Advanced Logistic Systems – Theory and Practice, Vol. 16, No. 2 (2022), pp. 5-23.
https://doi.org/10.32971/als.2022.009

APPLICATION OF FREELY MOVING AGVS IN A DISCRETE EVENT SIMULATION ENVIRONMENT: A FIRST MODEL

RÓBERT SKAPINYECZ[1] - CHRISTIAN LANDSCHÜTZER[2]

Abstract: The use of automated guided vehicles (AGVs) in manufacturing systems has become particularly widespread in the latter period, thanks to the intensification of automation and to the adoption of the Industry 4.0 principle. Modern AGVs can also have greater autonomy then before, thanks to the advancement of the related technologies which further improve their applicability in numerous complex settings. In the Industry 4.0 principle, the use of predictive simulations in relation to manufacturing logistics systems also plays an important role, which applies to the modelling of the operation of AGVs as well. In light of these developments, the current paper introduces an initial model with the aim to demonstrate a possible approach for modelling the operation of freely moving AGVs in a discrete event simulation environment. The model is made with Plant Simulation, a widely applied discrete event simulation software. The presented approach opens the door for multiple future research avenues and could also serve as a basis for a further developed general model with wider utilization possibilities.

Keywords: discrete event simulation, automated guided vehicles, freely moving AGVs, Industry 4.0

1. INTRODUCTION

In the latter period, the application of the Industry 4.0 principle became a cornerstone of the development and operation of state-of-the-art manufacturing systems. The concept includes the use of advanced automation, digitalization, extensive sensor systems, big data analysis, artificial intelligence and so-called digital twins, among other tools, applied together in an integrated methodology. Due to the recent technological advances in the mentioned fields, their joint application provides such opportunities for efficiency improvement which were simply unattainable before. This is especially true in the cases of manufacturing and material handling, where automation traditionally has had a significant role.

One of the typical hallmarks of a modern manufacturing system is the use of automated guided vehicles (AGVs). This type of material handling machine has been in use for many decades. So, from the late 1960ies on material handling industry with intralogistics knows and uses guided vehicles and forklifts, long before autonomous driving and AGV became modern. However, the level of autonomy of modern AGVs can be significantly higher than before, thanks to advances in the related navigation and control technologies, and of course to the exponential advancement of digital technology. For example, while the first AGVs where strictly bound to follow a prepositioned wire, modern vehicles can be quasi-freely moving in the sense that they position themselves through laser guidance or through novel indoor localization techniques (for example using different Real-time locating system (RTLS) solutions), while following arbitrarily defined paths which could be changed according to the material handling tasks and to the environmental circumstances. While these paths are usually still physically marked with standard solutions such as UV markers,

[1]associate professor, University of Miskolc, Institute of Logistics, Miskolc, Hungary
robert.skapinyecz@uni-miskolc.hu
[2]associate professor, Graz University of Technology, Institute of Logistics Engineering, Graz, Austria
landschuetzer@tugraz.at

QR codes, etc., these techniques still offer much greater flexibility, while the application of free navigation without the use of any physical markers is also on the rise. All this naturally comes together with an increased ability of autonomous collision avoidance, the application of more sophisticated scheduling and traffic management strategies, and often with the application of artificial intelligence (especially in the case of free navigation). As a result, it can be stated that today's AGVs are more-and-more becoming Autonomous Mobile Robots (AMRs), which can also be seen in the trend, that the latter definition is becoming increasingly applied for modern AGVs.

As the use of AGVs is becoming more-and-more widespread in many different industries, the need to model their operation in intralogistics and manufacturing simulations is also growing. This is especially true because of the fact that these types of simulations are also becoming a standard tool in the Industry 4.0 principle. Again, it can be stated that manufacturing simulations and especially discrete event-based simulations have been in use for decades, however the recent advancements in digitalization greatly increased their significance even more. Today, in the concept of Industry 4.0, these tools are often used as so called "digital twins", which highlights the fact that the exponential increase in the amount of the available data from manufacturing systems makes it possible to almost model the real system perfectly in a virtual simulation. Furthermore, as for any digital twin, the use of this simulation model is not only of interest for a pre-production simulation in the designing phase, but the more for continuous monitoring and control and for optimization and daily systems customization of the logistics environment itself.

However, with the widespread adoption of highly autonomous AGVs, the modelling of their operation in manufacturing simulations could also become a harder problem, depending on the requirements and circumstances. This serves as the starting point for the current research. In the followings, an initial discrete event-based simulation model is going to be presented with the aim to demonstrate a possible approach for the modelling of freely moving AGVs in a modern manufacturing logistics environment. The model is created with the use of Siemens Tecnomatix Plant Simulation (educational version 16.0), a widely utilized discrete event simulation software.

The presented model is only the first iteration in the current research and as such, it has limitations which make further significant research and improvements necessary before the introduced approach could be utilized in a broader sense. On the other hand, it could also serve as a basis for multiple future research directions. Another advantage of the approach is that the model structure can be customized according to the given system, while the utilized simulation environment itself is particularly suitable for the simulation of manufacturing systems and it is widely applied in the industry in many roles. Therefore, with further development, the introduced approach has the potential to be utilized with relative flexibility for the simulation and analysis of a wider range of AGV based material handling systems.

2. REVIEW OF THE RELATED LITERATURE

The literature related to the modelling of the operation of AGVs is naturally extensive. Therefore, the current representative literature review primarily includes such publications which to a great extent (but not exclusively) deal with the simulation of the high-level operation of AGVs in a virtual environment. Besides of that, it also covers multiple publications focused on the control of AGV systems. The literature review is mostly done through the use of Google Scholar, primarily with the keywords “AGV simulation”, “discrete event based simulation AGV routing” and “AGV control strategy”.

Ma, Zhou and Stephen developed a discrete event simulation model of battery-powered AGV systems in automated container terminals (Ma, Zhou, Stephen, 2021) [1]. Based on the model, they analyzed multiple operational scenarios in case of two different layout designs and two different recharging policies.

López, Zalama and Gómez-García-Bermejo developed a comprehensive simulation and control framework for AGV based transport systems (López, Zalama, Gómez-García-Bermejo, 2022) [2]. Characteristic for their work is that it builds upon the modular control framework approach. Also, their developed framework is used both for the simulation and for the implementation of the different policies and algorithms in the real global control system.

Luo and Zhao developed a discrete event simulation model for AGV based transport in robotic mobile fulfillment systems (RMFS), in accordance with the modular simulation theory (Luo, Zhao, 2022) [3]. They also presented multiple simulation results for large scale RMFS experiments.

Viharos and Németh developed a general discrete event simulation model for the modeling and scheduling of AGV based robotic assembly systems (Viharos, Németh, 2018) [4]. They also used the Plant Simulation software for their work. The main goal of their research was to support the layout planning and evaluation of reconfigurable assembly systems.

Bhosekar, Ekşioğlu, Işık and Allen developed three discrete event simulation models for the modeling of coordinated inventory management and material handling in hospitals using the ARENA software (Bhosekar, Ekşioğlu, Işık, Allen, 2021) [5]. Their models utilized AGVs for material handling and used real-world data for the simulation of their operation.

Greasley explored the possibilities for combining discrete event simulation with machine learning (Greasley, 2020) [6]. The example model presented in his paper utilized the combined approach for the navigation of a mobile robot in a factory.

Fu, Zhang, Ding, Qin and Jiang developed a method which combines discrete event simulation, sensitivity analysis, fractional factorial design and response surface methodology for determining the vehicle requirements of an AGV system (Fu, Zhang, Ding, Qin, Jiang, 2021) [7]. They also used Plant Simulation in the presented case study.

In their paper, Klaas, Laroque, Dangelmaier and Fischer presented a simulation aided, knowledge-based routing method for AGVs in a distribution warehouse (Klaas, Laroque, Dangelmaier, Fischer 2011) [8]. For the building of the knowledge base, they also applied machine learning together with discrete event simulation.

Kesen and Baykoç examined the adaptation of AGVs into just-in-time (JIT) systems in a job-shop environment (Kesen, Baykoç, 2007) [9]. For this purpose, they developed a simulation model using the ARENA software.

Um, Cheon, and Lee used simulation for the design and analysis of a Flexible Manufacturing System using AGVs (Um, Cheon, Lee, 2009) [10]. In their approach, they used the systematic analysis methods that combine Multi-Objective Non-Linear Programming and Evolution Strategy.

Digani, Sabattini, Secchi and Fantuzzi proposed a hierarchical traffic control algorithm for AGVs implementing a two layer architecture (Digani, Sabattini, Secchi, Fantuzzi, 2014)

[11]. They applied simulation for the validation of the proposed strategy using a real plant as a basis.

In their paper, Draganjac, Miklić, Kovačić, Vasiljević and Bogdan presented an algorithm for decentralized control of multiple AGVs in industrial and warehousing environments (Draganjac, Miklić, Kovačić, Vasiljević, Bogdan, 2016) [12]. They used simulation together with experiments for the validation of the algorithm's performance.

Azimi, Haleh and Alidoost examined the pick-up-dispatching problem together with delivery-dispatching problem of a multiple-load automated guided vehicle (AGV) system (Azimi, Haleh, Alidoost, 2010) [13]. Their aim was to select the best control rule for the problem. For this purpose, they developed a new framework based on Multiple Attribute Decision Making and conducted simulations experiments to find the results.

Neradilova and Fedorko developed a simulation to model the supply of workplaces by AGVs in a digital factory (Neradilova, Fedorko, 2017) [14]. They also applied the Plant Simulation software to create the model.

Hsueh proposed a bi-directional load-exchangeable AGV system as a solution for a number of commonly occuring problems (Hsueh, 2010) [15]. Simulation experiments were carried out to evaluate the performance of the proposed system.

Fedorko, Molnar, Vasil and Hanzl used discrete event simulation to model the handling of ocean containers using an AGV system (Fedorko, Molnar, Vasil, Hanzl, 2018) [16]. They also applied the Plant Simulation software for this purpose.

In their paper, Pedan, Gregor and Plinta discussed the design and implementation of an AGV system in a healthcare facility (Pedan, Gregor, Plinta, 2017) [17]. They used the SIMIO simulation software for the verification of the designed system, which was also followed by an economic evaluation.

Fanti, Mangini, Pedroncelli and Ukovich proposed a decentralized control strategy of AGV systems, in which the vehicles select their route through a consensus algorithm, while zone control is also applied to avoid deadlocks and collisions (Fanti, Mangini, Pedroncelli, Ukovich, 2018) [18]. They used simulation to demonstrate the applicability of the introduced approach.

Vivaldini, Galdames, Pasqual, Sobral, Araújo, Becker and Caurin developed an automatic routing system for AGVs in intelligent warehouses (Vivaldini, Galdames, Pasqual, Sobral, Araújo, Becker, Caurin, 2010) [19]. They applied a dynamic programming approach for the design of the system. Simulation was applied for the testing of the developed algorithm.

Möhring, Köhler, Gawrilow and Stenzel proposed an AGV routing algorithm for automated logistic systems (Möhring, Köhler, Gawrilow, Stenzel 2005) [20]. The algorithm works in real-time to calculate conflict-free shortest paths with time-windows, where the latter could be readjusted if necessary. They tested the algorithm on a graph with about 30000 arcs.

In their paper, Zhang, Guo, Chen and Yuan proposed a collision-free routing method for AGVs based on collision classification (Zhang, Guo, Chen, Yuan 2018) [21]. They identified four types of collisions and developed a solution for each case. The operation of the algorithm was demonstrated through a case study developed with Visual C++.

For the real-time path planning of AGV systems, Fransen, Van Eekelen, Pogromsky, Boon and Adan presented a dynamic approach using a graph representation of the path layout, where the vertexes are updated over time via exponential smoothing (Fransen, Van

Eekelen, Pogromsky, Boon, Adan, 2020) [22]. The evaluation of the method was done using discrete event simulation.

Liu, Jula, Vukadinovic, and Ioannou developed and used simulation models to demonstrate the impact of automation and terminal layout on terminal performance (Liu, Jula, Vukadinovic, Ioannou, 2004) [23]. Multi attribute decision making was also used to assess the performance of the two considered yard configurations and to determine the number of AGVs in both cases.

In their paper, Ho and Chien proposed a control methodology for multiple load AGVs (Ho, Chien, 2006) [24]. They identified the main problems a multi-load AGV will face in its operation and proposed a set of rules for the solution of these problems. The proposed methodology was tested using computer simulation.

Micieta, Edl, Krajcovic, Dulina, Bubenik, Durica, and Binasova proposed a proof-of-concept solution based on the delegate multi-agent systems approach for coordination and control of one-directional AGV systems (Micieta, Edl, Krajcovic, Dulina, Bubenik, Durica, Binasova, 2018) [25]. The concept was verified by simulation in a 3D environment.

Moorthy, Hock-Guan, Wing-Cheong, and Chung-Piaw proposed a deadlock prediction algorithm based on the use of simulation for a zone-controlled AGV system (Moorthy, Hock-Guan, Wing-Cheong, Chung-Piaw, 2003) [26]. The described solution makes it possible to both detect and avoid deadlock situations. The AutoMod simulation software was used both for the implementation of the algorithm and for the modelling of the AGV operations.

Cservenák designed a complex simulation model for aiding the motion control of an AGV using the Scilab software system (Cservenák, 2021) [27]. The model contains 6 main blocks (Path planning, Trajectory planning, Velocity-voltage converting, Dynamical model of DC motor, Path simulation, Data processing) and each one was developed in detail with the use of the simulation environment.

As it could be seen from the representative literature review above, in the last decades simulation tools have been frequently used for the modelling and analysis of AGV systems with very significant results. Discrete event simulations have also been frequently applied and naturally the Plant Simulation software itself has often been used in this field of application. The combination of discrete event simulation with the modelling of freely or quasi-freely moving AGVs is on the other hand a relatively less frequently taken approach. A characteristic of the current research is that it follows the latter direction, while it aims to potentially serve as a starting point in the development of a general modelling procedure for the simulation of freely moving AGVs in complex manufacturing logistics environments. Thereby, it can contribute to the existing literature on the topic by providing the basis for a relatively flexibly applicable modelling approach for complex and highly autonomous AGV systems which is integrated into the widely applied discrete event simulation paradigm. As the latter is one of the most important tools used for the creation of digital twins of modern logistics systems, the outlined research direction can therefore provide multiple practical and theoretical advantages for the future.

3. Introduction of the approach used and the initial model

In Plant Simulation, there are multiple ways to simulate AGV based transport systems. One standard and widely applied method is to use tracks or two-lane tracks, on which the “transporters” (the transporter object is the one that is used to represent an AGV) can operate according to the related control algorithms. Another possible approach is to use “Marker” objects, which serve as waypoints for freely moving transporters (AGVs) in the system. It is clear that the latter method is the one which suits best the goals of the current research. The developed initial model essentially builds upon this latter capability of the software for the purpose of being able to define a large number of flexibly selectable approximate paths for the AGVs. Between the Markers, the AGVs move freely according to their in-built navigation procedure which always finds the shortest direct path between two waypoints (from now on, Markers and waypoint are used synonymously in the paper), as well as along a set of waypoints, while taking the necessary turning radiuses into consideration.

For finding the optimal routes between the waypoints in accordance with the existing environmental constraints, the current approach utilizes a graph, in which the Marker objects represent the vertexes and the available direct pathways between adjacent Markers are essentially the edges. As it was mentioned in the previous paragraph, the AGVs automatically find the direct paths themselves between two Markers or along a specified set of Markers according to their in-built procedure, so the two main considerations that has to be taken into account during the definition of the graph is that the edges cannot cross any obstacle in the layout and also enough room has to be left free for maneuvering. Essentially, the utilized graph defines the lists of possible adjacent Markers that can be directly visited from each vertex (each Marker) without colliding with any static obstacle, also taking into account the necessary turning radiuses. To find the shortest route between the vertices along the graph itself, the current approach utilizes Dijkstra’s algorithm, which is one of the more frequently used pathfinding algorithms, both in general and in the case of AGV simulations (it was used in several of the cited literature as well, for example in [1], [2], [3], [20] and [21]). Of course, when strictly considering the previously outlined problem, the solution for it can be achieved in a much simpler way with the utilization of fixed tracks for the AGVs, but the focus of the current research is on the modelling of freely moving AGVs.

The initial model is mainly focused on the pathfinding problem, so the applied solution for it is going to be presented in greater detail in the current chapter. Of course, it is only one part of the problem set that has to be solved for the complete modelling of freely moving AGVs. Other important questions include the problems of collision avoidance, task management, and several other important considerations. In the current model, while the problems of collision avoidance and task management are addressed on a basic level, finally they still must be fully solved in the following stages of the research, along with other important modelling tasks. As such, all the mentioned problems beyond pathfinding will be addressed in the following main chapter.

3.1. Definition of the graph structure with the possible waypoints for the AGVs

As it was previously mentioned, in Plant Simulation a predefined Marker object can be used to set waypoints for a freely moving AGV (see Figure 1. on the next page). The AGV always automatically finds the shortest path between two Markers, by the above-mentioned algorithm. However, in a manufacturing environment, there are a lot of environmental constraints which could limit the available space for manoeuvring. Therefore, in the proposed model the Markers are mainly used to define the crossings between the transport corridors that can be used by the AGVs and for the designation of other important operational areas. Of course, these could represent real physical markers (for example UV markers) in the modelled system, but they can also represent completely virtual waypoints for approximating the path that is going to be taken by a freely navigating AGV. An advantage of using this approach is that it provides a high level of freedom in defining transport corridors and routes with arbitrary geometry.

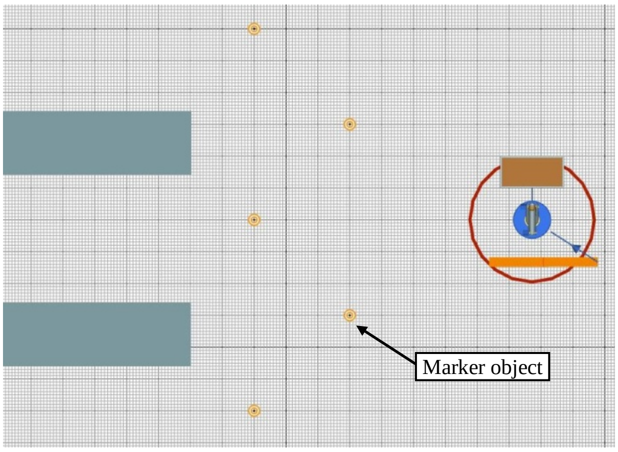

Figure 1. Picture with Marker objects in the middle in the 3D mode of the simulation environment from the top view

After the Markers are placed, in the proposed approach a graph structure is defined incorporating these waypoints. This graph essentially defines the possible routes the AGVs can take in-between the Markers. To construct the graph of available routes between the adjacent Markers, in the developed model the 3D coordinates of the waypoints are first collected by the initializing method and then written into a data table. Here, it is also worth mentioning that Plants Simulation has both a 2D and a 3D view, which can be used both for the development and for the running of a simulation (in case of the current research, the development took place in the 3D view). The previously mentioned graph structure is defined manually in a separate data table, in which "1" represents a connection between two Markers, while an empty cell means that there is no connection between the two waypoints. In other words, this latter data table serves as the connectivity matrix for the graph. Based on the coordinates of the vertices and on the structure of the connectivity matrix, the distances between each connected node (in other words, the length of the edges) are then calculated in a separate custom created method called "NetworkMapping", which also contains the pathfinding algorithm (the latter will be described in detail in the next subchapter). The calculated distances themselves are written into a third data table named "Distances", which is then also used by the latter algorithm. It must be noted that in this third table, the relations without existing connections along the defined graph are represented with 0 distance and as such they are ignored by the optimization algorithm. The results of the optimization – the calculation of the shortest paths between the vertices along the defined graph – is written into a fourth data table named "ShortestPaths" (this will be explained in detail in the next subchapter). In the next figure (Fig. 2), the relationship between the previously mentioned data tables and the "NetworkMapping" method is presented, followed by a picture of a section of the "Distances" data table (Fig. 3).

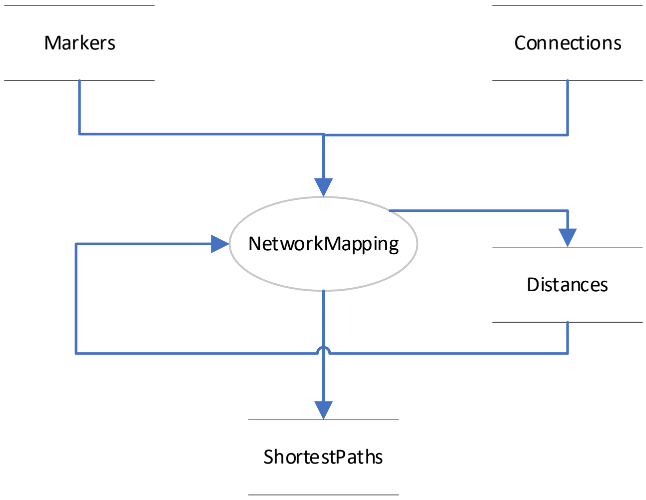

Figure 2. A flow-diagram showcasing the relationship between the "NetworkMapping" method and the mentioned four data tables

	length[m] 1	length[m] 2	length[m] 3	length[m] 4	length[m] 5	length[m] 6
1	0	5.92674540789462	0	13.0234062798486	0	0
2	5.92674540789462	0	6	0	13.0234062798486	0
3	0	6	0	0	0	13.0234062798486
4	13.0234062798486	0	0	0	5.92674540789462	0
5	0	13.0234062798486	0	5.92674540789462	0	6
6	0	0	13.0234062798486	0	6	0
7	0	0	0	13	0	0
8	0	0	0	0	12.9770070174135	0
9	0	0	0	0	0	12.9770070174135
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15						

Figure 3. A section of the structure of the "Distances" data table, which stores the distances between each Marker given in meters (0 means that there is no connection between the two Markers along the defined graph)

3.2. Finding the shortest paths for the AGVs

During the simulation the shortest paths are first calculated between every waypoint on the defined graph, essentially as part of the initialization process. As it was mentioned, this was implemented in the custom created "NetworkMapping" method, which writes each shortest path into a fourth data table, consequently named "ShortestPaths", a section of which is shown later in Figure 4. In the latter, each cell of the data table is itself a table which contains the list of the subsequent Markers that must be visited between the starting and the goal Marker, and the distances between each Marker along the route (the rows in the original table represent the starting locations and the columns represent the goal Markers).

As it was mentioned before, the applied algorithm itself is essentially Dijkstra’s algorithm, which here is applied to find the shortest paths between every vertex (Marker) along the graph. The “ShortesPaths” data table is not only used to store these optimal shortest paths, but it is also used by Dijkstra’s algorithm during the optimization process itself to store the actually found shortest paths leading to the already visited vertexes. By the time the optimization is finished due to the operation of the algorithm, each cell of the data table will contain the optimal shortest path. The entire expanded algorithm can be represented by the following pseudocode (naturally, the number of used rows and columns in the data table “ShortestPaths” is equal to the number of applied Markers in the actual model):

for **i** = 1 to the number of used rows in the data table “ShortestPaths”
  Set all vertices as unvisited.
  for **j** = 1 to the number of used columns in the data table “ShortestPaths”
    if there is an edge between Marker **i** and Marker **j**
      Append Marker **j** and the distance between Marker **i** and Marker **j** to the **sub-table ji** in the data table “**ShortestPaths**”. {In Plant Simulation, when addressing a cell in a table, the column number always precede the row number.}
  Mark **i** as a visited vertex.
  for **k** = 1 to the number of used columns in the data table “**ShortestPaths**”
    The vertex and the related distance value in the first non-empty **sub-table** in the **i**-th row of the “**ShortestPaths**” data table are selected as the temporary closest vertex to **i** with its distance value. If a closer vertex is found afterwards in the **i**-th row, it will replace the previous one, until the actual closest vertex is found with the completion of the current for cycle. The column number **k** of the actual closest vertex is passed on as **m**.
  repeat
    for **n** = 1 to the number of used columns in the data table “**ShortestPaths**”
      if there is an edge between Marker **m** and Marker **n** and a route was not found before from Marker **i** to Marker **n**
        Copy the route from the sub-table **mi** to the sub-table **ni** in the data table **“ShortestPaths”**
        Append Marker **n** and the distance between Marker **n** and Marker **m** to the **sub-table ni** in the data table “**ShortestPaths**”
      if there is an edge between Marker **m** and Marker **n** and a route was found before from Marker **i** to Marker **n**
        if the previous route is longer from Marker **i** to Marker **n** than the actually found one
          Empty the contents of the sub-table **ni** in the data table **“ShortestPaths”**
          Copy the route from the sub-table **mi** to the sub-table **ni** in the data table **“ShortestPaths”**
          Append Marker **n** and the distance between Marker **n** and Marker **m** to the **sub-table ni** in the data table “**ShortestPaths**”
    Mark **m** as a visited vertex.
    for **k** = 1 to the number of used columns in the data table “**ShortestPaths**”
      The route and its total distance value in the first non-empty **sub-table** in the **i**-

th row of the "**ShortestPaths**" data table is selected as the one that leads to the temporary closest vertex to the **i**-th Marker. If a shorter route is found afterwards in the **i**-th row to a yet unvisited vertex, it will replace the previous one, until the actual closest non visited vertex is found with the completion of the current for cycle. The column number **k** of the actual closest non visited vertex is passed on as **m**.

until all vertices are marked as visited

As it was stated, the algorithm always runs at the start of the simulation before any of the simulated operations would start, therefore by the time the AGVs start to leave the AGV pool, all the shortest paths between each vertex along the graph are calculated. As a result, giving a direction for an AGV simply means that the proper sub-table related to the starting location and to the destination must be selected from the "ShortestPaths" data table. In the following two Figures 4 and 5, both a section of the structure of the "ShortestPaths" data table and the contents of one of its sub-tables can be seen. In the latter, it can be seen how the shortest path for a particular relation is stored as an ordered list of the Marker objects which must be visited one after the other (the list does not contain the starting destination, but it contains the destination as the final Marker).

	table 8	table 9	table 10	table 11	table 12
1	table81	table91	table101	table111	table121
2	table82	table92	table102	table112	table122
3	table83	table93	table103	table113	table123
4	table84	table94	table104	table114	table124
5	table85	table95	table105	table115	table125
6	table86	table96	table106	table116	table126
7	table87	table97	table107	table117	table127
8	table88	table98	table108	table118	table128
9	table89	table99	table109	table119	table129
10	table810	table910	table1010	table1110	table1210
11	table811	table911	table1011	table1111	table1211
12	table812	table912	table1012	table1112	table1212
13	table813	table913	table1013	table1113	table1213
14	table814	table914	table1014	table1114	table1214
15					

Figure 4. A section of the structure of the "ShortestPaths" data table (each cell represents a sub-table containing the shortest path between two Markers along the graph)

	object 1	length[m] 2	string 3	string 4	string 5
1	*.Models.AGVrouting.Marker4	12.0234234363595			
2	*.Models.AGVrouting.Marker7	12			
3	*.Models.AGVrouting.Marker10	12			
4	*.Models.AGVrouting.Marker13	4.24264068711928			
5	*.Models.AGVrouting.Marker14	6			
6					

Figure 5. Picture of the contents of a sub-table in the "ShortestPaths" data table containing all the Markers that must be visited along the graph between the starting point and the destination (the travel distances are also stored and highlighted for each edge)

A clear advantage of using the previously outlined and proposed route-finding approach is that it provides a certain flexibility for the application of the general concept. This is because the Markers can be arbitrarily placed in a given Plant Simulation model and as such, it is relatively easy to reposition them (although the graph may need to be modified if the general topology is changed) and the graph structure can also be arbitrarily defined after the operational and environmental constraints are taken into account. Besides, the number of applied Markers can also be freely determined in the given model.

3.3. Description of the initial model

The developed initial model represents a relatively simple setup which contains an AGV Pool which generates the freely moving AGVs, besides two sources and one drain for the unit loads. The latter are equipped with robotic manipulators, while a rack system consisting of six modules represents a storage area, and finally two AGVs play the central materials handling role. The graphical representations of the customly developed methods and data structures for the model are placed at the top of the layout, above the operational area (the control methods for the robotic manipulators are placed adjacent to them). The top view of the layout in the 3D mode of the simulation environment can be seen in Figure 6.

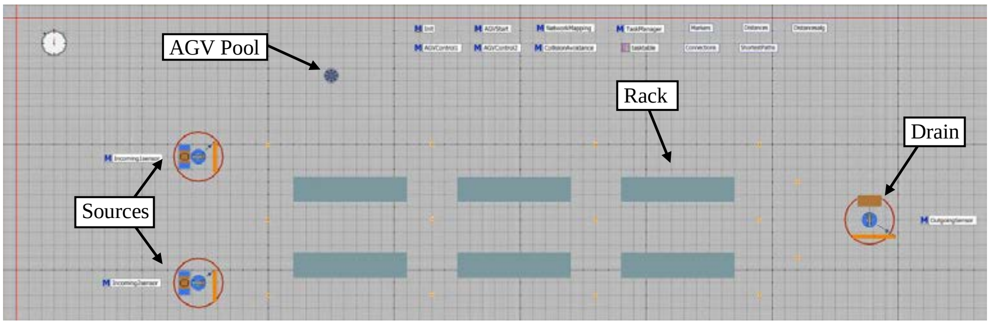

Figure 6. The layout of the initial model in the 3D mode of the simulation environment from the top view

As the developed initial model serves as a proof of concept mainly for the pathfinding approach, the material handling task done by the AGVs is relatively simple. Their job is basically to pick up the unit loads coming from the sources and deliver them to the drain, where they are unloaded and wait for the next cargo to come in from the sources. Both the loading and the unloading is done by the robotic manipulators (there is a standard animated robotic manipulator object called "PickAndPlace" in the simulation software, which is used here as well). Here it must be noted that short track segments are used in the initial model at the loading and unloading stations, mainly because they allow the placement of sensors that detect the arrival of the AGVs. The control methods of the manipulators are linked to these sensors in such a way that once an AGV passes a sensor, it is stopped by the control method and the loading or unloading procedure is initiated. Besides that, the control method also plays a part in the task management procedure developed for the model, but this will be described in the following chapter in more detail. Of course, another goal in the future is to replace these short track segments with Markers as well and change the operation of the model accordingly.

The rack system in the current initial model only serves the purpose to introduce the environmental constraints for the pathfinding problem, which of course means that the AGVs must navigate around the racks without colliding with them. This is accomplished with the proper utilization of the approach described in the previous two sub-chapters. In the initial model, 14 Markers are utilized, two of which are only used for the purpose to demonstrate the basic functionality of the current collision avoidance approach (the latter will also be discussed in more detail in the following chapter). A snapshot of the operation of the initial model can be seen in Figure 7.

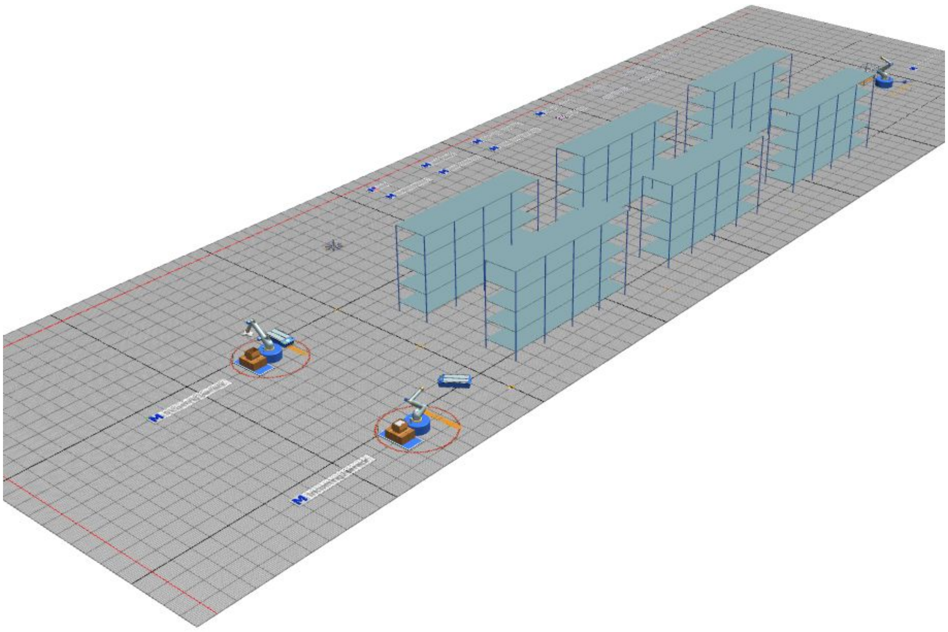

Figure 7. The operation of the initial model shown in the 3D view of the simulation environment

It is worth mentioning that at the beginning of each simulation run, the AGVs are sent to different locations in the model, shortly after which the task management method initiates and starts to send the vehicles to the newly arisen tasks (in this case, the transportation tasks arising at the two sources). As it was mentioned, the AGVs drive out from the AGV Pool object, with a short delay in between them. The movement orders themselves are mainly transmitted to the AGVs through three custom designed methods. The first of which initiates the latter two which then directly command the AGVs according to the tasks provided by the task management method (in fact, after the vehicles reached their initial locations, only one of these methods is used throughout the rest of the simulation):

The movement commands themselves are based on the previously described "ShortestPaths" data table, which means that the control method of the AGV only has to assign the adequate shortest path stored in the previous table to the starting position and destination pair provided by the task management method (this is achieved by an algorithm that locates the closest Marker to the starting position and to the destination and then selects the shortest path related to the identified Marker pair from the data table). It then commands the AGV to traverse the selected path – expanded by the destination provided by the task management method – and the rest is performed by the vehicle itself.

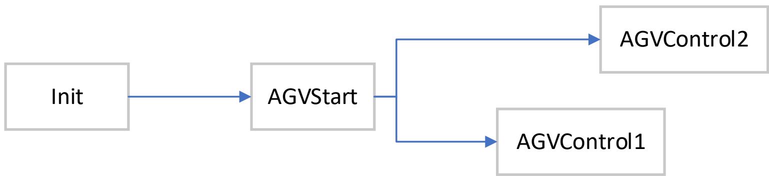

Figure 8. The initialization sequence of the AGV control algorithms (after the two AGVs reached their starting positions, from then onward only the "AGVControl1" method is used by the task management method to send both AGVs to the locations of their current tasks)

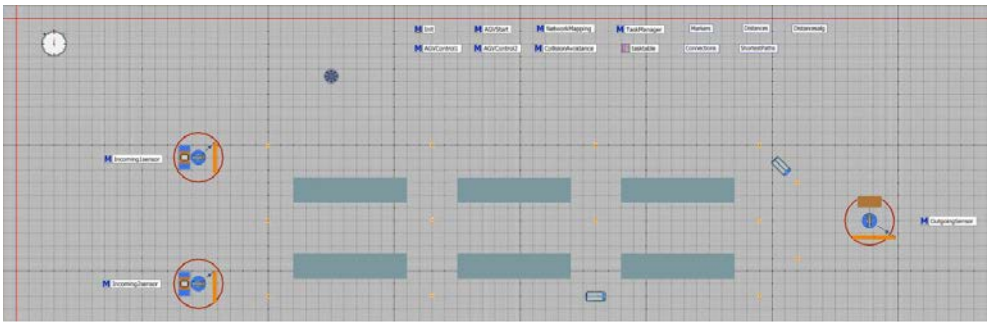

Figure 9. At the start of the simulation, the AGVs are sent to different locations in the model (3D mode, top view)

4. Current limitations of the initial model and future research directions

As it could be seen in the previous chapter, the developed initial model and the utilized approach mainly focused on the pathfinding problem of freely moving AGVs. This is however only the first step in solving the larger problem set of the simulation modelling of such vehicles, as they also must be able to avoid collisions with each other, must be able to adapt dynamically to quickly changing tasks, also need to be able to avoid collisions with other objects and actors – including human operators – on the factory floor in unforeseen situations, etc. To a basic degree, some of these considerations were already considered in the initial model, but only in a limited sense, as their complete solution will require additional significant work in the future. Naturally, these problems receive significant attention and are intensively studied in the related literature as well, for example also in [2], [8], [9], [12], [18], [20], [21], [22], [23], [25] and [26].

In the followings, the problems related to the collision avoidance between the AGVs and to the solution of task management will be discussed in more detail, while other questions will be also mentioned in a third subchapter.

4.1. Collision avoidance between the vehicles

One of the most important and in many ways the most complex problem of freely moving AGVs is posed by the question of how to avoid the collision of the vehicles with each other. This is a sophisticated problem because once the vehicles are allowed to move in any direction, the number of possible collision points theoretically becomes infinite, even in the case of only two vehicles and if the pathways are outlined by markers. Therefore, to solve this problem, intelligent collision avoidance algorithms are needed. Of course, this is a very important problem in the real world, where it is solved with the application of numerous sensor systems and sophisticated control algorithms on each AGV, while high-level strategies like zone control are also applied to prevent the emergence of the possibility of collisions in the first place.

In the current research, the goal would be to develop an algorithm in the frame of the simulation that is able to model the collision avoidance behaviours of freely moving AGVs with good precision, which could be later augmented with advanced zone control strategies. First, a rule-based approach is chosen, where the given AGV applies the collision avoidance algorithm once the distance between itself and another AGV falls below a predefined limit. In the initial model, the collision avoidance method in fact constantly runs and detects once the distance between the first and the second vehicle is fallen below the limit. To avoid a deadlock situation, in the current model it is always the second AGV which takes action once the distance limit is breached.

Currently, the algorithm follows a quite simple rule: once the distance is fallen below 8 meters, the second AGV always stops and it remains in that state until the distance becomes greater again (with the further development of the collision avoidance algorithm, this relatively large safety distance could be reduced in the future). This is a good solution in certain situations like the one presented in the Figure 10, but of course there are a lot of other situations where this would still lead to a collision, for example if the vehicles are heading to opposite directions on the same path, or if the other vehicle comes from the side and its trajectory would still intersect the position of the stopped AGV.

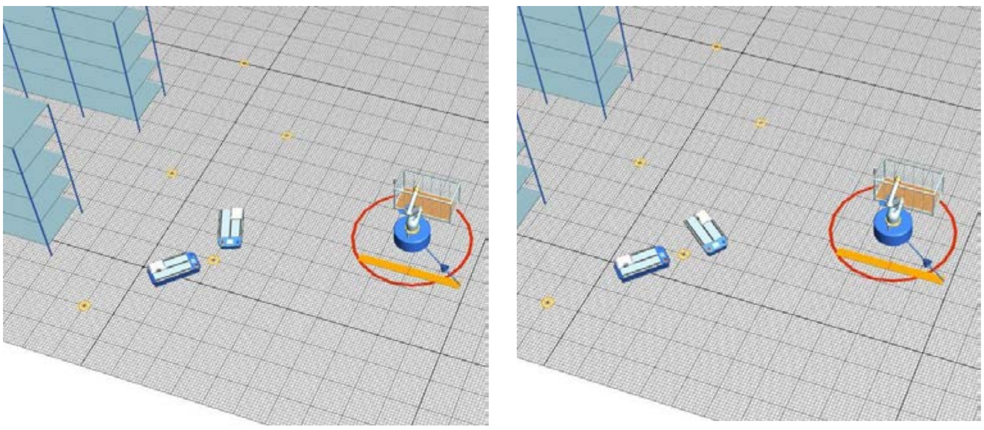

Figure 10. The developed basic collision avoidance algorithm simply stops the second AGV if it gets closer to the other than 8 meters, but this can only prevent collisions in certain situations like the one depicted above (the second AGV is stopped while the first one, which comes from the upper direction, turns in front of it towards the drain)

Another drawback of this solution is that the second AGV also stops in situations where it would be not necessary, for example if the two vehicles are heading to completely different directions, but the distance between them still falls below 8 meters temporarily. It must be also noted that there are situations where the vehicles might need to be closer to each other, for example when they are waiting for the loading/unloading operations or when they just leave their parking positions. In the latter cases, the current model in fact utilizes the ability of the track segments placed before the robotic manipulators to stop the vehicles before a collision would occur (it must be noted that for the track objects, an arbitrary safety distance can be set and distance control methods can be easily designed, but this is not in the focus of the current research). Therefore, while the AGVs are on the track, the collision avoidance method does not apply as the vehicles automatically stop before a collision. It also won't intervene when they leave the track in quick succession, as currently the algorithm does not trigger if the distance gets suddenly below 2 meters without breaching the 8 meter limit first, but in the current initial model this can only occur in a single situation, when the vehicles leave their parking position and then head for different directions, during which a collision in the current form of the model is operationally not possible. Naturally, in a later stage of the research, the short track segments should also be ideally replaced with Marker objects and the described situations should also be fully considered in the collision avoidance method. In Figure 11 the two AGVs can be seen parking at the track segment of the drain, waiting for the next material handling tasks:

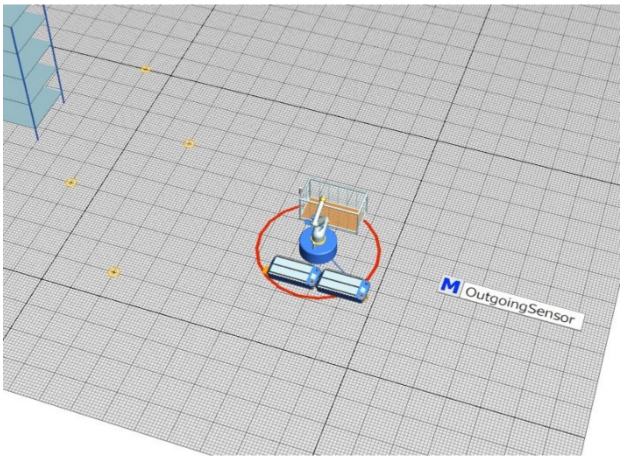

Figure 11. The two AGVs are waiting on the track segment of the drain for the upcoming material handling tasks

Overall, the developed simple collision avoidance method for the initial model can solve a smaller set of the possible situations, but a more sophisticated algorithm is needed to avoid the large majority of the possible collision events. As it could be seen from the previous, this is a very complex problem which represents an active field of research. In the frame of the current research, the goal is to first try to expand the applied rule-based approach in order to be able to handle more complex situations and a larger number of vehicles. However, it is possible that other types of approaches will also be needed to achieve this goal. For example, in [8] machine learning is used to create a knowledge base which is utilized for the re-routing of the vehicles to avoid collisions. As another example, in [12] the motion co-ordination and conflict resolution between the vehicles is realized through the use of a state lattice based approach, together with the application of private zones for the vehicles and a vehicle priority scheme. In yet another example, in [21] a method based on collision classification and time windows is applied to solve this problem. These examples also strongly imply that in the frame of the current research, the simple rule-based approach will have to be augmented with additional techniques in the future. Furthermore, if multiple vehicles are affected in a potential collision situation, naturally some form of prioritization and/or coordination is also needed between them in order to be able to determine the proper action for each vehicle, as it is also implied by the above examples (in the current model, it is always the second AGV which stops, but in case of a larger number of vehicles, the priority ranks of the different AGVs can be determined according to a given situation, based on a yet to be elaborated prioritization concept).

In parallel, the collision avoidance algorithm could also be augmented with the utilization of some form of zone control method, as in case of a large number of vehicles, at some point the collisions might become unavoidable regardless of the utilized approach, or alternatively the system could get stuck in a deadlock situation. Zone control strategies are frequently used in the literature. For example, this approach is utilized also in [2], [8], [9], [18], [22], [23], [25], [26], and as mentioned also in [12] in the form of private zones for the vehicles.

Finally, the problem of avoiding collisions with other types of objects – including those which represent human operators – should also receive greater attention in the future stages of the research, but this will be mentioned in the third subchapter.

4.2 Task management

Currently, the developed task management method constantly monitors the state of the two sources and if a new unit load appears, it assigns the first available AGV (which currently has no other duty) to the transportation task and calls the AGV control method with the given destination. Then, once the assigned vehicle reached the given source and the loading procedure was completed, the control method of the robotic manipulator at the source again calls the AGV control method and sends the vehicle back to the drain, which also serves as a parking station in the current model. It must be noted that the task management procedure also uses a few customly defined attributes for the AGVs and a simple task table that contains whether a vehicle is currently available or occupied with the previous task.

A clear limitation of the previous approach is that the task management method only controls the vehicles at the start of a new task and once the first phase of that is completed, the AGVs receive their instructions locally to return to their starting destination. This limits the complexity of the issuable tasks, while it also makes it difficult to apply more sophisticated task management strategies. Therefore, a clear goal in the future would be to develop a task management approach that makes it possible to direct the AGVs during the full task cycle. Based on that, more complex task management strategies could be applied which consider a number of different factors, both in terms of the present situation and in terms of the projected future state of the system. This of course will require further development, but it must be noted that the previously described collision avoidance problem presents by far the most challenges. The two are also somewhat connected, as in case of a larger number of vehicles, the task management and the zone control strategy should be synchronized together.

4.3 Other considerations

Besides the previously mentioned areas, there are other considerations that should be considered during the further development of the simulation model. One of these aspects is naturally the battery level of the AGVs. However, it must be noted that this is a standard attribute of the transporter object used in the model, so the only task that should be solved in this regard is how to integrate the recharging of the vehicles into the task management strategy.

Another aspect of the model that should be developed further is the collision avoidance regarding other objects in the simulation environment. Right now, the vehicles can avoid other types of objects simply through the proper placement of the Markers, as currently all other elements of the model are stationary (or quasi stationary in case of the robotic manipulators). This is a good solution when there are no unexpected situations, but with the later introduction of more vehicles and especially human operators into the simulation model, such unexpected situations can easily happen where other types of objects suddenly can get in the way of the vehicles (even if these objects are stationary and they simply get in the way because the vehicles have to manoeuvre around each other). Therefore, in the future the distances between the AGVs and all other nearby objects should also be considered in the collision avoidance algorithm. In a sense, this is a simpler problem than the collision avoidance with other vehicles, as the generally applied procedure when an AGV detects an unexpected obstacle in front of it is to just simply stop and wait until the path is clear. Still, in the future this procedure should be applied in relation to all nearby objects relative to the vehicles, naturally including the objects representing human operators, which would also exhibit movement (in the simulation environment, the Worker object is used for the latter purpose). Naturally, this will also add another level of complexity to the previously described problem of collision avoidance with other AGVs.

5. SUMMARY

The aim of the paper was to introduce a modelling approach for the simulation of freely moving AGVs in a discrete event simulation environment and to demonstrate its application through an initial model. The proposed approach utilizes a widely applied graph-based pathfinding algorithm (Dijkstra’s algorithm) for the calculation of the optimal routes for the AGVs, which then navigate themselves to the given destination along the vertices of the calculated path. The initial model was created using the Siemens Tecnomatix Plant Simulation software.

While the created model mainly focused on the pathfinding problem, this is only the first major step in the development of the proposed approach, as the significantly complex problem of collision avoidance also must be fully solved, while other aspects of the problem set also must be considered. These represent significant challenges, but also provide multiple avenues for the continuation of the research in the future. Naturally, in the following phases of the research, various performance tests are also planned on the model in order to evaluate its applicability in different application scenarios. The final goal would be the development of a general discrete event simulation-based modelling approach for freely moving AGVs which can be flexibly applied in a wide variety of logistics environments, and especially in the complex environment of manufacturing logistics. As the utilization of autonomous mobile robots is clearly on the rise in the Industry 4.0 paradigm, the application of such modelling methods is also becoming increasingly important, both in the present and especially in the future.

Acknowledgements

Project no. 2019-2.1.11-TÉT-2020-00195 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the 2019-2.1.11-TÉT funding scheme.

References

[1] Ma, N., Zhou, C. & Stephen, A. (2021). Simulation model and performance evaluation of battery-powered AGV systems in automated container terminals. *Simulation Modelling Practice and Theory* 106, 102146. https://doi.org/10.1016/j.simpat.2020.102146

[2] López, J., Zalama, E. & Gómez-García-Bermejo, J. (2022). A simulation and control framework for AGV based transport systems. *Simulation Modelling Practice and Theory* 116, 102430. https://doi.org/10.1016/j.simpat.2021.102430

[3] Luo, L. & Zhao, N. (2022). An efficient simulation model for layout and mode performance evaluation of robotic mobile fulfillment systems. *Expert Systems with Applications* 203, 117492. https://doi.org/10.1016/j.eswa.2022.117492

[4] Viharos, A. B. & Németh, I. (2018). Simulation and scheduling of AGV based robotic assembly systems. *IFAC-PapersOnLine* 51(11), 1415-1420. https://doi.org/10.1016/j.ifacol.2018.08.317

[5] Bhosekar, A., Ekşioğlu, S., Işık, T. & Allen, R. (2021). A discrete event simulation model for coordinating inventory management and material handling in hospitals. *Annals of Operations Research* 1-28. https://doi.org/10.1007/s10479-020-03865-5

[6] Greasley, A. (2020). Architectures for Combining Discrete-event Simulation and Machine Learning. *Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2020)* 47-58. https://doi.org/10.5220/0009767600470058

[7] Fu, J., Zhang, J., Ding, G., Qin, S. & Jiang, H. (2021). Determination of vehicle requirements of AGV system based on discrete event simulation and response surface methodology. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 235(9), 1425-1436. https://doi.org/10.1177/0954405421995602

[8] Klaas, A., Laroque, C., Dangelmaier, W. & Fischer, M. (2011). Simulation aided, knowledge based routing for AGVs in a distribution warehouse. *Proceedings of the 2011 Winter Simulation Conference (WSC)* 1668-1679. IEEE. https://doi.org/10.1109/WSC.2011.6147883

[9] Kesen, S. E. & Baykoç, Ö. F. (2007). Simulation of automated guided vehicle (AGV) systems based on just-in-time (JIT) philosophy in a job-shop environment. *Simulation Modelling Practice and Theory* 15(3), 272-284. https://doi.org/10.1016/j.simpat.2006.11.002

[10] Um, I., Cheon, H. & Lee, H. (2009). The simulation design and analysis of a flexible manufacturing system with automated guided vehicle system. *Journal of Manufacturing Systems* 28(4), 115-122. https://doi.org/10.1016/j.jmsy.2010.06.001

[11] Digani, V., Sabattini, L., Secchi, C. & Fantuzzi, C. (2014). Hierarchical traffic control for partially decentralized coordination of multi AGV systems in industrial environments. In *2014 IEEE international conference on robotics and automation (ICRA)* 6144-6149. IEEE. https://doi.org/10.1109/ICRA.2014.6907764

[12] Draganjac, I., Miklić, D., Kovačić, Z., Vasiljević, G. & Bogdan, S. (2016). Decentralized control of multi-AGV systems in autonomous warehousing applications. *IEEE Transactions on*

Automation Science and Engineering 13(4), 1433-1447. https://doi.org/10.1109/TASE.2016.2603781

[13] Azimi, P., Haleh, H. & Alidoost, M. (2010). The selection of the best control rule for a multiple-load AGV system using simulation and fuzzy MADM in a flexible manufacturing system. *Modelling and Simulation in Engineering* 2010. https://doi.org/10.1155/2010/821701

[14] Neradilova, H. & Fedorko, G. (2017). Simulation of the supply of workplaces by the AGV in the digital factory. *Procedia engineering* 192, 638-643. https://doi.org/10.1016/j.proeng.2017.06.110

[15] Hsueh, C. F. (2010). A simulation study of a bi-directional load-exchangeable automated guided vehicle system. *Computers & Industrial Engineering* 58(4), 594-601. https://doi.org/10.1016/j.cie.2009.12.008

[16] Fedorko, G., Molnar, V., Vasil, M. & Hanzl, J. (2018). Application of the Tecnomatix Plant Simulation Program to modelling the handling of ocean containers using the AGV system. *NAŠE MORE: znanstveni časopis za more i pomorstvo* 65(4 Special issue), 230-236. https://doi.org/10.17818/NM/2018/4SI.12

[17] Pedan, M., Gregor, M. & Plinta, D. (2017). Implementation of automated guided vehicle system in healthcare facility. *Procedia engineering* 192, 665-670. https://doi.org/10.1016/j.proeng.2017.06.115

[18] Fanti, M. P., Mangini, A. M., Pedroncelli, G. & Ukovich, W. (2018). A decentralized control strategy for the coordination of AGV systems. *Control Engineering Practice* 70, 86-97. https://doi.org/10.1016/j.conengprac.2017.10.001

[19] Vivaldini, K. C. T., Galdames, J. P. M., Pasqual, T. B., Sobral, R. M., Araújo, R. C., Becker, M. & Caurin, G. (2010). Automatic routing system for intelligent warehouses. In *IEEE International Conference on Robotics and Automation* 1, 1-6.

[20] Möhring, R. H., Köhler, E., Gawrilow, E. & Stenzel, B. (2005). Conflict-free real-time AGV routing. In *Operations research proceedings* 2004, 18-24. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-27679-3_3

[21] Zhang, Z., Guo, Q., Chen, J. & Yuan, P. (2018). Collision-free route planning for multiple AGVs in an automated warehouse based on collision classification. *IEEE Access* 6, 26022-26035. https://doi.org/10.1109/ACCESS.2018.2819199

[22] Fransen, K. J. C., Van Eekelen, J. A. W. M., Pogromsky, A., Boon, M. A. & Adan, I. J. (2020). A dynamic path planning approach for dense, large, grid-based automated guided vehicle systems. *Computers & Operations Research* 123, 105046. https://doi.org/10.1016/j.cor.2020.105046

[23] Liu, C. I., Jula, H., Vukadinovic, K. & Ioannou, P. (2004). Automated guided vehicle system for two container yard layouts. *Transportation Research Part C: Emerging Technologies* 12(5), 349-368. https://doi.org/10.1016/j.trc.2004.07.014

[24] Ho, Y. C. & Chien, S. H. (2006). A simulation study on the performance of task-determination rules and delivery-dispatching rules for multiple-load AGVs. *International journal of production research* 44(20), 4193-4222. https://doi.org/10.1080/00207540500442401

[25] Micieta, B., Edl, M., Krajcovic, M., Dulina, L., Bubenik, P., Durica, L. & Binasova, V. (2018). Delegate MASs for coordination and control of one-directional AGV systems: a proof-of-concept. *The International Journal of Advanced Manufacturing Technology* 94(1), 415-431. https://doi.org/10.1007/s00170-017-0915-8

[26] Moorthy, R. L., Hock-Guan, W., Wing-Cheong, N. & Chung-Piaw, T. (2003). Cyclic deadlock prediction and avoidance for zone-controlled AGV system. *International Journal of Production Economics* 83(3), 309-324. https://doi.org/10.1016/S0925-5273(02)00370-5

[27] Cservenák, Á. (2021). Simulation of a mobile robot's motion. Acad. J. Manuf. Eng. 9, 80-88.