}, f_{DM}, \mathbb{H}_{DM}, x_{DM,0}, \mathbb{Q}_{DM,m})$$

The state set of \mathbf{G}_{DM} is $\mathbb{Q}_{DM} = \{q_{DM,1}, q_{DM,2}, q_{DM,3}\}$. DM has three different operations (OFF operation, ON operation, and STOPPING operation). The state $q_{DM,1}$ represents the OFF mode, $q_{DM,2}$ represents the ON mode, and $q_{DM,3}$ represents the STOPPING mode. The alphabet of \mathbf{G}_{DM} is $\mathbb{E}_{DM} = \{e_{DM,1}, e_{DM,2}, e_{DM,3}, e_{DM,4}, e_{DM,5}\}$. The event $e_{DM,1}$ is the command to move forward. The event $e_{DM,2}$ is the command to DM to move backward. The event $e_{DM,3}$ is the command to DM to stop moving. The event $e_{DM,4}$ is the command to DM to be deactivated. The event $e_{DM,5}$ denotes the indication of a DM fault. $x_{DM,0} = q_{DM,1}$ is the initial state and $\mathbb{Q}_{DM,m} = \{q_{DM,1}\}$ is the marked state set of \mathbf{G}_{DM} . The active event sets of \mathbf{G}_{DM} are:

$$\mathbb{H}_{DM}(q_{DM,1}) = \{e_{DM,1}, e_{DM,2}, e_{DM,3}\}, \mathbb{H}_{DM}(q_{DM,2}) = \{e_{DM,3}, e_{DM,5}\}, \\ \mathbb{H}_{DM}(q_{DM,3}) = \{e_{DM,1}, e_{DM,2}, e_{DM,4}, e_{DM,5}\}$$

The transitions of G_{DM} are:

$$\begin{aligned} f_{DM}(q_{DM,1}, e_{DM,1}) &= q_{DM,2}, f_{DM}(q_{DM,1}, e_{DM,2}) = q_{DM,2}, f_{DM}(q_{DM,1}, e_{DM,3}) = q_{DM,3}, \\ f_{DM}(q_{DM,2}, e_{DM,3}) &= q_{DM,3}, f_{DM}(q_{DM,2}, e_{DM,5}) = q_{DM,1}, f_{DM}(q_{DM,3}, e_{DM,1}) = q_{DM,2}, \\ f_{DM}(q_{DM,3}, e_{DM,2}) &= q_{DM,2}, f_{DM}(q_{DM,3}, e_{DM,4}) = q_{DM,1}, f_{DM}(q_{DM,3}, e_{DM,5}) = q_{DM,1} \end{aligned}$$

The controllable event set is $\mathbb{E}_{DM,c} = \{e_{DM,1}, e_{DM,2}, e_{DM,3}, e_{DM,4}\}$ and the uncontrollable event set is $\mathbb{E}_{DM,uc} = \{e_{DM,5}\}$. The marked behavior and the closed behavior of G_{DM} are:

$$\begin{aligned} \mathbb{L}_m(G_{DM}) &= (((e_{DM,1} + e_{DM,2})(e_{DM,5} + (e_{DM,3}(e_{DM,1} + e_{DM,2}))^*(e_{DM,5} + e_{DM,3}(e_{DM,4} + e_{DM,5})))) + \\ &+ (e_{DM,3}((e_{DM,4} + e_{DM,5}) + ((e_{DM,1} + e_{DM,2})e_{DM,3})^*((e_{DM,4} + e_{DM,5}) + (e_{DM,1} + e_{DM,2})e_{DM,5}))))^* , \\ \mathbb{L}(G_{DM}) &= \overline{\mathbb{L}(G_{DM})}. \end{aligned}$$

Obviously, G_{DM} is a nonblocking automaton. In Figure 3, the state diagram of G_{DM} is depicted.

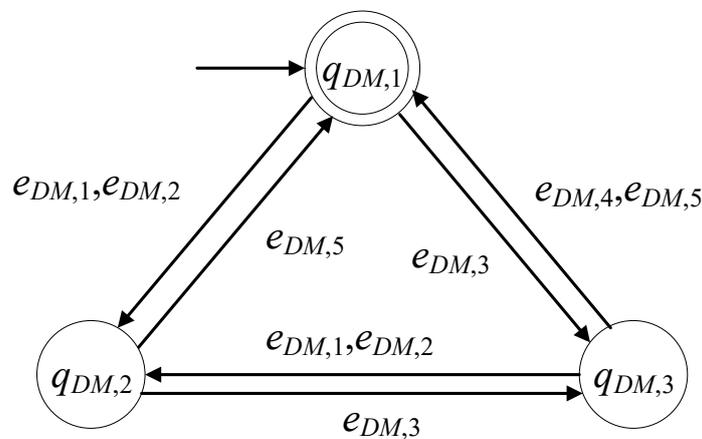


Figure 3. State diagram of G_{DM} .

2.4. Steer Motor Subsystem

The DES model of the steer motor (SM) is:

$$G_{SM} = (Q_{SM}, \mathbb{E}_{SM}, f_{SM}, \mathbb{H}_{SM}, x_{SM,0}, Q_{SM,m})$$

The state set of G_{SM} is $Q_{SM} = \{q_{SM,1}, q_{SM,2}\}$. SM has two different operations (OFF and ON). The state $q_{SM,1}$ represents OFF. The state $q_{SM,2}$ represents ON. The alphabet of G_{SM} is $\mathbb{E}_{SM} = \{e_{SM,1}, e_{SM,2}, e_{SM,3}\}$. The event $e_{SM,1}$ is the command for SM to be activated. The event $e_{SM,2}$ is the command for SM to be deactivated. The event $e_{SM,3}$ denotes an indication of an SM fault. $x_{SM,0} = q_{SM,1}$ is the initial state and $Q_{SM,m} = \{q_{SM,1}\}$ is the marked state set of G_{SM} . The active event sets of G_{SM} are:

$$\mathbb{H}_{SM}(q_{SM,1}) = \{e_{SM,1}, e_{SM,3}\}, \mathbb{H}_{SM}(q_{SM,2}) = \{e_{SM,2}, e_{SM,3}\}$$

The transitions of G_{SM} are:

$$\begin{aligned} f_{SM}(q_{SM,1}, e_{SM,1}) &= q_{SM,2}, f_{SM}(q_{SM,1}, e_{SM,3}) = q_{SM,1}, f_{SM}(q_{SM,2}, e_{SM,2}) = q_{SM,1}, \\ f_{SM}(q_{SM,2}, e_{SM,3}) &= q_{SM,1} \end{aligned}$$

The controllable event set is $\mathbb{E}_{SM,c} = \{e_{SM,1}, e_{SM,2}\}$ and the uncontrollable event set is $\mathbb{E}_{SM,uc} = \{e_{SM,3}\}$. The marked behavior and the closed behavior of G_{SM} are:

$$\mathbb{L}_m(G_{SM}) = (e_{SM,3}^* e_{SM,1} (e_{SM,2} + e_{SM,3}))^*, \mathbb{L}(G_{SM}) = \overline{\mathbb{L}_m(G_{SM})}$$

Obviously, G_{SM} is a nonblocking automaton. In Figure 4, the state diagram of G_{SM} is depicted.

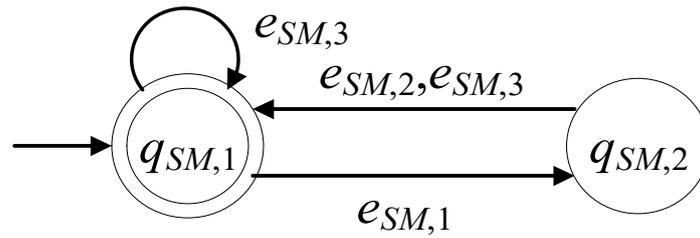


Figure 4. State diagram of G_{SM} .

2.5. Ride Control Subsystem

The DES model of ride control (RC) is:

$$G_{RC} = (Q_{RC}, E_{RC}, f_{RC}, H_{RC}, x_{RC,0}, Q_{RC,m})$$

The state set of G_{RC} is $Q_{RC} = \{q_{RC,1}, q_{RC,2}\}$. RC has two different operations (START and STOP). The state $q_{RC,1}$ represents START and $q_{RC,2}$ represents STOP. The alphabet of G_{RC} is $E_{RC} = \{e_{RC,1}, e_{RC,2}\}$. The event $e_{RC,1}$ is the signal from RC, indicating that it has been deactivated. The event $e_{RC,2}$ is the signal of RC, indicating that it has been activated. $x_{RC,0} = q_{RC,1}$ is the initial state, and $Q_{RC,m} = \{q_{RC,1}\}$ is the marked state set of G_{RC} . The active event set of G_{RC} are:

$$H_{RC}(q_{RC,1}) = H_{RC}(q_{RC,2}) = E_{RC}$$

The transitions of G_{RC} are:

$$f_{RC}(q_{RC,1}, e_{RC,1}) = q_{RC,2}, f_{RC}(q_{RC,1}, e_{RC,2}) = q_{RC,1}, f_{RC}(q_{RC,2}, e_{RC,1}) = q_{RC,2}, f_{RC}(q_{RC,2}, e_{RC,2}) = q_{RC,1}$$

The controllable event set is $E_{RC,c} = \emptyset$ and the uncontrollable event set is $E_{RC,uc} = E_{RC}$. The marked behavior and the closed behavior of G_{RC} are:

$$L_m(G_{RC}) = (e_{RC,2}^* e_{RC,1} e_{RC,1}^* e_{RC,2})^*, L(G_{RC}) = \overline{L_m(G_{RC})}$$

Obviously, G_{RC} is a nonblocking automaton. In Figure 5, the state diagram of G_{RC} is depicted.

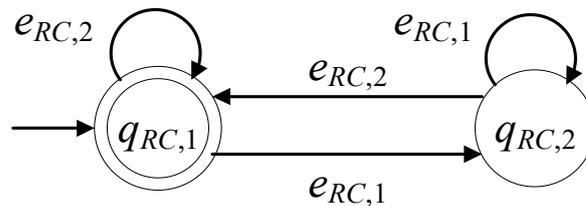


Figure 5. State diagram of G_{RC} .

2.6. Battery Level Sensor Subsystem

The DES model of the battery level (BL) sensor is:

$$G_{BL} = (Q_{BL}, E_{BL}, f_{BL}, H_{BL}, x_{BL,0}, Q_{BL,m})$$

The state set of G_{BL} is $Q_{BL} = \{q_{BL,1}, q_{BL,2}, q_{BL,3}\}$. The battery level sensor has three different operations (INACTIVE, ACTIVE, and RELOAD). The state $q_{BL,1}$ represents the INACTIVE operation. The state $q_{BL,2}$ represents the ACTIVE operation. The state $q_{BL,3}$ represents the RELOAD operation. The alphabet of G_{BL} is $E_{BL} = \{e_{BL,1}, e_{BL,2}, e_{BL,3}\}$. The event $e_{BL,1}$ is the signal from the battery level sensor, indicating that the battery is exhausted. The event $e_{BL,2}$ is the signal from the battery level sensor, indicating that the battery is

reloading. The event $e_{BL,3}$ is the signal from the battery level sensor, indicating that the battery is full. $x_{BL,0} = q_{BL,1}$ is the initial state and $\mathbb{Q}_{BL,m} = \{q_{BL,1}\}$ is the marked state set of \mathbf{G}_{BL} . The active event sets of \mathbf{G}_{BL} are:

$$\mathbb{H}_{BL}(q_{BL,1}) = \{e_{BL,1}\}, \mathbb{H}_{BL}(q_{BL,2}) = \{e_{BL,2}\}, \mathbb{H}_{BL}(q_{BL,3}) = \{e_{BL,3}\}.$$

The transitions of \mathbf{G}_{BL} are:

$$f_{BL}(q_{BL,1}, e_{BL,1}) = q_{BL,2}, f_{BL}(q_{BL,2}, e_{BL,2}) = q_{BL,3}, f_{BL}(q_{BL,3}, e_{BL,3}) = q_{BL,1}.$$

The controllable event set is $\mathbb{E}_{BL,c} = \emptyset$ and the uncontrollable event set is $\mathbb{E}_{BL,uc} = \mathbb{E}_{BL}$. The marked behavior and the closed behavior of \mathbf{G}_{BL} are:

$$\mathbb{L}_m(\mathbf{G}_{BL}) = (e_{BL,1}e_{BL,2}e_{BL,3})^*, \mathbb{L}(\mathbf{G}_{BL}) = \overline{\mathbb{L}_m(\mathbf{G}_{BL})}.$$

Obviously, \mathbf{G}_{BL} is a nonblocking automaton. In Figure 6, the state diagram of \mathbf{G}_{BL} is depicted.

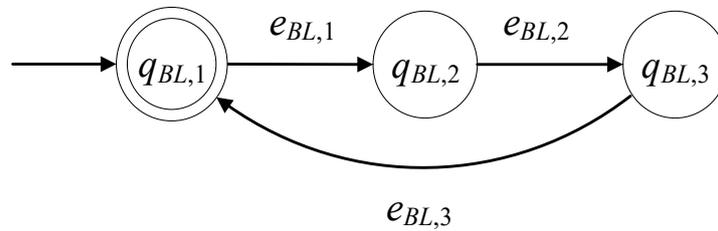


Figure 6. State diagram of \mathbf{G}_{BL} .

2.7. Proximity Sensor Subsystem

The DES model of the i -th Proximity sensor (PR), where $i \in \{1, 2\}$, is:

$$\mathbf{G}_{PS,i} = (\mathbb{Q}_{PS,i}, \mathbb{E}_{PS,i}, f_{PS,i}, \mathbb{H}_{PS,i}, x_{PS,i,0}, \mathbb{Q}_{PS,i,m})$$

The state set of $\mathbf{G}_{PS,i}$ is $\mathbb{Q}_{PS,i} = \{q_{PS,i,1}, q_{PS,i,2}\}$. PS has two different operations (INACTIVE and ACTIVE). The state $q_{PS,i,1}$ represents the case where PR is INACTIVE, and $q_{PS,i,2}$ represents the case where PS is ACTIVE. The alphabet of $\mathbf{G}_{PS,i}$ is $\mathbb{E}_{PS,i} = \{e_{PS,i,1}, e_{PS,i,2}\}$. The event $e_{PS,i,1}$ is the signal from PS indicating that an object has been detected nearby. The event $e_{PS,i,2}$ is the signal from PS indicating that the detected object has stopped. $x_{PS,i,0} = q_{PS,i,1}$ is the initial state and $\mathbb{Q}_{PS,i,m} = \{q_{PS,i,1}\}$ is the marked state set of $\mathbf{G}_{PS,i}$. The active event set of $\mathbf{G}_{PS,i}$ are:

$$\mathbb{H}_{PS,i}(q_{PS,i,1}) = \{e_{PS,i,1}\}, \mathbb{H}_{PS,i}(q_{PS,i,2}) = \{e_{PS,i,2}\}.$$

The transitions of $\mathbf{G}_{PS,i}$ are:

$$f_{PS,i}(q_{PS,i,1}, e_{PS,i,1}) = q_{PS,i,2}, f_{PS,i}(q_{PS,i,2}, e_{PS,i,2}) = q_{PS,i,1}$$

The controllable event set is $\mathbb{E}_{PS,i,c} = \emptyset$ and the uncontrollable event set is $\mathbb{E}_{PS,i,uc} = \mathbb{E}_{PS,i}$. The marked behavior and closed the behavior of $\mathbf{G}_{PS,i}$ are:

$$\mathbb{L}_m(\mathbf{G}_{PS,i}) = (e_{PS,i,1}e_{PS,i,2})^*, \mathbb{L}(\mathbf{G}_{PS,i}) = \overline{\mathbb{L}_m(\mathbf{G}_{PS,i})}$$

Obviously, $\mathbf{G}_{PS,i}$ is a nonblocking automaton. In Figure 7, the state diagram of $\mathbf{G}_{PS,i}$ is depicted.

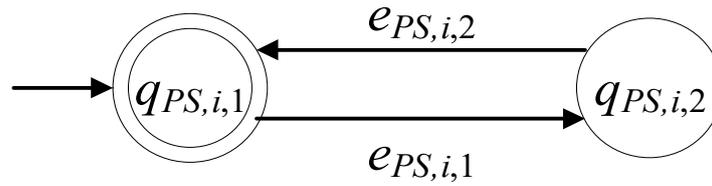


Figure 7. State diagram of $G_{PS,i}$.

2.8. Bumper Switch Subsystem

The DES model of the bumper switch (BS) is:

$$G_{BS} = (\mathbb{Q}_{BS}, \mathbb{E}_{BS}, f_{BS}, \mathbb{H}_{BS}, x_{BS,0}, \mathbb{Q}_{BS,m})$$

The state set of G_{BS} is $\mathbb{Q}_{PS} = \{q_{PS,1}, q_{PS,2}\}$. BS has two different operations (INACTIVE and ACTIVE) The state $q_{BS,1}$ represents the case where BS is INACTIVE, and $q_{BS,2}$ represents the case where BS is ACTIVE. The alphabet of G_{BS} is $\mathbb{E}_{BS} = \{e_{BS,1}, e_{BS,2}\}$. The event $e_{BS,1}$ is the signal indicating that BS has been activated. The event $e_{BS,2}$ is the signal indicating that BS has been deactivated. $x_{BS,0} = q_{BS,1}$ is the initial state and $\mathbb{Q}_{BS,m} = \{q_{BS,1}\}$ the marked state set of G_{BS} . The active event set of G_{BS} are:

$$\mathbb{H}_{BS}(q_{BS,1}) = \{e_{BS,1}\}, \mathbb{H}_{BS}(q_{BS,2}) = \{e_{BS,2}\}$$

The transitions of G_{BS} are:

$$f_{BS}(q_{BS,1}, e_{BS,1}) = q_{BS,2}, f_{BS}(q_{BS,2}, e_{BS,2}) = q_{BS,1}$$

The controllable event set is $\mathbb{E}_{BS,c} = \emptyset$ and the uncontrollable event set is $\mathbb{E}_{BS,uc} = \mathbb{E}_{BS}$. The marked behavior and the closed behavior of G_{BS} are:

$$\mathbb{L}_m(G_{BS}) = (e_{BS,1}e_{BS,2})^*, \mathbb{L}(G_{BS}) = \overline{\mathbb{L}_m(G_{BS})}$$

Obviously, G_{BS} is a nonblocking automaton. In Figure 8, the state diagram of G_{BS} is depicted.

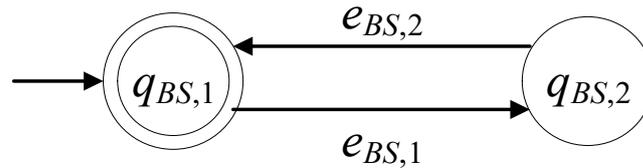


Figure 8. State diagram of G_{BS} .

2.9. LED Subsystem

The DES model of the LEDs in 6-tuple DES form is:

$$G_{L,j} = (\mathbb{Q}_{L,j}, \mathbb{E}_{L,j}, f_{L,j}, \mathbb{H}_{L,j}, x_{L,j,0}, \mathbb{Q}_{L,j,m}); j \in \{1, 2, 3\}$$

The state set of $G_{L,j}$ is $\mathbb{Q}_{L,j} = \{q_{L,j,1}, q_{L,j,2}\}$. The LEDs has two different modes (OFF and ON). The state $q_{L,j,1}$ represents the OFF mode, and the state $q_{L,j,2}$ represents the ON mode. The alphabet of $G_{L,i}$ is $\mathbb{E}_{L,j} = \{e_{L,j,1}, e_{L,j,2}\}$. The event $e_{L,j,1}$ is the command to activate the j -th LED. The event $e_{L,j,2}$ is the command to deactivate the j -th LED. Regarding the initial state of $G_{L,j}$, if $j \in \{1, 2\}$ then $x_{L,j,0} = q_{L,j,1}$, and if $j = 3$ then $x_{L,j,0} = q_{L,j,2}$. The marked state set of $G_{L,j}$ is $\mathbb{Q}_{L,j,m} = \{q_{L,j,1}\}$. The active event set of $G_{L,j}$ are:

$$\mathbb{H}_{L,j}(q_{L,j,1}) = \{e_{L,j,1}\}, \mathbb{H}_{L,j}(q_{L,j,2}) = \{e_{L,j,2}\}$$

The transitions of $\mathbf{G}_{L,j}$ are::

$$f_{L,j}(q_{L,j,1}, e_{L,j,1}) = q_{L,j,2}, f_{L,j}(q_{L,j,2}, e_{L,j,2}) = q_{L,j,1}$$

The controllable event set is $\mathbb{E}_{L,j,c} = \mathbb{E}_{L,j}$ and the uncontrollable event set is $\mathbb{E}_{L,j,uc} = \emptyset$. The marked and the closed behavior of $\mathbf{G}_{L,j}$ is

$$\mathbb{L}_m(\mathbf{G}_{L,j}) = \begin{cases} (e_{L,j,1}e_{L,j,2})^*, & \text{if } j \in \{1,2\} \\ e_{L,j,2}(e_{L,j,1}e_{L,j,2})^*, & \text{if } j = 3 \end{cases},$$

$$\mathbb{L}(\mathbf{G}_{L,j}) = \overline{\mathbb{L}_m(\mathbf{G}_{L,j})}$$

Obviously, $\mathbf{G}_{L,j}$ is a nonblocking automaton. In Figure 9, the state diagram of $\mathbf{G}_{L,j}$ is depicted.

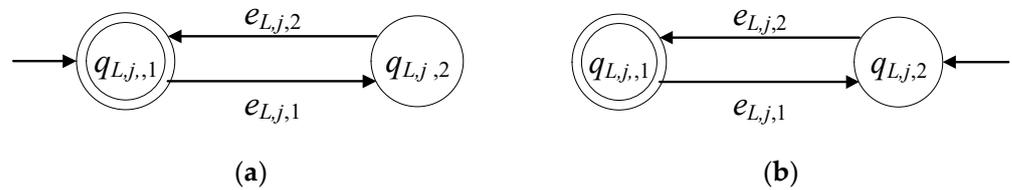


Figure 9. State diagram of $\mathbf{G}_{L,j}$ (a) for $j \in \{1,2\}$ and (b) for $j = 3$.

Note that $\mathbf{G}_{L,1}$ describes the behavior of the LED, which is related to the forward movement of the AGV. Also, note that $\mathbf{G}_{L,2}$ describes the behavior of the LED, which is related to the backward movement of the AGV. Finally, note that $\mathbf{G}_{L,3}$ describes the behavior of the LED, which is related to the case where the AGV is in reset mode.

2.10. Buttons Subsystem

The DES model of the buttons is:

$$\mathbf{G}_{B,k} = (\mathbb{Q}_{B,k}, \mathbb{E}_{B,k}, f_{B,k}, \mathbb{H}_{B,k}, x_{B,k,0}, \mathbb{Q}_{B,k,m}); k \in \{1,2,3\}$$

The state set of $\mathbf{G}_{B,k}$ is $\mathbb{Q}_{B,k} = \{q_{B,k,1}, q_{B,k,2}\}$. The buttons have two different modes. The first is the RELEASED mode, and the second is the PRESSED mode. The state $q_{B,k,1}$ represents the RELEASED mode. The state $q_{B,k,2}$ represents the PRESSED mode. The alphabet of $\mathbf{G}_{B,k}$ is $\mathbb{E}_{B,k} = \{e_{B,k,1}, e_{B,k,2}\}$. The event $e_{B,k,1}$ is the signal indicating that the k -th button has been pressed. The event $e_{B,k,2}$ is the signal indicating that the k -th button has been released. $x_{B,k,0} = q_{B,k,1}$ is the initial state and $\mathbb{Q}_{B,k,m} = \{q_{B,k,1}\}$ is the marked state set of $\mathbf{G}_{B,k}$ is. The active event set of $\mathbf{G}_{B,k}$ are:

$$\mathbb{H}_{B,k}(q_{B,k,1}) = \{e_{B,k,1}\}, \mathbb{H}_{B,k}(q_{B,k,2}) = \{e_{B,k,2}\}$$

The transitions of $\mathbf{G}_{B,k}$ are:

$$f_{B,k}(q_{B,k,1}, e_{B,k,1}) = q_{B,k,2}, f_{B,k}(q_{B,k,2}, e_{B,k,2}) = q_{B,k,1}$$

The controllable event set is $\mathbb{E}_{B,k,c} = \emptyset$ and the uncontrollable event set is $\mathbb{E}_{B,k,uc} = \mathbb{E}_{B,k}$. The marked behavior and the closed behavior of $\mathbf{G}_{B,k}$ are:

$$\mathbb{L}_m(\mathbf{G}_{B,k}) = (e_{B,k,1}e_{B,k,2})^*, \mathbb{L}(\mathbf{G}_{B,k}) = \overline{\mathbb{L}_m(\mathbf{G}_{B,k})}$$

Obviously, $\mathbf{G}_{B,k}$ is a nonblocking automaton. In Figure 10, the state diagram of $\mathbf{G}_{B,k}$ is depicted.

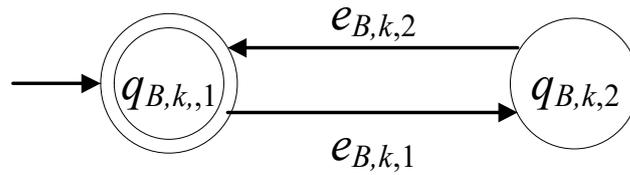


Figure 10. State diagram of $G_{B,k}$.

Note that $G_{B,1}$ describes the behavior of the button, which is related to the forward movement of the AGV. Also, note that $G_{B,2}$ describes the behavior of the button, which is related to the backward movement of the AGV. Finally, note that $G_{B,3}$ describes the behavior of the reset button on the AGV.

3. Performance Requirements

In the present section, the desired languages for formulating the desired behavior of the AGV will be presented. The languages are based on the performance rules proposed in [6]. Following [6], the goal is to state the desired performance in the form of rules that are as simple as possible. This characteristic will benefit the synchronization of the supervisors in the controller implementation.

3.1. Requirements for Emergency Behavior

The transition of the AGV to the RESET and ACTIVE states is desired to be allowed only if the battery sensor and the bumper switch are inactive. The following two regular languages formulate this specification:

$$\mathbb{K}_{D,1} = \overline{((e_{MM,1} + e_{MM,2} + e_{BL,3})^* e_{BL,1} (e_{BL,1})^* e_{BL,3})^*},$$

$$\mathbb{K}_{D,2} = \overline{((e_{MM,1} + e_{MM,2} + e_{BS,2})^* e_{BS,1} (e_{BS,1})^* e_{BS,2})^*}$$

3.2. Requirements for the Motors

The steer motor is allowed to be disabled only if the AGV is in the RESET or EMERGENCY state and the drive motor is in the OFF state. The following two regular languages formulate this specification:

$$\mathbb{K}_{D,3} = \overline{((e_{SM,2} + e_{MM,1} + e_{MM,3})^* e_{MM,2} (e_{MM,1} + e_{MM,3}))^*},$$

$$\mathbb{K}_{D,4} = \overline{((e_{SM,2} + e_{DM,5})^* (e_{DM,1} + e_{DM,2}) (e_{DM,3} + e_{DM,5}))^*}$$

The steer motor is allowed to be disabled only if the AGV is in the ACTIVE state. The following regular language formulates this specification:

$$\tilde{\mathbb{K}}_{D,5} = \overline{((e_{MM,1} + e_{MM,3})^* e_{MM,2} (e_{SM,1})^* (e_{MM,1} + e_{MM,3}))^*}$$

The drive motor is allowed to be enabled, forward or backward, only if the AGV is in the ACTIVE state and the steer motor is ON. The following two regular languages formulate this specification, where the first language is a modification of the previous one:

$$\mathbb{K}_{D,5} = \overline{((e_{MM,1} + e_{MM,3})^* e_{MM,2} (e_{SM,1} + e_{DM,1} + e_{DM,2})^* (e_{MM,1} + e_{MM,3}))^*},$$

$$\mathbb{K}_{D,6} = \overline{(e_{SM,1} (e_{DM,1} + e_{DM,2})^* e_{SM,2})^*}$$

The drive motor is allowed to stop driving, forward or backward, only if the ride control is in the STOP state or the respective range proximity sensor on the front or back side is in the ACTIVE state. The following two regular languages formulate this specification:

$$\mathbb{K}_{D,7,i} = \overline{((e_{RC,2} + e_{PS,i,2})^* (e_{RC,1} + e_{PS,i,1}) (e_{RC,1} + e_{DM,3} + e_{PS,i,1})^* (e_{RC,2} + e_{PS,i,2}))^*}; i \in \{1,2\}$$

3.3. Requirements for the Buttons

The activation of the AGV is allowed only if either the forward button or the backward button is pressed, while the reset button is not pressed. The following two regular languages formulate this specification:

$$\mathbb{K}_{D,8} = \overline{((e_{B,1,2} + e_{B,2,2})^* (e_{B,1,1} + e_{B,2,1})(e_{B,1,1} + e_{B,2,1} + e_{MM,2})^* (e_{B,1,2} + e_{B,2,2}))^*},$$

$$\mathbb{K}_{D,9} = \overline{((e_{B,3,2} + e_{MM,2})^* e_{B,3,1}(e_{B,3,1})^* e_{B,3,2})^*}$$

The reset of the AGV is allowed only if the reset button is pressed. The following regular language formulates this specification:

$$\mathbb{K}_{D,10} = \overline{((e_{B,3,2})^* e_{B,3,1}(e_{B,3,1} + e_{MM,1})^* e_{B,3,2})^*}$$

3.4. Requirements for the LEDs

The reset mode LEDs are allowed to be switched off only if the AGV is in the ACTIVE or RESET state. The following regular language formulates this specification:

$$\mathbb{K}_{D,11} = \overline{(e_{MM,3})^* (e_{MM,1} + e_{MM,2})(e_{MM,1} + e_{MM,2} + e_{L,3,2})^* e_{MM,3})^*}$$

The reset mode LED is allowed to be turned on only if the AGV is in the EMERGENCY state. The following regular language formulates this specification:

$$\mathbb{K}_{D,12} = \overline{(e_{MM,1} + e_{MM,2})^* e_{MM,3}(e_{L,3,1} + e_{MM,3})^* (e_{MM,1} + e_{MM,2})^*}$$

The LED indicating the forward move and the LED indicating the backward move are allowed to be switched on only if the AGV is in its RESET state. The following regular language formulates this specification:

$$\mathbb{K}_{D,13} = \overline{(e_{MM,1} + e_{MM,3})^* e_{MM,2}(e_{L,1,1} + e_{L,2,1})^* (e_{MM,1} + e_{MM,3})^*}$$

The LEDs indicating the forward and backward moves is allowed to be switched off only if the AGV is in the ACTIVE or EMERGENCY state. The following regular language formulates this specification:

$$\mathbb{K}_{D,14} = \overline{(e_{MM,1} + e_{MM,3})(e_{L,1,2} + e_{L,2,2} + e_{MM,1} + e_{MM,3})^* e_{MM,2})^*}$$

4. Supervisory Control Design

4.1. The Supervisors Realizing the Desired Languages

First, a general two-state supervisor automaton will be developed. This automaton will be used for the design of the supervisors realizing the desired languages, presented in the previous section. The general two-state supervisor automaton is determined to be of the following 6-tuple form:

$$\mathbf{S} = (\mathbb{Q}_S, \mathbb{E}_S, f_S, \mathbb{H}_S, x_{S,0}, \mathbb{Q}_S)$$

where,

$$\begin{aligned} \mathbb{Q}_S &= \{q_{S,1}, q_{S,2}\}, \\ \mathbb{E}_S &= \mathbb{E}_{S,1} \cup \mathbb{E}_{S,2} \cup \mathbb{E}_{S,3} \cup \mathbb{E}_{S,4}, \\ x_{S,0} &= q_{S,1}, f_S(q_{S,1}, e) = q_{S,1}; e \in \mathbb{E}_{S,1}, \\ f_S(q_{S,1}, e) &= q_{S,2}; e \in \mathbb{E}_{S,2}, \\ f_S(q_{S,2}, e) &= q_{S,2}; e \in \mathbb{E}_{S,3}, \\ f_S(q_{S,2}, e) &= q_{S,1}; e \in \mathbb{E}_{S,4}. \end{aligned}$$

The sets of active events are:

$$\mathbb{H}_S(q_{S,1}) = \mathbb{E}_{S,1} \cup \mathbb{E}_{S,2}, \mathbb{H}_S(q_{S,2}) = \mathbb{E}_{S,3} \cup \mathbb{E}_{S,4}.$$

The alphabets $\mathbb{E}_{S,1}$, $\mathbb{E}_{S,2}$, $\mathbb{E}_{S,3}$, and $\mathbb{E}_{S,4}$, will be determined by the respective language realized by the supervisor. The state diagram of **S** is presented in Figure 11.

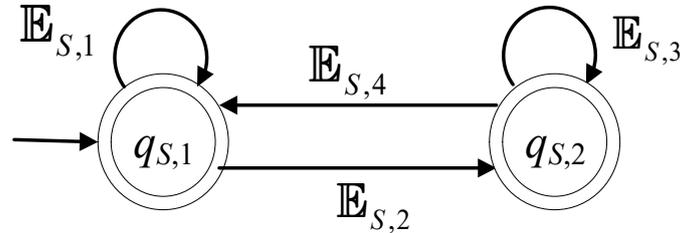


Figure 11. State diagram of **S**.

The complexity triad of the supervisor **S** is:

$$\left(|\mathbb{Q}|, |\mathbb{E}|, \sum_{q \in \mathbb{Q}} |\mathbb{H}| \right) = \left(2, |\mathbb{E}_S|, \sum_{\lambda=1}^4 |\mathbb{H}_S(q_{S,\lambda})| \right)$$

where $|\cdot|$ denotes the cardinality of the argument set. More about the complexity triad of an automaton can be found in [25].

The supervisor **S**₁, realizing the language $\mathbb{K}_{D,1}$, can be developed by the general supervisor **S** through the following,

$$\mathbb{E}_S = \{e_{MM,1}, e_{MM,2}, e_{BL,1}, e_{BL,3}\}, \mathbb{E}_{S,1} = \{e_{MM,1}, e_{MM,2}, e_{BL,3}\}, \\ \mathbb{E}_{S,2} = \{e_{BL,1}\}, \mathbb{E}_{S,3} = \mathbb{E}_{S,2}, \mathbb{E}_{S,4} = \{e_{BL,3}\}.$$

The complexity of **S**₁ is (2, 4, 6).

The supervisor **S**₂, realizing the language $\mathbb{K}_{D,2}$, can be developed by the general supervisor **S** through the following,

$$\mathbb{E}_S = \{e_{MM,1}, e_{MM,2}, e_{BS,1}, e_{BS,2}\}, \mathbb{E}_{S,1} = \{e_{MM,1}, e_{MM,2}, e_{BS,2}\}, \\ \mathbb{E}_{S,2} = \{e_{BS,1}\}, \mathbb{E}_{S,3} = \mathbb{E}_{S,2}, \mathbb{E}_{S,4} = \{e_{BS,2}\}.$$

The complexity of **S**₂ is (2, 4, 6).

The supervisor **S**₃, realizing the language $\mathbb{K}_{D,3}$, can be developed by the general supervisor **S** through the following,

$$\mathbb{E}_S = \{e_{SM,2}, e_{MM,1}, e_{MM,2}, e_{MM,3}\}, \mathbb{E}_{S,1} = \{e_{SM,2}, e_{MM,1}, e_{MM,3}\}, \\ \mathbb{E}_{S,2} = \{e_{MM,2}\}, \mathbb{E}_{S,3} = \emptyset, \mathbb{E}_{S,4} = \{e_{MM,1}, e_{MM,3}\}$$

The complexity of **S**₃ is (2, 4, 6).

The supervisor **S**₄, realizing the language $\mathbb{K}_{D,4}$, can be developed by the general supervisor **S** through the following,

$$\mathbb{E}_S = \{e_{SM,2}, e_{DM,1}, e_{DM,2}, e_{DM,3}, e_{DM,5}\}, \mathbb{E}_{S,1} = \{e_{SM,2}, e_{DM,5}\}, \\ \mathbb{E}_{S,2} = \{e_{DM,1}, e_{DM,2}\}, \mathbb{E}_{S,3} = \emptyset, \mathbb{E}_{S,4} = \{e_{DM,3}, e_{DM,5}\}$$

The complexity of **S**₄ is (2, 5, 6).

The supervisor **S**₅, realizing the language $\mathbb{K}_{D,5}$, can be developed by the general supervisor **S** through the following,

$$\mathbb{E}_S = \{e_{DM,1}, e_{DM,2}, e_{SM,1}, e_{MM,1}, e_{MM,2}, e_{MM,3}\}, \mathbb{E}_{S,1} = \{e_{MM,1}, e_{MM,3}\}, \\ \mathbb{E}_{S,2} = \{e_{MM,2}\}, \mathbb{E}_{S,3} = \{e_{DM,1}, e_{DM,2}, e_{SM,1}\}, \mathbb{E}_{S,4} = \{e_{MM,1}, e_{MM,3}\}$$

The complexity of S_5 is (2, 6, 8).

The supervisor S_6 , realizing the language $\mathbb{K}_{D,6}$, can be developed by the general supervisor S through the following,

$$\mathbb{E}_S = \{e_{DM,1}, e_{DM,2}, e_{SM,1}\}, \mathbb{E}_{S,1} = \emptyset, \mathbb{E}_{S,2} = \{e_{SM,1}\}, \\ \mathbb{E}_{S,3} = \{e_{DM,1}, e_{DM,2}\}, \mathbb{E}_{S,4} = \{e_{SM,2}\}$$

The complexity of S_6 is (2, 3, 4).

For $i \in \{1, 2\}$, the supervisor $S_{7,i}$, realizing the language $\mathbb{K}_{D,7,i}$, can be developed by the general supervisor S through the following,

$$\mathbb{E}_S = \{e_{RC,1}, e_{RC,2}, e_{PS,i,1}, e_{PS,i,2}, e_{DM,3}\}, \mathbb{E}_{S,1} = \{e_{RC,2}, e_{PS,i,2}\}, \mathbb{E}_{S,2} = \{e_{RC,1}, e_{PS,i,1}\}, \\ \mathbb{E}_{S,3} = \{e_{RC,1}, e_{PS,i,1}, e_{DM,3}\},$$

The complexity of $S_{7,i}$ is (2, 5, 9).

The supervisor S_8 , realizing the language $\mathbb{K}_{D,8}$, can be developed by the general supervisor S through the following,

$$\mathbb{E}_S = \{e_{B,1,1}, e_{B,1,2}, e_{B,2,1}, e_{B,2,2}, e_{MM,2}\}, \mathbb{E}_{S,1} = \{e_{B,1,2}, e_{B,2,2}\}, \\ \mathbb{E}_{S,2} = \{e_{B,1,1}, e_{B,2,1}\}, \mathbb{E}_{S,3} = \{e_{B,1,1}, e_{B,2,1}, e_{MM,2}\},$$

The complexity of S_8 is (2, 5, 9).

The supervisor S_9 , realizing the language $\mathbb{K}_{D,9}$, can be developed by the general supervisor S through the following,

$$\mathbb{E}_S = \{e_{B,3,1}, e_{B,3,2}, e_{MM,2}\}, \mathbb{E}_{S,1} = \{e_{B,3,2}, e_{MM,2}\}, \mathbb{E}_{S,2} = \{e_{B,3,1}\}, \\ \mathbb{E}_{S,3} = \mathbb{E}_{S,2}, \mathbb{E}_{S,4} = \{e_{B,3,2}\}$$

The complexity of S_9 is (2, 3, 5).

The supervisor S_{10} , realizing the language $\mathbb{K}_{D,10}$, can be developed by the general supervisor S through the following,

$$\mathbb{E}_S = \{e_{B,3,1}, e_{B,3,2}, e_{MM,1}\}, \mathbb{E}_{S,1} = \{e_{B,3,2}\}, \mathbb{E}_{S,2} = \{e_{B,3,1}\}, \\ \mathbb{E}_{S,3} = \{e_{B,3,1}, e_{MM,1}\},$$

The complexity of S_{10} is (2, 3, 5).

The supervisor S_{11} , realizing the language $\mathbb{K}_{D,11}$, can be developed by the general supervisor S through the following,

$$\mathbb{E}_S = \{e_{MM,1}, e_{MM,2}, e_{MM,3}, e_{L,3,2}\}, \mathbb{E}_{S,1} = \{e_{MM,3}\}, \\ \mathbb{E}_{S,2} = \{e_{MM,1}, e_{MM,2}\}, \mathbb{E}_{S,3} = \mathbb{E}_{S,2} \cup \{e_{L,3,2}\}, \mathbb{E}_{S,4} = \mathbb{E}_{S,1}$$

The complexity of S_{11} is (2, 4, 7).

The supervisor S_{12} , realizing the language $\mathbb{K}_{D,12}$, can be developed by the general supervisor S through the following,

$$\mathbb{E}_S = \{e_{MM,1}, e_{MM,2}, e_{MM,3}, e_{L,3,1}\}, \mathbb{E}_{S,1} = \{e_{MM,1}\}, \mathbb{E}_{S,2} = \{e_{MM,3}\}, \\ \mathbb{E}_{S,3} = \{e_{L,3,1}, e_{MM,3}\}, \mathbb{E}_{S,4} = \{e_{MM,1}, e_{MM,2}\}$$

The complexity of S_{12} is (2, 4, 6).

The supervisor S_{13} , realizing the language $\mathbb{K}_{D,13}$, can be developed by the general supervisor S through the following,

$$\mathbb{E}_S = \{e_{MM,1}, e_{MM,2}, e_{MM,3}, e_{L,1,1}, e_{L,2,1}\}, \mathbb{E}_{S,1} = \{e_{MM,1}, e_{MM,3}\}, \\ \mathbb{E}_{S,2} = \{e_{MM,2}\}, \mathbb{E}_{S,3} = \{e_{L,1,1}, e_{L,2,1}\}, \mathbb{E}_{S,4} = \{e_{MM,1}, e_{MM,3}\}$$

The complexity of S_{13} is (2, 5, 7).

The supervisor \mathbf{S}_{14} , realizing the language $\mathbb{K}_{D,14}$, can be developed by the general supervisor \mathbf{S} through the following,

$$\mathbb{E}_S = \{e_{MM,1}, e_{MM,2}, e_{MM,3}, e_{L,1,2}, e_{L,2,2}\}, \mathbb{E}_{S,1} = \emptyset, \\ \mathbb{E}_{S,2} = \{e_{MM,1}, e_{MM,3}\}, \mathbb{E}_{S,3} = \{e_{L,1,2}, e_{L,2,2}, e_{MM,1}, e_{MM,3}\}, \mathbb{E}_{S,4} = \{e_{MM,2}\}.$$

The complexity of \mathbf{S}_{14} is (2, 5, 7).

4.2. Performance and Properties of the Controlled Automaton

The total automaton of the AGV is the synchronous product of all the automata of the subsystems (devices of the multimover), i.e.,

$$\mathbf{G} = \mathbf{G}_{MM} \parallel \mathbf{G}_{DM} \parallel \mathbf{G}_{SM} \parallel \mathbf{G}_{RC} \parallel \mathbf{G}_{BA} \parallel \left(\prod_{i=1}^2 \mathbf{G}_{PS,i} \right) \parallel \mathbf{G}_{BS} \parallel \left(\prod_{j=1}^3 \mathbf{G}_{L,j} \right) \parallel \left(\prod_{k=1}^3 \mathbf{G}_{B,k} \right) \quad (1)$$

Regarding the synchronous product and its properties, see [9,10]. The marked behavior of the total automaton is:

$$\mathbb{L}_m(\mathbf{G}) = P_{MM}^{-1}(\mathbb{L}_m(\mathbf{G}_{MM})) \cap P_{DM}^{-1}(\mathbb{L}_m(\mathbf{G}_{DM})) \cap P_{SM}^{-1}(\mathbb{L}_m(\mathbf{G}_{SM})) \cap P_{RC}^{-1}(\mathbb{L}_m(\mathbf{G}_{RC})) \cap P_{BA}^{-1}(\mathbb{L}_m(\mathbf{G}_{BA})) \cap \\ P_{BS}^{-1}(\mathbb{L}_m(\mathbf{G}_{BS})) \left(\cap_{i=1}^2 \left(P_{PS,i}^{-1}(\mathbb{L}_m(\mathbf{G}_{PS,i})) \right) \right) \left(\cap_{j=1}^3 \left(P_{L,j}^{-1}(\mathbb{L}_m(\mathbf{G}_{L,j})) \right) \right) \left(\cap_{k=1}^3 \left(P_{B,k}^{-1}(\mathbb{L}_m(\mathbf{G}_{B,k})) \right) \right) \quad (2)$$

where P_{MM} is the projection of \mathbb{E}^* to \mathbb{E}_{MM}^* , P_{DM} is the projection of \mathbb{E}^* to \mathbb{E}_{DM}^* , P_{SM} is the projection of \mathbb{E}^* to \mathbb{E}_{SM}^* , P_{RC} is the projection of \mathbb{E}^* to \mathbb{E}_{RC}^* , P_{BL} is the projection of \mathbb{E}^* to \mathbb{E}_{BL}^* , P_{BS} is the projection of \mathbb{E}^* to \mathbb{E}_{BS}^* , $P_{PS,i}$ is the projection of \mathbb{E}^* to $\mathbb{E}_{PS,i}^*$, $P_{L,j}$ is the projection of \mathbb{E}^* to $\mathbb{E}_{L,j}^*$, $P_{B,k}$ is the projection of \mathbb{E}^* to $\mathbb{E}_{B,k}^*$, and where,

$$\mathbb{E} = \mathbb{E}_{MM} \cup \mathbb{E}_{DM} \cup \mathbb{E}_{SM} \cup \mathbb{E}_{RC} \cup \mathbb{E}_{BL} \cup \mathbb{E}_{BS} \left(\cup_{i=1}^2 \mathbb{E}_{PS,i} \right) \left(\cup_{j=1}^3 \mathbb{E}_{L,j} \right) \left(\cup_{k=1}^3 \mathbb{E}_{B,k} \right)$$

The controlled automaton, namely the automaton resulting after the application of all supervisors to the total automaton, is expressed as follows:

$$\mathbf{G}_c = \mathbf{G} \parallel \left(\prod_{\ell=1}^6 \mathbf{S}_\ell \right) \parallel \left(\prod_{\ell=8}^{14} \mathbf{S}_\ell \right) \parallel \mathbf{S}_{7,1} \parallel \mathbf{S}_{7,2} \quad (3)$$

The marked behavior of the controlled automaton is computed to be:

$$\mathbb{L}_m(\mathbf{G}_c) = \mathbb{L}_m(\mathbf{G}) \cap \left(\cap_{\ell=1}^6 \left(P_i^{-1}(\mathbb{K}_{D,\ell}) \right) \right) \cap \left(\cap_{\ell=8}^{14} \left(P_i^{-1}(\mathbb{K}_{D,\ell}) \right) \right) \cap P_{7,1}^{-1}(\mathbb{K}_{D,7,1}) \cap P_{7,2}^{-1}(\mathbb{K}_{D,7,2}) \quad (4)$$

where P_i is the projection of \mathbb{E}^* to \mathbb{E}_i^* with $i \in \{1, \dots, 6\} \cup \{8, \dots, 14\}$, $P_{7,1}$ is the projection of \mathbb{E}^* to $\mathbb{E}_{7,1}^*$, and $P_{7,2}$ is the projection of \mathbb{E}^* to $\mathbb{E}_{7,2}^*$. From (4), it is observed that the performance of the controlled automaton is the combined influence of all desired languages in the framework of the marked behavior of the AGV.

To guarantee the realizability of all proposed supervisors, regarding the total automaton \mathbf{G} , the results in [29] for the physical realizability will be used. Before presenting the respective, the set of all uncontrollable events is defined to be:

$$\mathbb{E}_{uc} = \mathbb{E}_{MM,uc} \cup \mathbb{E}_{DM,uc} \cup \mathbb{E}_{SM,uc} \cup \mathbb{E}_{RC,uc} \cup \mathbb{E}_{BL,uc} \cup \mathbb{E}_{BS,uc} \left(\cup_{i=1}^2 \mathbb{E}_{PS,i,uc} \right) \left(\cup_{j=1}^3 \mathbb{E}_{L,j,uc} \right) \left(\cup_{k=1}^3 \mathbb{E}_{B,k,uc} \right)$$

Proposition 1. *The supervisors \mathbf{S}_1 to \mathbf{S}_{14} are physically realizable, regarding \mathbf{G} , through (3).*

Proof of Proposition 1. The automata of the supervisors are designed to have all uncontrollable events in the active event sets of all their states. Thus, it holds that $\mathbb{H}_S(q_{S,1}) \cap \mathbb{H}_S(q_{S,2}) \cap \mathbb{E}_{S,uc} = \mathbb{E}_{S,uc}$, where $\mathbb{E}_{S,uc} = \mathbb{E}_{uc} \cap \mathbb{E}_S$. Therefore, using Corollary 1 in [29], it is derived that all supervisors are PR, with respect to \mathbf{G} and through (3). \square

According to [9,10], an automaton is a nonblocking automaton if and only if the prefix closure of its marked behavior is equal to its closed behavior. In the following proposition, the nonblocking property of G_c will be proved without analytic manipulations of the two behaviors. It will be proved by proving that the automata of all subsystems, under the influence of the supervisors, are always allowed to return to or stay in a marked state.

Proposition 2. *The automaton G_c is a nonblocking automaton.*

Proof of Proposition 2. The proof of nonblocking G_c will be formed by proving that the controlled automaton of each subsystem preserves the nonblocking property. The proof of nonblocking of each subsystem under the influence of the associated supervisors, namely the supervisors having common events with the subsystem, will be presented.

The supervisors, having common events with G_{MM} , are $S_1, S_2, S_3, S_5, S_8, S_9, S_{10}, S_{11}, S_{12}, S_{13}$, and S_{14} . In G_{MM} , the transitions from the two non-marked states to the marked ones can take place through the event $e_{MM,3}$. The alphabets of the supervisors S_1, S_2, S_8, S_9 , and S_{10} do not include $e_{MM,3}$. The alphabets of the supervisors $S_3, S_5, S_{11}, S_{12}, S_{13}$, and S_{14} include $e_{MM,3}$ and it holds that $e_{MM,3} \in \mathbb{H}_5(q), (\forall q \in \mathbb{Q}_S)$. Thus, the return of G_{MM} to its marked state is not restricted.

The supervisors, having common events with G_{DM} , are S_4, S_6 , and $S_{7,i}$. In G_{DM} , the transitions from the two non-marked states to the marked can take place through the events $e_{DM,4}$ and $e_{DM,5}$. The alphabets of the supervisors S_6 , and $S_{7,i}$ do not include $e_{DM,4}$ and $e_{DM,5}$. Furthermore, the alphabet of S_4 does not include, $e_{DM,4}$ and it holds that $e_{DM,5} \in \mathbb{H}_{S,4}(q), (\forall q \in \mathbb{Q}_{S,4})$. Thus, the return of G_{MM} to its marked state is not restricted.

The supervisors, having common events with G_{SM} , are S_3, S_4, S_5 , and S_6 . In G_{SM} , the transition from its non-marked state to the marked one can take place through the events $e_{SM,2}$ and $e_{SM,3}$. The alphabets of the supervisors S_3, S_4, S_5 , and S_6 . do not include $e_{SM,3}$. Thus, the transition of G_{SM} to a marked state is not obstructed.

The supervisor, having common events with G_{RC} , is $S_{7,i}$. In G_{RC} , the transition from the non-marked state to the marked one can take place through the event $e_{RC,2}$. For $S_{7,i}$ it holds that $e_{RC,2} \in \mathbb{H}_{S,7,i}(q), (\forall q \in \mathbb{Q}_{S,7,i})$. Thus, the transition of G_{RC} to a marked state is not obstructed.

The supervisor, having common events with G_{BL} , is S_1 . In G_{BA} , the transition from its non-marked states to the marked one can take place through the event $e_{BL,3}$. Furthermore, in G_{BA} , the event $e_{BL,2}$ triggers the transition from one non-marked state to the other. For S_1 it holds that the event $e_{BL,2}$ is not in the alphabet of S_1 and $e_{BL,3} \in \mathbb{H}_{S,1}(q), (\forall q \in \mathbb{Q}_{S,1})$. Thus, the transition of G_{RC} to a marked state is not obstructed.

The supervisor, having common events with $G_{PS,i}$, is $S_{7,i}$. In $G_{PS,i}$, the transition from the non-marked state to the marked one can take place through the event $e_{PS,i,2}$. For $S_{7,i}$ it holds that $e_{PS,i,2} \in \mathbb{H}_{S,7,i}(q), (\forall q \in \mathbb{Q}_{S,7,i})$. Thus, the transition of $G_{PS,i}$ to a marked state is not obstructed.

The supervisor, having common events with G_{BS} , is S_2 . In G_{BS} , the transition from the non-marked state to the marked one can take place through the event $e_{BS,2}$. For S_2 it holds that $e_{BS,2} \in \mathbb{H}_{S,2}(q), (\forall q \in \mathbb{Q}_{S,2})$. Thus, the transition of G_{BS} to a marked state is not obstructed.

The supervisors, having common events with $G_{L,1}$ and $G_{L,2}$, are S_{13} and S_{14} . In $G_{L,1}$ and $G_{L,2}$, the transition from the non-marked states to the marked ones can take place through the events $e_{L,1,2}$ and $e_{L,2,2}$. The alphabet of S_{13} does not include $e_{L,1,2}$ and $e_{L,2,2}$. For S_{14} it holds that $\{e_{L,1,2}, e_{L,2,2}\} \notin \mathbb{H}_{S,14}(q_{S,14,1})$ and $\{e_{L,1,2}, e_{L,2,2}\} \in \mathbb{H}_{S,14}(q_{S,14,2})$. Thus, the transition from $q_{S,14,1}$ to $q_{S,14,2}$ is always feasible through the uncontrollable event $e_{MM,3}$.

The supervisors, having common events with $G_{L,3}$, are S_{11} and S_{12} . In $G_{L,3}$, the transition from the non-marked state to the marked one can take place through the event $e_{L,3,2}$. The alphabet of S_{12} does not include $e_{L,3,2}$. For S_{11} it holds that $e_{L,3,2} \in \mathbb{H}_{S,11}(q_{S,11,1})$ and $e_{L,3,2} \notin \mathbb{H}_{S,11}(q_{S,11,2})$. Thus, the transition from $q_{S,12,2}$ to $q_{S,12,1}$ is always feasible through the uncontrollable event $e_{MM,3}$.

The supervisor, having common events with $G_{B,1}$ and $G_{B,2}$, is S_8 . In $G_{B,1}$ and $G_{B,2}$, the transition from the non-marked states to the marked ones can take place through the events $e_{B,1,2}$ and $e_{B,2,2}$. For S_8 it holds that $\{e_{B,1,2}, e_{B,2,2}\} \in \mathbb{H}_{S,8}(q)$, $(\forall q \in \mathbb{Q}_{S,8})$. Thus, the transition of $G_{B,1}$ to a marked state and the transition of $G_{B,2}$ to a marked state are not obstructed.

The supervisors, having common events with $G_{B,3}$, are S_9 and S_{10} . In $G_{B,3}$, the transition from the non-marked state to the marked one can take place through the event $e_{B,3,2}$. For S_9 and S_{10} it holds that $\{e_{B,3,2}, e_{B,3,2}\} \in \mathbb{H}_S(q)$, $(\forall q \in \mathbb{Q}_S)$. Thus, the transition of $G_{B,3}$ to a marked state is not obstructed.

Finally, it is important to mention that the alphabets of the subsystem are disjoint sets. So, the examination of the nonblocking for each individual subsystem is sufficient to prove the nonblocking of the total controlled system. \square

Remark 1. Using the complexities of the supervisors, determined in Section 4.1, the complexity of the total supervisor control scheme is computed to be.

$$\left(\sum_{\substack{\lambda=1 \\ \lambda \neq 7}}^{14} |\mathbb{Q}_{S,\lambda}| + \sum_{\lambda=1}^2 |\mathbb{Q}_{S,7,\lambda}|, \sum_{\lambda=1}^{14} |\mathbb{E}_{S,\lambda}| + \sum_{\lambda=1}^2 |\mathbb{E}_{S,7,\lambda}|, \sum_{\lambda=1}^{14} \left(\sum_{\mu=1}^2 |\mathbb{H}_{S,\lambda}(q_{S,\lambda,\mu})| \right) + \sum_{\lambda=1}^2 \left(\sum_{\mu=1}^2 |\mathbb{H}_{S,\lambda}(q_{S,7,\lambda,\mu})| \right) \right) = (30, 65, 100)$$

5. Implementation Issues of the Two-State Supervisor Automata

As already mentioned, the main advantage of the proposed supervisor architecture is the simple two-state structure of all designed supervisors. Moreover, this simple structure of the supervisors facilitates the adaptation of programming methods, guaranteeing the secure functionality of the program. For instance, according to [30–32], the avalanche effect in PLC’s control program is the sequential state transition in the supervisor implementation program, triggered by a single event occurrence. The avalanche effect problem in PLC control programs, which can be faced with a simple inversion of the appearance of the states in the program, can easily be handled in the proposed scheme. Furthermore, the structure of the present supervisors can be easily implemented in various programming languages, such as Visual Basic, Ladder Diagrams, Structured Text etc., making them suitable for SCADA, PLC, and microcontroller environments ([33,34]).

As already mentioned, the implementability of the proposed scheme is guaranteed by the controllability of the desired languages via the physical realizability of the synchronous product of the system’s automaton and the supervisors. All supervisors are implemented in the Control Unit of the system. The input signals of the Control Unit are the signals corresponding to the uncontrollable events and the signals (commands) corresponding to the controllable events. The signals outputted by the Control Unit are the controllable events. The controllable events, ending to multimover, are the commands of multimover. The signals corresponding to the uncontrollable events ending with multimover directly without passing through the Control Unit. In Figure 12, a block diagram of the controlled automaton is presented, where the controllable and uncontrollable events are connected to the system automaton and the control unit.

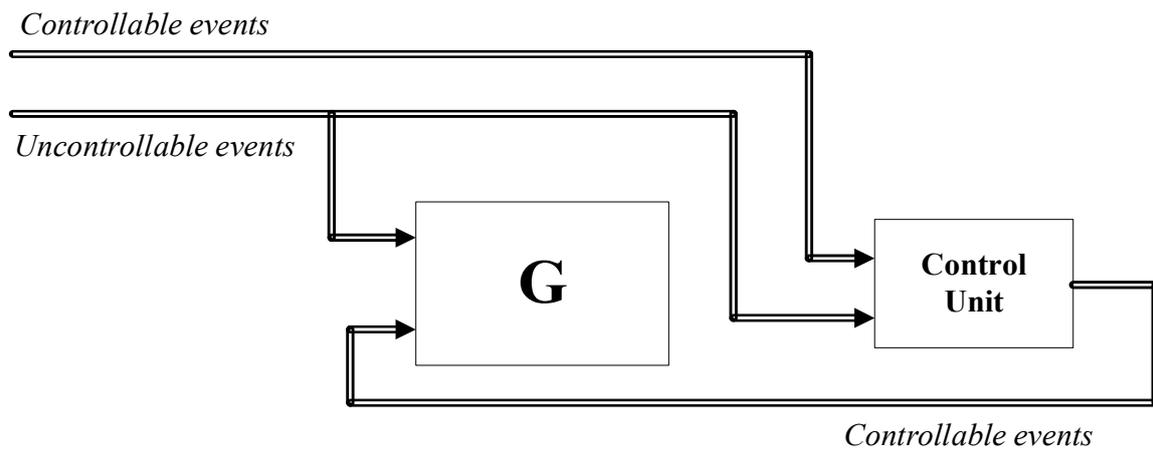


Figure 12. Block diagram of the controlled automaton.

6. Simulation

A simulation of the controlled automaton will be presented for the validation of the proposed control scheme. Consider the following word:

$$e_{B,3,1}e_{MM,1}e_{B,3,2}e_{MM,2}e_{B,1,1}e_{MM,2}e_{DM,1}e_{SM,1}e_{DM,1}e_{DM,3}e_{RC,1}e_{DM,3}$$

It is clear that only events of the subsystems Button 1, Button 3, Operation, Drive Motor, Steer Motor, and ride control take part in the word, thus only the behavior of the six subsystems will be investigated. In Table 1, the transitions of the automata of the Button 1 subsystem, Button 3 subsystem, Operation subsystem, the Drive Motor subsystem, the Steer Motor subsystem, and the ride control subsystem are presented. The three shadowed lines represent the three lines where the transitions to the respective automata are not triggered, as they are restricted by the supervisor scheme.

Table 1. Transitions of the Controlled Automaton.

Events	Button 1	Button 3	Operation	Drive Motor	Steer Motor	Ride Control
—		$q_{B,3,1}$	$q_{MM,1}$	$q_{DM,1}$	$q_{SM,1}$	$q_{RC,1}$
$e_{B,3,1}$	$q_{B,1,1}$	$q_{B,3,2}$	$q_{MM,1}$	$q_{DM,1}$	$q_{SM,1}$	$q_{RC,1}$
$e_{MM,1}$	$q_{B,1,1}$	$q_{B,3,2}$	$q_{MM,2}$	$q_{DM,1}$	$q_{SM,1}$	$q_{RC,1}$
$e_{B,3,2}$	$q_{B,1,1}$	$q_{B,3,1}$	$q_{MM,2}$	$q_{DM,1}$	$q_{SM,1}$	$q_{RC,1}$
$e_{MM,2}$	$q_{B,1,1}$	$q_{B,3,1}$	$q_{MM,2}$	$q_{DM,1}$	$q_{SM,1}$	$q_{RC,1}$
$e_{B,1,1}$	$q_{B,1,2}$	$q_{B,3,1}$	$q_{MM,2}$	$q_{DM,1}$	$q_{SM,1}$	$q_{RC,1}$
$e_{MM,2}$	$q_{B,1,2}$	$q_{B,3,1}$	$q_{MM,3}$	$q_{DM,1}$	$q_{SM,1}$	$q_{RC,1}$
$e_{DM,1}$	$q_{B,1,2}$	$q_{B,3,1}$	$q_{MM,3}$	$q_{DM,1}$	$q_{SM,1}$	$q_{RC,1}$
$e_{SM,1}$	$q_{B,1,2}$	$q_{B,3,1}$	$q_{MM,3}$	$q_{DM,1}$	$q_{SM,2}$	$q_{RC,1}$
$e_{DM,1}$	$q_{B,1,2}$	$q_{B,3,1}$	$q_{MM,3}$	$q_{DM,2}$	$q_{SM,2}$	$q_{RC,1}$
$e_{DM,3}$	$q_{B,1,2}$	$q_{B,3,1}$	$q_{MM,3}$	$q_{DM,2}$	$q_{SM,2}$	$q_{RC,1}$
$e_{RC,1}$	$q_{B,1,2}$	$q_{B,3,1}$	$q_{MM,3}$	$q_{DM,2}$	$q_{SM,2}$	$q_{RC,2}$
$e_{DM,3}$	$q_{B,1,2}$	$q_{B,3,1}$	$q_{MM,3}$	$q_{DM,1}$	$q_{SM,2}$	$q_{RC,2}$

Initially, the six automata are in states $q_{B,1,1}$ (Button 1), $q_{B,3,1}$ (Button 3), $q_{MM,1}$ (Operation subsystem), $q_{DM,1}$ (Drive Motor), $q_{SM,1}$ (Steer Motor), and $q_{RC,1}$ (ride control). Furthermore, all supervisors are in their initial state. After the occurrence of the event $e_{B,3,1}$ the six automata are in states $q_{B,1,1}$, $q_{B,3,2}$, $q_{MM,1}$, $q_{DM,1}$, $q_{SM,1}$ and $q_{RC,1}$ (Line 2 of Table 1). The supervisors having the event $e_{B,3,1}$ in their alphabets are S_9 and S_{10} . The event is active

to the current states of both supervisors. Hence, after the occurrence of $e_{B,3,1}$, the new states of the two supervisors are $q_{S,9,2}$ and $q_{S,10,2}$, respectively.

After the occurrence of $e_{MM,1}$, the six automata are in states $q_{B,1,1}$, $q_{B,3,2}$, $q_{MM,2}$, $q_{DM,1}$, $q_{SM,1}$, and $q_{RC,1}$ (Line 3 of Table 1). The supervisors having $e_{MM,1}$ in their alphabets are S_1 , S_2 , S_3 , S_5 , S_{10} , S_{11} , S_{12} , S_{13} , and S_{14} . The event is active to the current states of all supervisors being $q_{S,1,1}$, $q_{S,2,1}$, $q_{S,3,1}$, $q_{S,5,1}$, $q_{S,10,2}$, $q_{S,11,1}$, $q_{S,12,1}$, $q_{S,13,1}$, and $q_{S,14,1}$, respectively. Hence, after the occurrence of $e_{MM,1}$, the new states of the supervisors are $q_{S,1,1}$, $q_{S,2,1}$, $q_{S,3,1}$, $q_{S,5,1}$, $q_{S,10,2}$, $q_{S,11,2}$, $q_{S,12,1}$, $q_{S,13,1}$, and $q_{S,14,2}$, respectively.

After the occurrence of the event $e_{B,3,2}$, the six automata are in states $q_{B,1,1}$, $q_{B,3,1}$, $q_{MM,2}$, $q_{DM,1}$, $q_{SM,1}$, and $q_{RC,1}$ (Line 4 of Table 1). The supervisors having $e_{B,3,2}$ in their alphabets are S_9 and S_{10} . The event is active in the current states of both supervisors being $q_{S,9,2}$ and $q_{S,10,2}$, respectively. Hence, after the occurrence of $e_{B,3,2}$, the new states of the supervisors are $q_{S,9,1}$ and $q_{S,10,1}$, respectively.

After the occurrence of $e_{MM,2}$, the six automata remain in states $q_{B,1,1}$, $q_{B,3,1}$, $q_{MM,2}$, $q_{DM,1}$, $q_{SM,1}$, and $q_{RC,1}$ (Line 5 of Table 1). The supervisors having $e_{MM,2}$ in their alphabets are S_1 , S_2 , S_3 , S_5 , S_8 , S_9 , S_{11} , S_{12} , S_{13} , and S_{14} . The event $e_{MM,2}$ is not active in the current state of supervisor S_8 , i.e., $e_{MM,2} \notin \mathbb{H}(q_{S,8,1})$. Hence, after the occurrence of $e_{MM,2}$, the supervisors remain in the states $q_{S,1,1}$, $q_{S,2,1}$, $q_{S,3,1}$, $q_{S,5,1}$, $q_{S,8,1}$, $q_{S,11,1}$, $q_{S,11,2}$, $q_{S,12,1}$, $q_{S,13,1}$, and $q_{S,14,2}$, respectively.

After the occurrence of $e_{B,1,1}$, the six automata are in states $q_{B,1,2}$, $q_{B,3,1}$, $q_{MM,2}$, $q_{DM,1}$, $q_{SM,1}$, and $q_{RC,1}$ (Line 6 of Table 1). The supervisor having the event $e_{B,1,1}$ in its alphabet is S_8 . The event is active in the current state of the supervisor being $q_{S,8,1}$. Hence, after the occurrence of $e_{B,1,1}$, the new state of the supervisor is $q_{S,8,2}$.

After the occurrence of the event $e_{MM,2}$, the six automata are in states $q_{B,1,2}$, $q_{B,3,1}$, $q_{MM,3}$, $q_{DM,1}$, $q_{SM,1}$, and $q_{RC,1}$ (Line 7 of Table 1). The supervisors having $e_{MM,2}$ in their alphabets are S_1 , S_2 , S_3 , S_5 , S_8 , S_9 , S_{11} , S_{12} , S_{13} , and S_{14} . The event is active in the current states of all supervisors being $q_{S,1,1}$, $q_{S,2,1}$, $q_{S,3,1}$, $q_{S,5,1}$, $q_{S,8,2}$, $q_{S,11,1}$, $q_{S,11,2}$, $q_{S,12,1}$, $q_{S,13,1}$, and $q_{S,14,2}$. Hence, after the occurrence of $e_{MM,2}$, the new states of the supervisors are $q_{S,2,1}$, $q_{S,3,2}$, $q_{S,5,2}$, $q_{S,8,2}$, $q_{S,11,1}$, $q_{S,11,2}$, $q_{S,12,1}$, $q_{S,13,1}$, and $q_{S,14,2}$, respectively.

After the occurrence of $e_{DM,1}$, the six automata remain in states $q_{B,1,2}$, $q_{B,3,1}$, $q_{MM,3}$, $q_{DM,1}$, $q_{SM,1}$, and $q_{RC,1}$ (Line 8 of Table 1). The supervisors having $e_{DM,1}$ in their alphabets are S_4 , S_5 , and S_6 . The event $e_{MM,2}$ is not active in the current state of supervisor S_6 , i.e., $e_{DM,1} \notin \mathbb{H}(q_{S,6,1})$. Hence, after the occurrence of $e_{DM,1}$, the supervisors remain in the states $q_{S,4,1}$, $q_{S,5,2}$, and $q_{S,6,1}$, respectively.

After the occurrence of $e_{SM,1}$, the six automata are in states $q_{B,1,2}$, $q_{B,3,1}$, $q_{MM,3}$, $q_{DM,1}$, $q_{SM,2}$, and $q_{RC,1}$ (Line 9 of Table 1). The supervisors having event $e_{SM,1}$ in their alphabet are S_5 , and S_6 . The event is active the current states of both supervisors being $q_{S,5,2}$ and $q_{S,6,1}$. Hence, after the occurrence of $e_{SM,1}$, the new states of the supervisors are $q_{S,5,2}$ and $q_{S,6,2}$, respectively.

After the occurrence of $e_{DM,1}$, the six automata are in states $q_{B,1,2}$, $q_{B,3,1}$, $q_{MM,3}$, $q_{DM,2}$, $q_{SM,2}$, and $q_{RC,1}$ (Line 9 of Table 1). The supervisors having event $e_{DM,1}$ in their alphabets are S_4 , S_5 , and S_6 . The event is active in the current states of all supervisors being $q_{S,4,1}$, $q_{S,5,2}$, and $q_{S,6,2}$. Hence, after the occurrence of $e_{DM,1}$, the new states of the supervisors are $q_{S,4,2}$, $q_{S,5,2}$, and $q_{S,6,2}$, respectively.

After the occurrence of $e_{DM,3}$, the six automata remains in states $q_{B,1,2}$, $q_{B,3,1}$, $q_{MM,3}$, $q_{DM,2}$, $q_{SM,2}$, and $q_{RC,1}$ (Line 10 of Table 1). The supervisors having event $e_{DM,3}$ in their alphabet are S_4 and $S_{7,i}$. The event $e_{DM,3}$ is not active in supervisors $S_{7,i}$, i.e., $e_{DM,3} \notin \mathbb{H}(q_{S,7,i,1})$. Hence, after the occurrence of $e_{DM,3}$, the supervisors remain in the states $q_{S,4,2}$ and $q_{S,7,i,1}$, respectively.

After the occurrence of $e_{RC,1}$, the six automata are in states $q_{B,1,2}$, $q_{B,3,1}$, $q_{MM,3}$, $q_{DM,2}$, $q_{SM,2}$, and $q_{RC,2}$ (Line 11 of Table 1). The supervisor having the event $e_{RC,1}$ in its alphabet is $S_{7,i}$. The event is active in the current state of the supervisor being $q_{S,7,i,1}$. Hence, after the occurrence of $e_{RC,1}$, the new state of the supervisor is $q_{S,7,i,2}$.

After the occurrence of $e_{DM,3}$, the six automata are in states $q_{B,1,2}$, $q_{B,3,1}$, $q_{MM,3}$, $q_{DM,1}$, $q_{SM,2}$, and $q_{RC,2}$ (Line 12 of Table 1). The supervisors having event $e_{DM,3}$ in their alphabet are S_4 and $S_{7,i}$. The event is active in the current states of both supervisors being $q_{S,4,2}$ and $q_{S,7,i,2}$. Hence, after the occurrence of $e_{DM,3}$, the new states of the supervisors are $q_{S,4,1}$ and $q_{S,7,i,2}$, respectively.

7. Conclusions

A Cyberphysical system for several and diverse applications, including product transfer in manufacturing units, has been presented. The system is an industrial AGV. All subsystems of the AGV have been presented using discrete event systems. The desired behavior of the system has been expressed in the form of rules and consequently formulated in the form of regular languages. A set of supervisor automata, realizing the desired languages and being simple two-state automata, has been designed. This way, the implementation of the proposed supervisory architecture can be easily implemented in any language and computer system. The property of the controllability of the desired languages, via the physical realizability of the synchronous product of the total system's automaton and the automata of the supervisors, is proven. The nonblocking property of all the total controlled system automaton is guaranteed. Finally, indicative simulation results of the controlled automaton are presented.

The implementation of the proposed supervisory architecture in experimental environments is under investigation. The problem of collision avoidance of AGVs in industrial environments, which is of great importance, is in the future concerns of the authors.

Author Contributions: Conceptualization, F.N.K. and D.G.F.; methodology, F.N.K. and D.G.F.; validation, D.G.F., G.E.P. and E.M.; investigation, F.N.K. and D.G.F.; resources, G.E.P.; writing—original draft preparation, F.N.K., D.G.F. and G.E.P.; writing—review and editing, D.G.F., G.E.P. and E.M.; visualization, G.E.P. and E.M.; supervision, F.N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

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

References

1. National Science Foundation. Cybermanufacturing Systems. 15 November 2015. Available online: <http://www.nsf.gov/pubs/2015/nsf15061/nsf15061.jsp> (accessed on 31 May 2023).
2. Pivoto, D.G.S.; de Almeida, L.F.F.; da Rosa Righi, R.; Rodrigues, J.J.P.C.; Baratella Lugli, A.; Alberti, A.M. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *J. Manuf. Syst.* **2021**, *58*, 176–192. [CrossRef]
3. Xu, L.D.; Duan, L. Big data for cyber physical systems in industry 4.0: A survey. *Enterp. Inf. Syst.* **2019**, *13*, 148–169. [CrossRef]
4. Jeschke, S.; Brecher, C.; Meisen, T.; Özdemir, D.; Eschert, T. Industrial Internet of Things and Cyber Manufacturing Systems. In *Industrial Internet of Things*; Jeschke, S., Brecher, C., Song, H., Rawat, D., Eds.; Springer Series in Wireless Technology; Springer: Cham, Switzerland, 2017. [CrossRef]
5. Ahmed, U.; Lin, J.C.-W.; Srivastava, G. Exploring the Potential of Cyber Manufacturing Systems in the Digital Age. *ACM Trans. Internet Technol.* **2023**, *4*, 1–38. [CrossRef]
6. Reniers, M.; van de Mortel-Fronczak, J.; Roelofs, K. Model-Based Engineering of Supervisory Controllers for Cyber-Physical Systems. In *Industrial Internet Things*; Springer Series in Wireless Technology; Springer: Cham, Switzerland, 2017. [CrossRef]
7. Yin, X.; Li, S. Recent advances on formal methods for safety and security of cyber-physical systems. *Control Theory Technol.* **2020**, *18*, 459–461. [CrossRef]
8. Ferrari, A.; Ter Beek, M.H. Formal Methods in Railways: A Systematic Mapping Study. *ACM Comput. Surv.* **2022**, *55*, 37. [CrossRef]
9. Wonham, W.M.; Kai, C. *Supervisory Control of Discrete-Event Systems*; Springer: Cham, Switzerland, 2019. [CrossRef]
10. Cassandras, C.G.; Lafortune, S. *Introduction to Discrete Event Systems*, 3rd ed.; Springer: Cham, Switzerland, 2021. [CrossRef]

11. Komenda, J.; Masopust, T. Hierarchical Supervisory Control Under Partial Observation: Normality. *IEEE Trans. Autom. Control* **2023**, *68*, 7286–7298. [[CrossRef](#)]
12. Komenda, J.; Masopust, T.; van Schuppen, J.H. Synthesis of controllable and normal sublanguages for discrete-event systems using a coordinator. *Syst. Control Lett.* **2011**, *60*, 492–502. [[CrossRef](#)]
13. Meira-Góes, R.; Kang, E.; Kwong, R.H.; Lafortune, S. Synthesis of sensor deception attacks at the supervisory layer of Cyber-Physical Systems. *Automatica* **2020**, *121*, 109172. [[CrossRef](#)]
14. Fritz, R.; Zhang, P. Detection and Localization of Stealthy Cyber Attacks in Cyber-Physical Discrete Event System. *IEEE Trans. Autom. Control* **2023**, *68*, 7895–7902. [[CrossRef](#)]
15. Yu, Z.; Duan, X.; Cong, X.; Li, X.; Zheng, L. Detection of Actuator Enablement Attacks by Petri Nets in Supervisory Control Systems. *Mathematics* **2023**, *11*, 943. [[CrossRef](#)]
16. Wang, Y.; Weyrich, M. Towards a novel learning assistant for networked automation systems. In *Machine Learning for Cyber Physical Systems*; Niggemann, O., Beyerer, J., Eds.; Technologien für die Intelligente Automation; Springer: Berlin/Heidelberg, Germany, 2016. [[CrossRef](#)]
17. Tran, N.-H.; Park, H.-S.; Nguyen, Q.-V.; Hoang, T.-D. Development of a Smart Cyber-Physical Manufacturing System in the Industry 4.0 Context. *Appl. Sci.* **2019**, *9*, 3325. [[CrossRef](#)]
18. Dafflon, B.; Moalla, N.; Ouzrout, Y. The challenges, approaches, and used techniques of CPS for manufacturing in Industry 4.0: A literature review. *Int. J. Adv. Manuf. Technol.* **2021**, *113*, 2395–2412. [[CrossRef](#)]
19. Maza, S. Hybrid supervisory-based architecture for robust control of Bi-directional AGVs. *Comput. Ind.* **2023**, *144*, 103797. [[CrossRef](#)]
20. Bolbhat, S.S.; Bhosale, A.S.; Sakthivel, G.; Saravanakumar, D.; Sivakumar, R.; Lakshmipathi, J. Intelligent Obstacle Avoiding AGV Using Vector Field Histogram and Supervisory Control. *J. Phys. Conf. Ser.* **2020**, *1716*, 012030. [[CrossRef](#)]
21. Forschelen, S.T.; van de Mortel-Fronczak, J.M.; Su, R.; Rooda, J.E. Application of supervisory control theory to theme park vehicles. *Discret. Event Dyn. Syst.* **2012**, *22*, 511–540. [[CrossRef](#)]
22. Forschelen, S. Supervisory Control of Theme Park Vehicles. Master's Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2010.
23. Baeten, J.C.M.; van de Mortel-Fronczak, J.M.; Rooda, J.E. Integration of Supervisory Control Synthesis in Model-Based Systems Engineering. In *Complex Systems; Studies in Systems, Decision and Control*; Springer: Cham, Switzerland, 2016; Volume 55. [[CrossRef](#)]
24. ETF Group Ride-Systems. Available online: <https://www.etf.nl> (accessed on 13 December 2023).
25. Kouvakas, N.D.; Koumboulis, F.N.; Fragkoulis, D.G.; Souliotis, A. Modular Supervisory Control for the Coordination of a Manufacturing Cell with Observable Faults. *Sensors* **2022**, *23*, 163. [[CrossRef](#)]
26. Koumboulis, F.N.; Fragkoulis, D.G.; Kalkanias, I.; Fragulis, G.F. Supervisor Design for a Pressurized Reactor Unit in the Presence of Sensor and Actuator Faults. *Electronics* **2022**, *11*, 2534. [[CrossRef](#)]
27. Koumboulis, F.N.; Fragkoulis, D.G.; Michos, A.A. Modular Supervisory Control for multi-floor manufacturing processes. *Control Theory Technol.* **2023**, *21*, 148–160. [[CrossRef](#)]
28. Koumboulis, F.N.; Fragkoulis, D.G.; Georgakopoulos, P. A Distributed Supervisor architecture for a General Wafer Production System. *Sensors* **2023**, *23*, 4545. [[CrossRef](#)]
29. Koumboulis, F.N.; Fragkoulis, D.G.; Arapakis, S. Supervisor design for an assembly line in the presence of faults. In Proceedings of the 27th IEEE International Conference on Emerging Technology and Factory Automation, Stuttgart, Germany, 6–9 September 2022. [[CrossRef](#)]
30. Vieira, A.D.; Santos, E.A.P.; de Queiroz, M.H.; Leal, A.B.; de Paula Neto, A.D.; Cury, J.E.R. A Method for PLC Implementation of Supervisory Control of Discrete Event Systems. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 175–191. [[CrossRef](#)]
31. Prenzel, L.; Provost, J. PLC implementation of symbolic, modular supervisory controllers. *IFAC-PapersOnLine* **2018**, *51*, 304–309. [[CrossRef](#)]
32. James, L.D.; Teixeira, C.A.; Leal, A.B. Formal Design and Implementation of Supervisory Controller for a Didactic Manufacturing Cell. In Proceedings of the 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, VIC, Australia, 13–15 February 2019. [[CrossRef](#)]
33. Scotti, W.A.; Portilla, N.B.; de Queiroz, M.H.; Cury, J.E.R. Modular and systematic design of supervisory control system integrating PLC, SCADA and task routing for a modular production system. In Proceedings of the 2015 IEEE International Conference on Automation Science and Engineering (CASE), Gothenburg, Sweden, 24–28 August 2015. [[CrossRef](#)]
34. Alves, M.V.S.; Carvalho, L.K.; Basilio, J.C. Supervisory Control of Networked Discrete Event Systems With Timing Structure. *IEEE Trans. Autom. Control* **2020**, *66*, 2206–2218. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.