EXAMINING THE COLLISION AVOIDANCE PROBLEM OF AGVS IN A DISCRETE EVENT SIMULATION ENVIRONMENT

Róbert SKAPINYECZ[1]

[1] *University of Miskolc, Institute of Logistics, H-3515 Miskolc-Egyetemváros Miskolc, Hungary, E-mail: robert.skapinyecz@uni-miskolc.hu*

ABSTRACT: *Advanced Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs) are becoming standard material handling machines in modern Industry 4.0 based logistics and production environments. As a result, the simulation modeling of the operation of such highly autonomous devices is also becoming an important task, especially since the use of process simulations as digital twins is one of the cornerstones of the Industry 4.0 principle.*

The direct goal of this article is to compare two rule-based collision avoidance algorithms that may have some relevance in modeling the previously mentioned machines in discrete event simulation environments. The focus is on preventing collision situations forming within a short distance, while the vehicles follow the shortest paths to their destinations. The analysis is realised with the use of Siemens Tecnomatix Plant Simulation, a widely applied discrete event simulation tool. The applied evaluation method could be improved and extended to further develop the proposed modelling approach, with the possible future inclusion of machine learning to fully solve the collision avoidance problem within the introduced framework.

KEYWORDS*: AGV, AMR, discrete event simulation, collision avoidance, Industry 4.0*

1 INTRODUCTION

In the recent period, advanced Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs) became standard material handling equipment in modern Industry 4.0 based production and logistics environments. Unlike their predecessors, these machines can perform a variety of tasks with high flexibility and autonomy, usually eliminating the need for fixed tracks and instead relying on novel methods of free navigation for pathfinding and collision avoidance. Their widespread application falls in line with the continuously increasing trend of automation in the Industry 4.0 principle.

With the proliferation of the previously mentioned material handling machines, eventually they also have to be included more and more frequently in manufacturing and logistics process simulations as well, as these types of modelling methods also play an important role in the Industry 4.0 concept, both as planning tools and as so called "digital twins". However, this poses a significant question: how the behaviour of the aforementioned autonomous vehicles can be precisely and effectively modelled in a discrete event simulation without the need to completely emulate their hardware and software? This is a significant question, as it is usually not realistic to fully represent the internal software and all hardware components of every autonomous device of the real-world system in a discrete event simulation environment, due to computational and other practical constraints. Therefore, an optimal solution to solve this problem with relatively smaller resource-requirements and less complexity would be highly desirable.

Considering the above, the main goal of the current paper is to introduce a modelling approach for testing different algorithms in order to solve the collision avoidance problem when using multiple AGVs and AMRs in an industrial process simulation environment. Of course, collision avoidance between the vehicles is just one part of the given modelling problem, but it is probably the hardest one, as there are practically an unlimited number of possible collision scenarios. The research is conducted with the use of Siemens Tecnomatix Plant Simulation, which is one of the most frequently utilized discrete event simulation software in the industry (discrete event simulation itself is one of the most widely used forms of process simulation). In the current paper, two relatively straightforward rule-based algorithms are going to be compared through simulation modelling, with the future goal of expanding upon the results and develop a machine-learning based solution to broadly solve the generalized collision avoidance problem within the discrete event simulation paradigm.

2 LITERATURE REVIEW

Using various simulation methods itself is a standard practice for modelling the operation of AGVs and AMRs. For example, according to the literature review presented by Fragapane et. al., in 59 papers out of the examined 130 the simulation method was used in relation to the various sub-problems of the planning and control of AMRs in intralogistics environments (Fragapane et. al., 2021). For this reason, the following representative literature review mainly includes such papers that mostly deal with the navigation and collision avoidance problem, and at least partially apply the simulation method.

Turhanlar et. al. used an agent-based simulation approach to model a flexible AMR system in a warehouse environment (Turhanlar et. al., 2022). The results showed that the flexible system, in which the individual vehicles can communicate with each other and can thus make smart decisions, offers significant improvements over a non-flexible design (where collision avoidance is realized by dedicating a single vehicle to a specific operational zone). Praserttaweelap et. al. used particle swarm optimization for solving the pathfinding and collision avoidance problem in a manufacturing layout (Praserttaweelap et. al., 2019). The Matlab software was used for the implementation of the simulation model. A Matlab based simulation was also used by Thai et. al. to model the navigation of an AGV in a flexible manufacturing environment with a matrix layout (Thai et. al., 2021). The pathfinding was realized with an improved form of Dijkstra's algorithm, while a solution for collision detection and avoidance was also developed. A discrete event simulation approach using the Delmia QUEST software was utilized by Zając and Małopolski to model a newly developed collision and deadlock resolution control policy in multi-AGV systems (Zając & Małopolski, 2021). The results demonstrated the efficiency of the presented solution compared to other methods. Stączek et. al. used a digital twin approach for modeling an AMR and its operational environment, with the purpose to aid the preliminary development of the vehicle, which was followed by real-life trials (Stączek et. al., 2021). The simulation model was created with the Gazebo software. A decentralized sensor-level collision-avoidance policy for multi-robot systems was introduced based on deep reinforcement learning by Fan et. al. (Fan et. al., 2020). The policy was trained on data generated through simulation. A reinforcement-learning-based approach was also utilized by Kozjek et. al. for the purpose of route generation for heavy-traffic autonomous mobile robot systems (Kozjek et. al., 2021). A physics-based simulation was developed using the robot operating system (ROS) and the Gazebo simulator to test the developed algorithm and compare it with a baseline method. Feng et. al. also used deep reinforcement learning to solve the collision avoidance problem (Feng et. al., 2021). Here, simulations were developed and used both for the training of the neural network and to model the kinematic characteristics of the robot design, which was also implemented in real life. The long-term goal of the research presented in the paper is to develop a novel robotic system based on Self-reconfigurable and Transformable Omni-Directional Robotic Modules (STORM). Grosset et. al. presented a method to estimate positions of autonomous industrial vehicles (AIVs) moving in a closed industrial environment, alongside a collision detection algorithm and the development of an agent-based simulation platform for simulating these two methods and algorithms (Grosset et. al., 2023). The aim of the simulation in this case as well was to serve as a basis for subsequent testing in real conditions. As an improvement of the standard Dynamic Window Approach (DWA), an adaptive DWA (ADWA) method was introduced by Yang et. al. for solving the collision avoidance problem in complex environments (Yang et. al., 2022). The pro-posed method applied an adaptive neuro-fuzzy controller obtained through neural network training. MATLAB based simulations were carried out which showed the increased performance of the new approach. A fuzzy ant colony optimization (FACO) method was presented by Yen et. al. for shortest path planning and obstacle avoidance of a mobile robot (Yen et. al., 2018). Again, MATLAB simulations were used to compare the pro-posed method with other approaches, with positive results. Cong used Q-learning, a type of reinforcement learning for real time dynamic obstacle avoidance (Cong, 2023). Simulation was used for the training and testing of the algorithm, which was followed by a real life experiment. The results showed that the proposed algorithm has better performance compared to a fuzzy controller. Finally, Mohammadpour et. al. investigated the energy-efficient motion of autonomous wheeled mobile robots, partially with the aid of simulation utilizing the Gazebo software (Mohammadpour et. al., 2021). While this study did not specifically focus on collision avoidance, nevertheless this problem was also addressed during the investigation, as obstacles were introduced both in the simulation and in the experimental environment.

As it could be seen from the above presented literature, the general problem of collision avoidance represents a complex challenge which is usually approached with the combined utilization of multiple advanced methods. It was also evident that the application of simulation as one of the tools for analyzing the problem and developing/testing different solutions is a widely utilized practice. However, it could also be seen that discrete event simulations are less frequently used in this context. Yet as it was mentioned before, this represents an important field of investigation with plenty of research opportunities, as these types of simulations play a significant role in the realization of the Industry 4.0 concept. This is one of the motivating factors behind the current research, which aims to examine the collision avoidance problem between multiple AGVs and AMRs in a typical discrete event simulation environment.

3 INTRODUCTION OF THE SIMULATION MODEL

The first step was the creation of the simulation model, which would serve as a basic framework for the subsequent phases of the research. An earlier version of this model was introduced in a previous paper (Skapinyecz & Landschützer, 2022), therefore the current chapter will focus less on detailing every aspect of the concept and somewhat more on the later additions to the framework.

The current version comprises four material-flow sources, with an additional drain on the right side of the model. In the center, there is a rack system comprising six stands (these don't participate in the material flow, their sole purpose is to serve as obstacles for the AGVs to be avoided). For the current investigation, two slightly different versions of this model (A and B) were applied, which will be discussed later. The general layout with two moving AGVs can be seen in the 3D view of the simulation environment below in Figure 1.

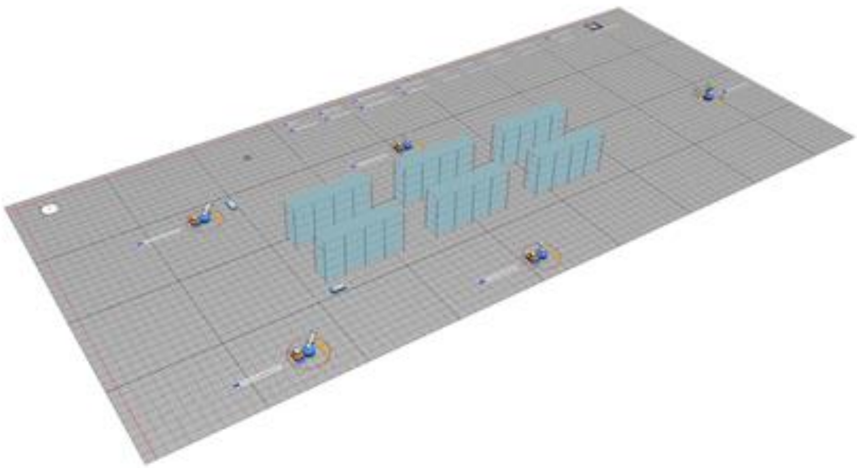

Fig. 1 The created simulation model (Model A) with two operating AGVs in the 3D view

The pathfinding of the AGVs is solved with the combined application of markers placed at different points of the layout and the utilization of Dijkstra's algorithm, which is realized within one of the developed methods for the model (this is described in more detail in the previously mentioned paper). The collision avoidance with static objects is currently solved through the appropriate definition of the pathways available between the adjacent markers, but this can also be improved upon in the future. However, as it was mentioned before, the current focus of the research is to address the collision avoidance problem between the vehicles, which represents probably the hardest challenge. The developed and applied methods and variables in the model will be briefly described in the following subchapter.

3.1 Methods and variables developed and applied in the simulation model

The custom designed methods and variables for the model, together with the standard

EperimentManager object, are represented in the simulation environment as can be seen on Figures 2 and 3.

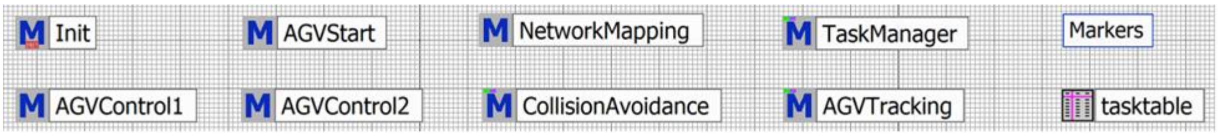

Fig. 2 Custom designed methods and variables for the simulation model

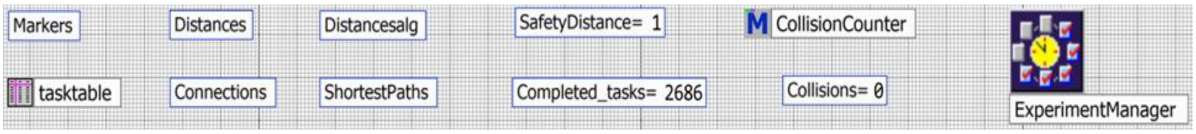

Fig. 3 Additional custom designed methods and variables for the simulation model, with the standard ExperimentManager object on the right

In the followings, a brief summary of the functions of the individual methods and variables will be provided:

Init method: In Plant Simulation, Init is always the first model-specific method which is called after the start of a simulation run. Its function, as the name implies, is to initialize the variables and call other methods at the start of the simulation (the code of the Init method is always custom designed for each model).

AGVStart: In accordance with its name, this method starts the AGVs in the model, which are launched from a so-called "AGVPool" object.

AGVControl1: Control method for the first AGV.

AGVControl2: Control method for the second AGV.

Markers: This data table contains the positions of each marker placed in the model (this data is automatically collected after the start of the simulation through the Init method).

Connections: The Connections data table defines the available pathways between each adjacent marker (effectively, it serves as a connectivity matrix).

NetworkMapping: This is the most essential method developed for the simulation model, as it comprises the implementation of Dijkstra's algorithm. It also calculates the distances between each marker based on the previous two data tables. It runs during the start of every simulation run, so by the time the AGVs start their operation, the shortest paths between each marker are calculated.

Distances: This is the data table into which the NetworkMapping method writes the distances between each adjacent marker.

ShortetsPaths: This is the data table into which the NetworkMapping method writes the calculated shortest paths.

Distancesalg: An auxiliary data table used during the execution of Dijkstra's algorithm.

TaskManager: A relatively simple task distribution method which gives the latest task to the first available AGV (a task in the model means that a unit load has to be transported from a material flow source to the drain).

tasktable: A simple data table which contains the actual state of each AGV (free or occupied).

CollisionAvoidance: The most important method from the aspect of the current study, as it contains the implemented collision avoidance algorithm (these will be described in the following chapter). It constantly monitors the distance between the vehicles and if the latter falls below the predefined safety distance, it initiates the avoidance procedure.

AGVTracking: This method keeps track of the already covered path of a given AGV in relation to its current task. This is required to restart the AGV on the remaining path after it executed a collision avoidance maneuver.

SafetyDistance: An important variable for the current study, as it contains the preset safety distance between the vehicles. If the distance falls below this threshold, the collision avoidance algorithm will be triggered.

CollisionCounter: This method counts the number of collisions between the AGVs and always writes the updated value into the Collisions variable.

Collisions: This variable contains the number of occurred collisions between the vehicles during a simulation run.

Completed tasks: This variable contains the number of completed material handling tasks by the vehicles during a simulation run.

ExperimentManager: This is a standard object provided in the simulation environment and it facilitates the automation of experiments by running a simulation model multiple times with different input parameters. This is a key feature utilized in the current study, as the analysis of the performance of the collision avoidance algorithms is implemented through this approach.

3.2 Description of the model variants applied in the current analysis

For the current study, two slightly different variants of the same model were applied, referred to simply as Model A and Model B. The following two figures (Figure 4 and 5) below show the difference between the two models, which comes mainly from the distance between the drain on the right and the central rack system. As it can be seen, in case of Model B this distance is roughly two thirds of that which can be found in model A. The reason for this difference is that the main conflict zone between the AGVs is the route leading to the drain, so it was necessary to check the effect of changing the length of this route on the performance of the collision avoidance algorithms. For this reason, the simulation runs were conducted in the case of both models with both collision avoidance algorithms. The approximate areas of the main conflict zones for both models were also highlighted with red rectangles on the figures.

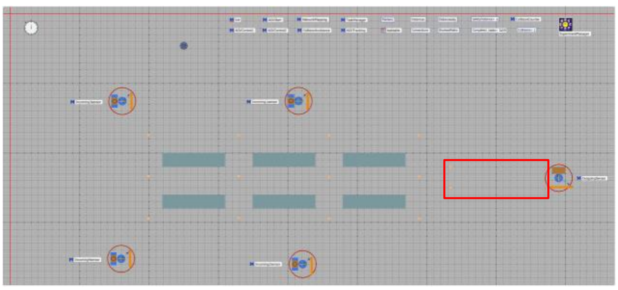

Fig. 4 Layout of Model A with the approximated area of the main conflict zone

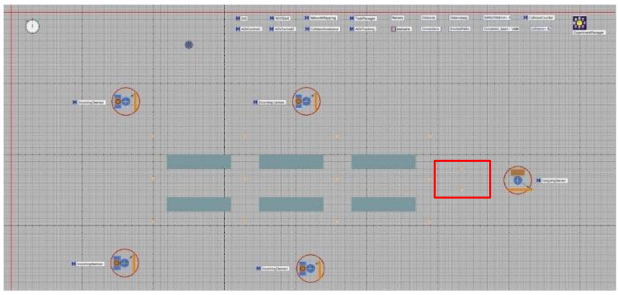

Fig. 5 Layout of Model B with the approximated area of the main conflict zone

4 DESCRIPTION OF THE COMPARED COLLISION AVOIDANCE ALGORITHMS

In the current study, two collision avoidance algorithms were compared in the context of the previously described simulation model. These algorithms themselves were also discussed separately in an additional paper (Skapinyecz, 2023), however they will also be succinctly described here for the purpose of clarity. However, before that it is important to define what qualifies as a collision or contact in the current analysis.

In case of the current experiments, a "collision boundary" for the AGVs was defined in the form of a circle around the vehicles, with a radius of 0,9m, as it is illustrated in the figure below (as the vehicle is 1.6m long and has a width of 0.8 meters, this circle almost precisely fits to the corners of the AGV):

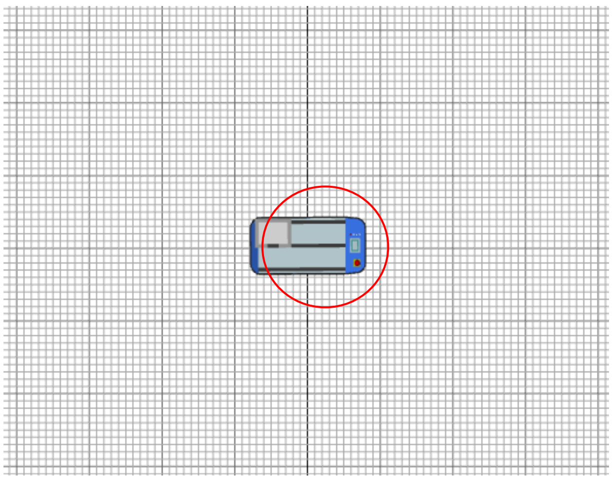

Fig. 6 Representation of the "collision zone" around an AGV in the current simulation model

As the "collision boundary" is defined in such a way that it slightly extends beyond the vehicle in case of the sides, in certain cases a collision is registered even if the two AGVs would just avoid each other with a slight margin. While this could slightly increase the number of collisions in the case of all simulation runs, it also guarantees that all possible collision events are registered, and as it is uniformly applied to every experiment, it does not alter the results of the comparisons. It must be also noted that both algorithms activate only when the vehicles are at a few meters distance away from the drain, in order to prevent artificially induced collisions in case of the second procedure in model A (right at the drain, a short rail segment prevents any collisions).

4.1 Description of the first collision avoidance algorithm

The first algorithm examined in the current analysis is relatively simple: if the distance between two AGVs falls below the defined safety distance, one of the vehicles stops and waits until the other moves away far enough so that it is no longer in the safety zone. This algorithm mainly serves as a benchmark to which the other more advanced procedure is compared to. The flowchart of this algorithm is shown below:

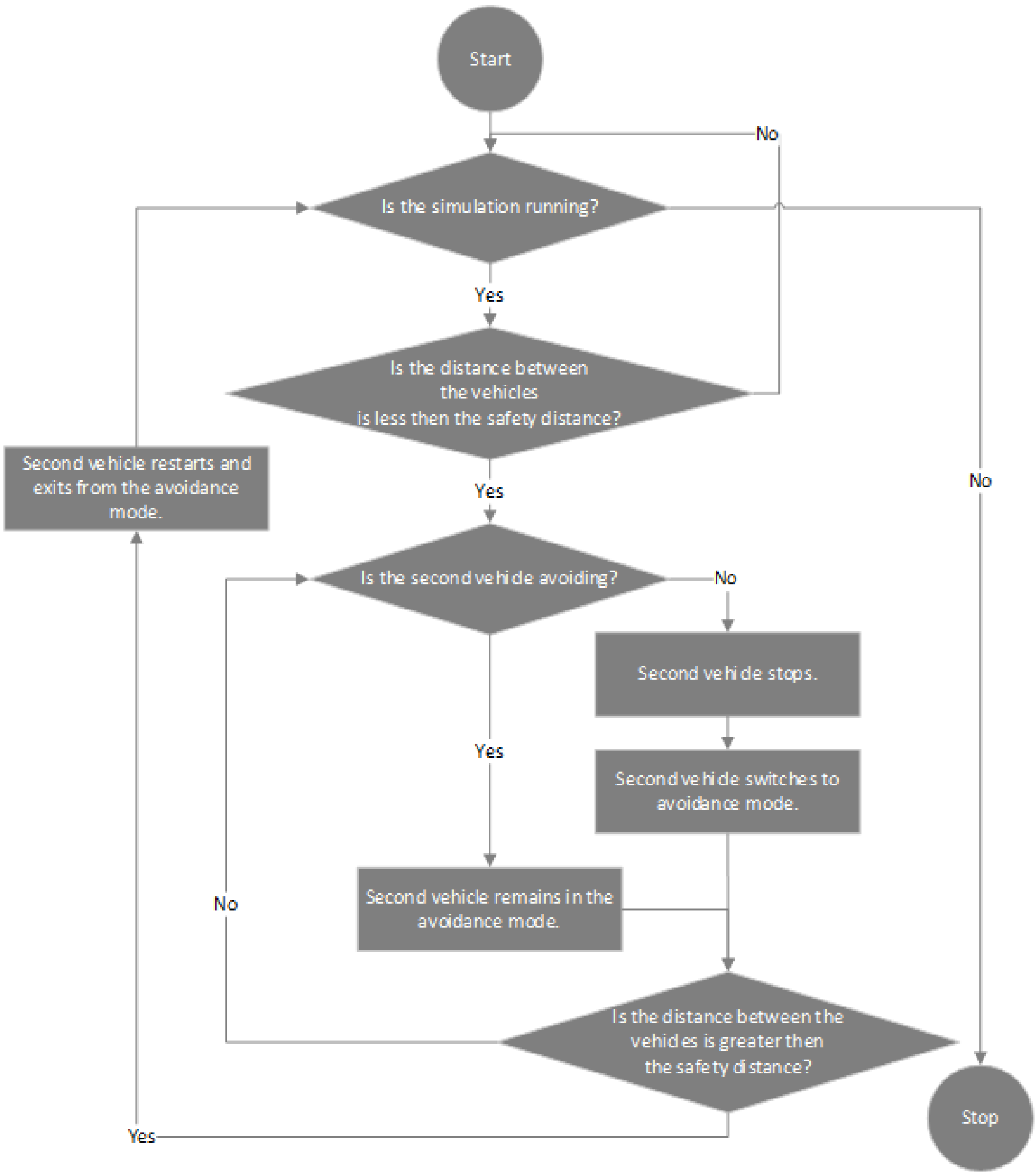

Fig. 7 Diagram of the first algorithm

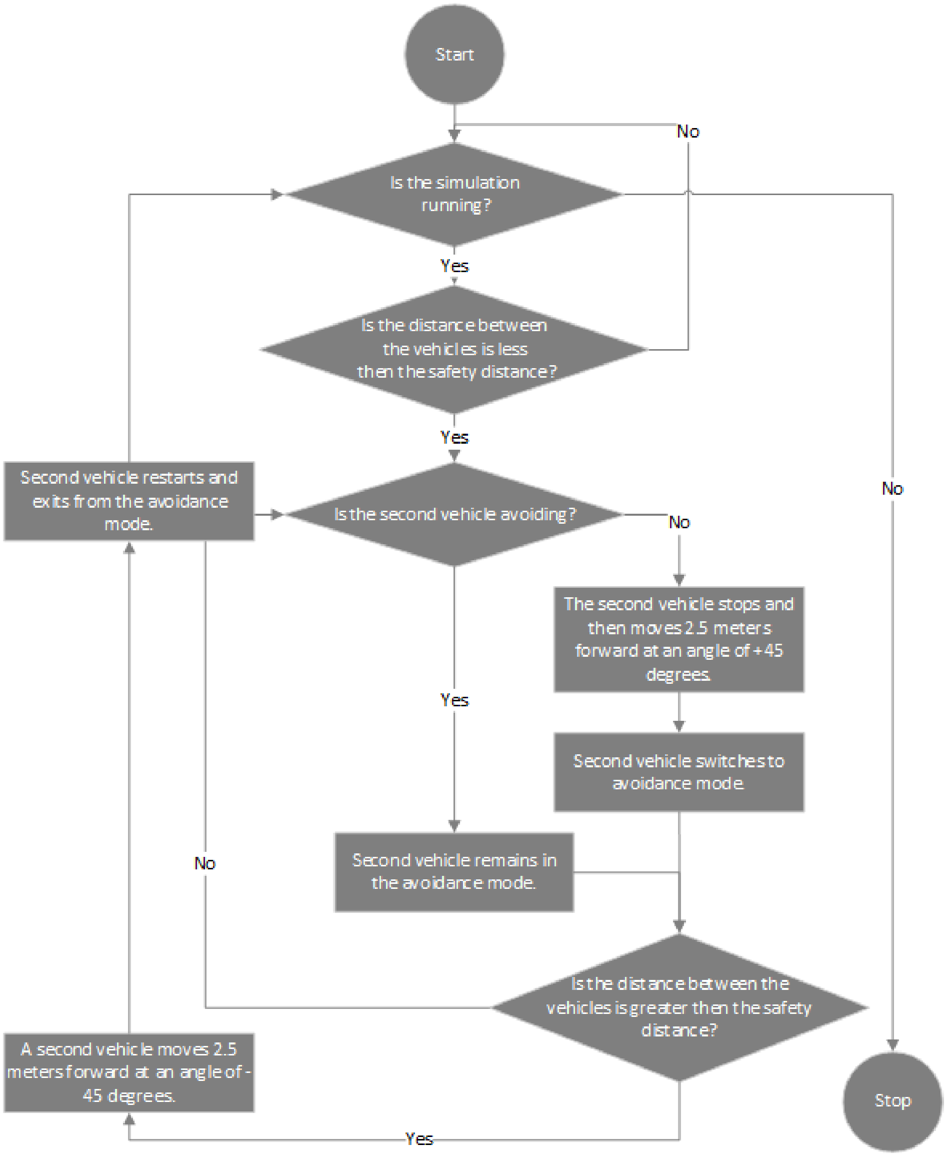

Fig. 8 Diagram of the second algorithm

4.2 Description of the second collision avoidance algorithm

The main difference between the first and second algorithms is that in case of the latter, one of the vehicles not just simply stops, but also executes an evasion maneuver as well. The flowchart of this second algorithm can be seen on the next page in Figure 8. According to the algorithm, once again the collision avoidance mode is triggered when the distance between the vehicles falls below the defined safety distance. However, in this case the vehicle switching to the avoidance mode first drives 2.5 meters forward at a 45-degree angle compared to its direction of travel and stops only then. Afterwards, when the other vehicle moved away sufficiently, it first moves again 2,5 meters forward but at a -45-degree angle, before continuing its original path.

5 SIMULATION-BASED COMPARISON OF THE COLLISION AVOIDANCE ALGORITHMS UNDER DETERMINISTIC CONDITIONS

The simulation experiments in the current study were conducted in two phases: first, the two algorithms were compared under deterministic conditions in the case of both model A and model B with different safety distance parameters, which was then followed by a second phase during which specific cases from the previous comparison were also analyzed under stochastic circumstances. The terms deterministic and stochastic in this context means that in case of the former, the material flow sources produce a new unit load after the passing of a precisely defined time period, while in case of the latter, this time period is characterized by a distribution function. It is also important to highlight that two AGVs were present in the simulations, and one of them had the responsibility to always execute the collision avoidance protocol when it was needed.

The figure below presents the operation of the model (B variant) with the two AGVs bringing their cargo to the drain:

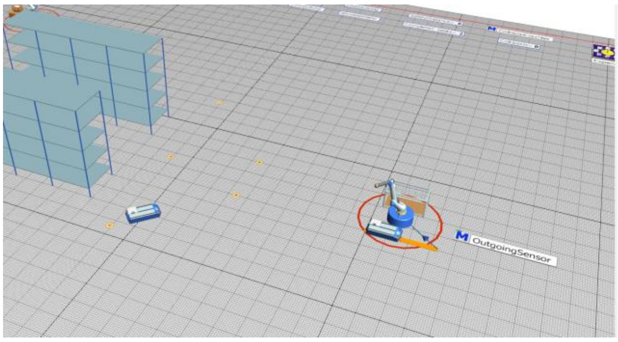

Fig. 9 One of the AGVs unloads at the drain, while the other is approaching from the direction of the rack system

As it was mentioned, in the first phase of the analysis, a discrete time period was chosen between the appearance of new unit loads at the material flow sources. This time period was uniformly set as 3 minutes and 30 second for all four sources, which guaranteed that the AGVs had no idle time and always received a new assignment once they unloaded at the drain. Here, it is also worth mentioning that the models where not optimized from a material flow perspective, which meant that over time, several unit loads were accumulated at the sources. However, this type of optimization was not the goal of the current analysis, as the main aim was to simply provide the AGVs with new tasks in a continuous manner.

Because the operational environment in the first phase was completely deterministic, that meant that there was no difference between two consecutive simulation runs with the same input parameter. For this reason, in the first part of the analysis the ExperimentManager was set to run the simulations with the same input parameter only once, as the results were always deterministic. The input parameter in all cases was the safety distance applied in the previously described algorithms. The ExperimentManager was set to run each model/algorithm combination altogether six times, with the safety distance ranging from 1 to 6 meters. The duration of a single simulation was equivalent to two days.

5.1 Simulation results for model A

The simulation results for model A with the two collision avoidance algorithms are summarized in the two tables below:

Table 1. Results for model A with the first collision avoidance algorithm

Experiment	Safety distance between the vehicles [m]	Number of collisions	Number of com-pleted tasks	Efficiency [%]
1	1	0	2686	100
2	2	0	2686	100
3	3	81	2686	96,98
4	4	142	2681	94,7
5	5	187	2649	92,94
6	6	578	2617	77,91

Table 2. Results for model A with the second collision avoidance algorithm

Experiment	Safety distance between the vehicles [m]	Number of collisions	Number of com-pleted tasks	Efficiency [%]
1	1	0	2686	100
2	2	0	2686	100
3	3	576	2686	78,56
4	4	209	2672	92,18
5	5	21	2668	99,21
6	6	158	2645	94,03

The efficiency value in the tables is inversely proportional to the ratio of collisions and completed tasks during a simulation run. As it can be seen from the two tables above, surprisingly, both the first and the second algorithm were able to prevent all collisions when the safety distance was set to either 1 or 2 meters, which translates to a 100% efficiency rating. However, the fact that the simpler first algorithm had the same performance as the more sophisticated second one raises the possibility that this is more of the result of the specific geometry and operational conditions inherent in model A, rather than the actual effectiveness of the algorithms. This suspicion is reinforced by the fact that once the safety distance started to become larger, the number of collisions also drastically increased in case of both algorithms, implying that the more frequent activation of the procedures (resulting from the larger safety distances) even degraded their effectiveness. Regarding the latter, the efficiency of both algorithms was similar in case of larger safety distances as well, though the second algorithm had a very slight advantage in this regard. However, it is clear from the results that once the safety distance became larger than 2 meters, both procedures failed to prevent a significant number of collisions.

Due to the previously presented unexpected results, it was determined that the performance of both algorithms for model A with the safety distance values of 1 and 2 meters should be further investigated. This was realized in the second phase of the analysis under stochastic operational conditions. These results will be presented in the next main chapter of the paper.

5.2 Simulation results for model B

The simulation results for model B with the compared algorithms can be seen in the following two tables:

Table 3. Results for model B with the first collision avoidance algorithm

Experiment	Safety distance between the vehicles [m]	Number of collisions	Number of com-pleted tasks	Efficiency [%]
1	1	224	2878	92,22
2	2	226	2878	92,15
3	3	228	2878	92,08
4	4	228	2878	92,08
5	5	216	2877	92,49
6	6	193	2878	93,29

Table 4. Results for model B with the second collision avoidance algorithm

Experiment	Safety distance between the vehicles [m]	Number of collisions	Number of com-pleted tasks	Efficiency [%]
1	1	224	2878	92,22
2	2	226	2878	92,15
3	3	225	2879	92,18
4	4	227	2878	92,11
5	5	76	2880	97,36
6	6	18	2879	99,37

It is apparent from the results that in case of this model variant, the second algorithm performed much better, but only in case of the 5- and 6-meter safety distances. Still, for model B, neither algorithm was able to completely prevent the occurrence of collisions regardless of the applied safety distance value. This highlights and reinforces the fact that the model geometry can have a fundamental effect on the performance of rule-based algorithms.

As the second algorithm clearly performed better for model B with the 6-meter safety distance, therefore no further investigation was needed in this case.

6 FURTHER ANALYSIS OF SPECIFIC SOLUTIONS UNDER STOCHASTIC CONDITIONS

As it could be seen from the previous section, for model A the analysis provided some unexpected results. Namely, in case of both algorithms the collisions were completely prevented with a 1- and 2-meter safety distance, but once this value exceeded 2, the performance of both procedures was significantly degraded.

To further investigate the behavior of the two algorithms in case of model A, they were also compared under stochastic operational conditions. This meant that the time interval between the appearance of two unit loads in case of all four sources was chosen to be represented with a normal distribution, with a mean value of 3 minutes and 30 seconds, and a standard deviation of 45, 60 and 75 seconds (the latter were tested in consecutive simulation experiments). Because the time interval cannot be negative, therefore a lower and an upper limit also had to be set, which were determined as 0 and 7 minutes. Besides, it is worth noting that the carrying capacity of the AGVs also had to be significantly expanded (by a factor of 10) in the model, in order for them to be able to cope with the fluctuation of the output frequency at the material flow sources.

Because of the previous changes, the results themselves also became stochastic, which meant that this time the ExperimentManager had to be configured in such a way that multiple consecutive simulations (called attempts) could be performed with the same input parameter and could produce the results as a min-max interval. As a standard setup in the ExperimentManager, the number of attempts (consecutive simulation runs with the same input) were set to 5. The following figure shows the graphical representation of the results for the number of collisions in one of the experiments as a min-max interval (the figure is generated in the report of the ExperimentManager):

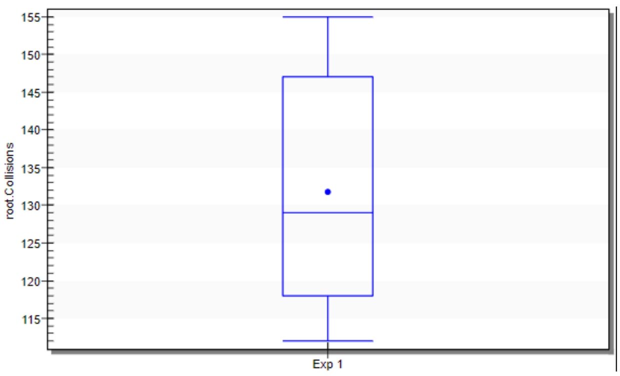

Fig. 10 Representation of the min-max interval for the collisions in the report of the ExperientManager with a 75 sec standard deviation at the sources (Model A)

The aim of using the previously described stochastic operational conditions is that by introducing statistical variability to the generation of the unit loads, the workflows of the two AGVs will also become stochastic, which will largely negate the effect of the specific model geometry on the experimental results. In the following two tables, the statistical results for the number of collisions in the case of both algorithms with a 1m safety distance are presented:

Table 5. First algorithm with a 1m safety distance

Standard deviation of the time interval at the sources [sec]	Mean number of collisions	Deviation	Minimum number of collisions	Maximum number of collisions
45	116.4	11.7175	102	132
60	116,6	9.44986	109	133
75	129	9.84885	120	145

Table 6. Second algorithm with a 1m safety distance

Standard deviation of the time interval at the sources [sec]	Mean number of collisions	Deviation	Minimum number of collisions	Maximum number of collisions
45	121	14.4395	96	130
60	129.6	8.79204	123	145
75	131.8	16.2080	112	155

The results indicate that in case of a 1m meter safety distance, the first algorithm performs slightly better in the stochastic case, even though both procedures fail to prevent a significant number of collisions. The latter reinforces the fundamental observation that while it is possible to find optimal rule-based solutions for specific situations, it is usually hard to extend these for a more generalized problem. Furthermore, the results also seem to reinforce the assumption that the previously flawless performance of the compared algorithms in case of a 1m safety distance was mostly the combined result of the specific layout geometry and the preset unit load generation frequency, as once stochasticity was introduced into the workflow, the effectiveness of the collision avoidance procedures was significantly decreased.

For the sake of completeness, the actual efficiency values for both algorithms are also highlighted in the next table (these are determined by dividing the mean numbers of the collisions with the respective mean numbers of the completed tasks):

Table 7. Efficiency of the first and second algorithms with a 1m safety distance under stochastics conditions

Standard deviation of the time interval at the sources [sec]	Efficiency of the first algorithm	Efficiency of the second algorithm
45	95,268	95,083
60	95,256	94,727
75	94,746	94,636

The analysis was also conducted with a 2-meter safety distance. The results are presented in the following two tables:

Table 8. First algorithm with a 2m safety distance

Standard deviation of the time interval at the sources [sec]	Mean number of collisions	Deviation	Minimum number of collisions	Maximum number of collisions
45	159,4	19.0866	132	181
60	158,4	10.6677	144	169
75	163	8.57321	157	178

Table 9. Second algorithm with a 2m safety distance

Standard deviation of the time interval at the sources [sec]	Mean number of collisions	Deviation	Minimum number of collisions	Maximum number of collisions
45	135.4	8.64869	128	150
60	129.2	8.34865	120	142
75	144.8	5.35724	138	150

Interestingly, the results indicate that with a 2-meter safety distance, the second algorithm clearly performed better, though again, none of the two procedures were able to achieve a 100% percent efficiency. The following table represents the actual efficiency values for the two algorithms:

Table 10. Efficiency of the first and second algorithms with a 2m safety distance under stochastics conditions

Standard deviation of the time interval at the sources [sec]	Efficiency of the first algorithm	Efficiency of the second algorithm
45	93,509	94,496
60	93,544	94,746
75	93,353	94,106

Overall, based on both the efficiency values and the number of collisions, it must be highlighted that the first algorithm with a 1-meter safety distance produced the best results in the stochastics case. As it was mentioned however, none of the procedures was able to achieve a 100% efficiency in this second phase of the analysis, which again underlines the importance of the effects of the model geometry and other operational circumstances when using rule-based approaches.

7 DISCUSSION AND THE POSSIBLE CONTINUATION OF THE RESEARCH

While some of the results of the simulation-based comparisons support the assumption that certain rule-based algorithms can be optimized to solve the collision avoidance problem under very specific conditions, in most of the cases the analyzed procedures were unable to reach a 100% reliability (zero collisions during a simulation run). This was especially apparent during the second phase of the analysis, in which the algorithms were examined under stochastic conditions and were unable to prevent a larger number of collisions, even though previously they were seemingly able to achieve a 100% efficiency in the deterministic case under specific conditions (though this was probably resulted mainly from the specific layout of model A, and the relationship between the geometrical arrangement of the operational area and the workflow of the AGVs). These results support the conclusion that the general solution of the collision avoidance problem requires the application of more sophisticated algorithms, or perhaps a combination of procedures. Of course, it must be noted that the two examined algorithms themselves could be further upgraded (especially in the case of the second one) to be able to react more flexibly to the different collision scenarios, while it would also be possible to use a combination of rule-based procedures depending on a more diverse set of parameters besides the distance. However, this latter approach would very likely still have its limits, so the use of an entirely different concept would probably promise more success.

As it was mentioned in the introduction, one of the goals of the current research was to serve as a starting point for the further development of the proposed modelling approach. An important part of the latter would be the further development and extension of the presented simulation model itself. This could include the scaling up of the model so it would incorporate larger and more sophisticated material flow processes, which would result in a more complex environment for the AGVs to navigate in with more varied material handling tasks. This could be relatively easily done with the standard material flow objects provided by the simulation environment and the result would be a large and generally applicable test model which would facilitate the analysis of any reasonable traffic environment. In on itself, the inclusion of a larger number of freely moving AGVs in the model could also be easily done and this would be necessary as well in the future, as most systems usually contain more than two vehicles. However, the harder part of this would be to guarantee the proper separation between the vehicles, which of course leads to the core problem that is in the focus of the current research. Finally, certain parts of the basic model could also be elaborated further. For example, the collision boundaries around the vehicles could be refined to match more closely the precise outer geometry of the vehicles, while the distance measurement from stationary objects should also be included in a more advanced collision avoidance procedure, even if this part of the problem is simpler and can be at least partially solved with the proper placement of the markers, as it is done in the current version of the simulation model. Besides, a more refined solution could also be developed to handle the collision avoidance problem at the close vicinity of the drain, as while the few meter delay radius around it prevents artificially induced collisions mainly in case of the second algorithm applied for model A, for some of the suboptimal results in other cases the number of collisions could be perhaps somewhat decreased by activating the collision avoidance procedure closer to the drain, though this didn’t have any effect on any of the best results in the first phase of the analysis.

In terms of the actual collision avoidance algorithms, a natural continuation of the research would be the utilization of machine learning techniques to solve the more generalized problem. At first glance, this would require a completely different approach, however, the main advantage of the described framework is that it could be relatively easily modified to generate training data for an artificial neural network (ANN) to be developed later. For this purpose, the monitoring of the collision events (both actual collisions and the successfully avoided ones) in the model has to be expanded in such a way that not only the number of collisions is registered, but also a number of other parameters in each collision event as well. These parameters at minimum should include the direction of travel of the AGVs at the start of the collision avoidance procedure, the angle at which an AGV diverges from its original distance of travel (which in this case should be a variable input parameter for the ExperimentManager) and of course the outcome of the event (whether the collision was prevented or not). Of course, at one point the distance measurement from other nearby objects should also be introduced as well. There could be several other parameters that could be also taken into account, but these would be the most fundamental for the planned application.

With the introduction of the previously mentioned parameters, the model then can be modified to generate training data for an ANN, where the main output parameter would be the success or failure of preventing a collision with a given set of the previously described inputs. In this context, the rule-based approach would only be used in the generation of this training data on which an ANN can be trained afterwards. One further advantage of the utilized simulation environment is that the ExperimentManager object would be especially suitable for automating the previously described training data generation process. With the utilization of this data, the aim of the subsequent development would be to design an ANN that is complex enough to handle the generalized collision avoidance problem for multiple vehicles, but at the same time is compact enough to be trained and executed on desk-top machines, in the latter case in combination with running the simulation. This network could be implemented in the simulation environment itself, or as a separate code interfaced with the simulation software. While the latter option is also theoretically possible as the software provides multiple interfacing options, for reasons of practicality the first option would probably be preferable. It is also worth mentioning here that the Plant Simulation software also provides a premade customizable ANN object as well, though utilizing this solution would only be one of the multiple available options.

The overall goal would be to develop an algorithm which is significantly less resource-intensive then the full-scale emulation of the software and hardware of advanced AGVs and AMRs, but can still adequately solve the general collision avoidance problem within the discrete event simulation paradigm simultaneously for a larger number of vehicles and in a dynamic industrial environment (of course, this algorithm could not be used in a real-life industrial environment and could not replace the algorithms designed for real-world implementation). It is also worth mentioning here that various high-level collision prevention strategies could also be combined later with the planned machine learning based solution, but the focus of the current research is to avoid unexpected collision situations emerging within a short distance from the vehicle, as this is an important task when a larger number of fully autonomous vehicles are present in a dynamic operational environment.

8 CONCLUDING REMARKS

The paper demonstrated that under very specific conditions, it is possible to apply certain rule-based approaches for the collision avoidance of AGVs and AMRs, at least in a process simulation environment. However, the results also clearly showed that the slightest change in the operational environment can significantly decrease the efficiency of a rule-based algorithm optimized for a specific set of conditions, while their performance also largely depends on the actual layout and other characteristics of the operational environment. Therefore, to solve the generalized collision avoidance problem in the simulation environment, a more advanced and complex algorithm is needed.

The future aim of the research would be to use the presented discrete even simulation-based test method as a starting point to develop a more advanced algorithm. The latter would likely be based on machine learning, possibly in the form of an ANN. As it was described, the presented approach could not only be used for the testing of collision avoidance algorithms designed for a discrete event simulation environment, but with the right modifications, also to generate the training data required by a neural network model. Besides, the simulation model itself could also be further expanded and elaborated, thereby enabling the analysis of any traffic environment relevant for an Industry 4.0 based material flow system, with the inclusion of a larger number of freely moving AGVs and AMRs.

9 REFERENCES

Fragapane, G., De Koster, R., Sgarbossa, F., & Strandhagen, J. O. (2021). *Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda.* European Journal of Operational Research, 294(2), 405-426

Turhanlar, E. E., Ekren, B. Y., & Lerher, T. (2022). *Autonomous mobile robot travel under deadlock and collision prevention algorithms by agent-based modelling in warehouses.* International Journal of Logistics Research and Applications, DOI: 10.1080/13675567.2022.2138290, 1-20

Praserttaweelap, R., Kaitwanidvilai, S., & Aoyama, H. (2019). *Safety path planning with obstacle avoidance using particle swarm optimization for AGV in manufacturing layout.* Int. J. Innovative Computing, Inf. Control, 15(1), 351-368

Thai, N. H., & Trinh, L. T. K. (2021). *Roadmap, routing and obstacle avoidance of AGV robot in the static environment of the flexible manufacturing system with matrix devices layout.* VNUHCM Journal of Science and Technology Development, 24(3), 2091-2099

Zając, J., & Małopolski, W. (2021). *Structural on-line control policy for collision and deadlock resolution in multi-AGV systems.* Journal of Manufacturing Systems, 60, 80-92

Stączek, P., Pizoń, J., Danilczuk, W., & Gola, A.

(2021). *A digital twin approach for the improvement of an autonomous mobile robots (AMR's) operating environment—A case study.* Sensors, 21(23), 7830

Fan, T., Long, P., Liu, W., & Pan, J. (2020). *Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios.* The International Journal of Robotics Research, 39(7), 856-892

Kozjek, D., Malus, A., & Vrabič, R. (2021). *Reinforcement-learning-based route generation for heavy-traffic autonomous mobile robot systems.* Sensors, 21(14), 4809

Feng, S., Sebastian, B., & Ben-Tzvi, P. (2021). *A collision avoidance method based on deep reinforcement learning.* Robotics, 10(2), 73

Grosset, J., Ndao, A., Fougeres, A. J., Djoko-Kouam, M., Couturier, C., & Bonnin, J. M. (2023). *A cooperative approach to avoiding obstacles and collisions between autonomous industrial vehicles in a simulation platform.* Integrated Computer-Aided Engineering, (Preprint), 1-22

Yang, D., Su, C., Wu, H., Xu, X., & Zhao, X. (2022). *Construction of novel self-adaptive dynamic window approach combined with fuzzy neural network in complex dynamic environments.* IEEE Access, 10, 104375-104383

Yen, C. T., & Cheng, M. F. (2018). *A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance.* Microsystem Technologies, 24, 125-135

Cong, V. D. (2023). *Path Following and Avoiding Obstacle for Mobile Robot Under Dynamic Environments Using Reinforcement Learning.* Journal of Robotics and Control (JRC), 4(2), 157-164

Mohammadpour, M., Zeghmi, L., Kelouwani, S., *Gaudreau, M. A., Amamou, A., & Graba, M.* (2021). *An investigation into the energy-efficient motion of autonomous wheeled mobile robots.* Energies, 14(12), 3517

Skapinyecz, R., & Landschützer, C. (2022). *Application of freely moving AGVs in a discrete event simulation environment: a first model.* Advanced Logistic Systems-Theory and Practice, 16(2), 5-23

Skapinyecz, R. (2023). *Examples for the application of rule-based algorithms in the simulation modeling of the physical separation of automated guided vehicles.* Advanced Logistic Systems-Theory and Practice, 17(2), 5-13