Simulation in Produktion
und Logistik 2019
Matthias Putz & Andreas Schlegel (Hrsg.)
Wissenschaftliche Scripten, Auerbach 2019

Test Course Creation for Automated Guided Vehicles

Erstellung von Testparcours für fahrerlose Transportfahrzeuge

Christoph Allmacher, Philipp Klimant, TU Chemnitz, Chemnitz (Germany),
christoph.allmacher@mb.tu-chemnitz.de, philipp.klimant@mb.tu-chemnitz.de

Abstract: The development process of automated guided vehicles (AGVs) includes thorough testing of the navigation capabilities. Therefore, adequate test courses are required for virtual and real tests before commissioning the AGV at the customer's site. The standard F3244-17 introduces 2D area shapes and a method for testing navigation capabilities. This paper deals with the extension of the standard F3244-17 2D area shapes, introduces obstacles and adjusts the test method accordingly. Furthermore, it proposes a corresponding method for test course creation based on a given layout. The content is validated via simulation in an intralogistics environment using Unity3D. The experiments indicate that the results achieved in test courses are transferable to the original environment.

1 Introduction

The market for automated guided vehicles (AGVs) with free navigation is rapidly growing (Fitz 2018). In consequence, developers have to overcome challenges caused by the increasing complexity of the control software and the navigation algorithms necessary for providing free navigation. In order to meet these challenges, developers rely on simulation methods. The goal is to test the navigation algorithms and the control software in early stages of the development process. However, additional tests using a physical prototype have to be conducted before delivering the AGV to the customer since the simulation is not an exact replication of the real AGV behaviour.

For both virtual and real tests, an adequate test course is required. It defines the shape and content of the environment the AGV is navigating in. Test courses for virtual scenarios are required to be compact and concise to reduce modelling efforts and costs. The same requirements apply to real test courses since available facilities are often limited in their size. Furthermore, test courses should be geometrically well defined to be reproducible so that the virtual test course matches the real one.

This paper targets the extension of existing standards and methods to better meet industrial requirements and to propose an additional method for creating test courses based on real layouts. Furthermore, the goals also include the validation that the performance of navigation algorithms within deduced test courses is comparable to the original environment.

Therefore, the state of the art will be presented. Then existing issues will be analysed and new additions and methods will be introduced to solve them. A final conclusion will be drawn after the validation of the methods.

2 State of the Art

In the following an overview of the current work on navigation algorithms of AGVs, on simulation methods and tools for AGVs, and on test methods and environments is presented.

2.1 Navigation Algorithms of Automated Guided Vehicles

The core ability of a freely moving AGV is its navigation, which includes localisation, path planning and obstacle avoidance (Fan et al. 2018). The common localisation method is simultaneous localisation and mapping (SLAM) in combination with filter techniques such as Kalman or Particle filters (Kohlbrecher et al. 2011; Montemerlo et al. 2002). Nevertheless, other approaches exist, e.g. triangulation (Font and Batlle 2006). The fleet manager often conducts global path planning by using predefined roadmaps (Uttendorf et al. 2016). Local obstacle avoidance is achieved by using techniques such as the dynamic window approach, artificial potential fields or fuzzy logic (Fox et al. 1997; Allmacher et al. 2018; Li et al. 2012). These navigation algorithms have been thoroughly researched, but their parameter settings can be optimised depending on the specific AGV and the target use case.

2.2 Simulation Methods and Tools

Simulation methods such as software-in-the-loop (SIL) or hardware-in-the-loop (HIL) can be used to test the navigation algorithms in early development stages. In these two methods, the real control software is coupled to a virtual AGV, containing simulated sensors and actuators (Makris et al. 2012). To generate exploitable sensor data, the virtual AGV drives within a modelled environment which shall represent the target use case. Numerous appropriate tools for simulation purposes exist in the field of robotics such as GAZEBO, SimTwo, MORSE or Unity3D (Costa et al. 2011; Koenig and Howard 2004; Echeverria et al. 2011; Allmacher et al. 2018).

2.3 Test Environments and Methods

The shape and content of the environment have to be defined regardless of the utilised simulation tool. An environment for testing the navigation capabilities of the AGV (hereafter called a test course) shall represent the target use case. A methodical approach towards creating test courses for AGV navigation algorithms was proposed in the standard F3244-17 (ASTM F3244-17). It defines seven generic 2D area shapes (2DA-shapes) which can be selected to create a test course:

- A straight aisle
- A left/right turn with or without interior chamfers
- A dual intersection with or without interior chamfers

Figure 1 shows the example of the dual intersection without interior chamfers. Here the prefix "dual" indicates that two other paths join the current path at the intersection. Furthermore, F3244-17 introduces a method to test navigation algorithms. The AGV is placed behind the start line (area A in Fig. 1). If the AGV crosses the finish line of either area B or C without collision, the test was successful. The calculated mean velocity of the AGV can be used as benchmarking criterion. The focus of the method lies on its reproducibility. Its result is the reliability of the navigation algorithms, which is determined by the ratio of failed to passed tests. Bostelman et al. proposed an extension to F3244-17 for undefined areas (Bostelman et al. 2016).

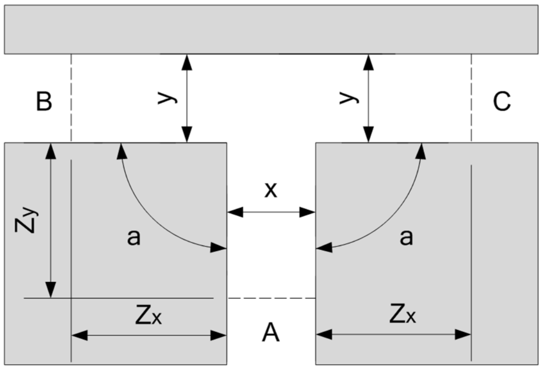

Figure 1: *Dual intersection without interior chamfers based on (ASTM F3244-17).*

3 2D Area Shapes and Methods for Test Course Creation

Initially, the AGV developer possesses certain standard information about the use case. It includes the layout of the target environment and usually information on the objects that are present in this environment. The 2DA-shapes proposed in the standard F3244-17 can be used to create test courses based on the layout. Figure 2 illustrates the layout of a fictive intralogistics scenario, which is based on a real use case. The AGV is operating in three different areas. On the left side, euro-pallets form two-way paths that are connected via triple intersections. A wide-open area is located in the middle. On the right side, racks form one-way paths tighter than the broa der euro-pallet paths. The fictive scenario is representative for many intralogistics and production layouts, which are characterised by their well-structured orthogonal pattern. Thus, it qualifies as a reference for the deduction of the following extensions and methods.

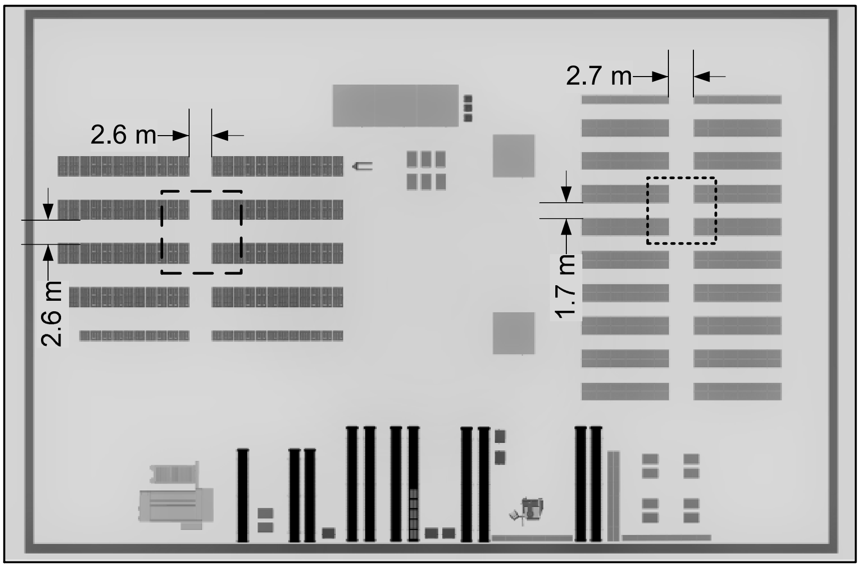

Figure 2: Layout of a fictive intralogistics scenario.

3.1 Extending the Standard F3244-17

Comparing the fictive intralogistics scenario (Fig. 2) with the available 2DA-shapes of the standard F3244-17 results in the conclusion that it does not cover triple intersections (dashed/dotted rectangles in Fig. 2), in which three other paths join the current path. However, triple intersections are highly relevant for the validation of the AGV's navigation algorithms. Due to the higher number of paths joining, more AGVs can encounter either each other or other dynamic obstacles, complicating the path planning and obstacle avoidance. Thus, the addition of a triple intersection is proposed in the style of F3244-17 with or without interior chamfers, extending the number of available 2DA-shapes to nine. This design is shown in Figure 3.

Concerning the two-way paths and intersections, it is required to consider the scenario in which two AGVs are facing each other, or an obstacle blocks part of the path. Thus, two obstacle types are proposed in the shape of a square and a circle. Square-shaped obstacles shall represent other AGVs, forklifts or trolleys, while circular-shaped ones shall represent humans.

Similar to the 2DA-shapes, the method of F3244-17 needs to be adjusted. The original method defines that the AGV is initially placed behind the start line. Dimensions, initial placement and movement trajectory also have to be defined for the proposed obstacles. The dimensions of the obstacles depend on the narrowest path modelled. The width and length of the square-shaped obstacle as well as the diameter of the circular-shaped obstacle amount to one third of the path width (see Fig. 4). A static obstacle can be placed freely under the condition that it does not intersect with a central line of a path (dotted and dashed line in Fig. 4).

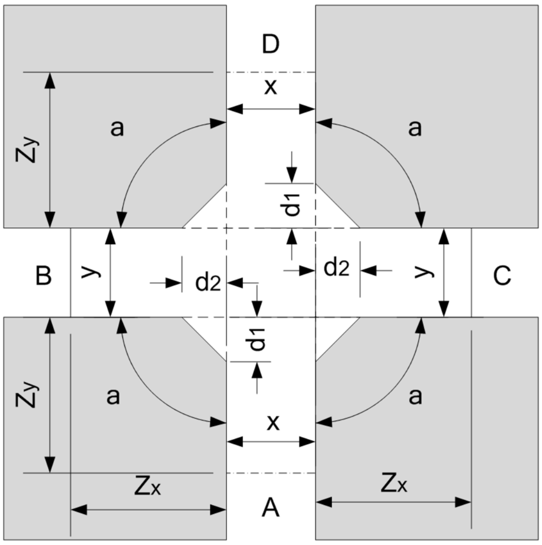

Figure 3: *Triple intersection with interior chamfers extends F3244-17*

Dynamic obstacles are initially placed in the middle of the right-hand lane of a path. From this starting point, they can move along one of three possible trajectories at a constant velocity: left turn, right turn or straight-ahead (see Fig. 4). In case of a turn, the obstacle follows a circle with the radius of a quarter or three quarters of the path width, depending on a right or left turn, respectively. The square-shaped obstacle is always oriented along the trajectory's tangent.

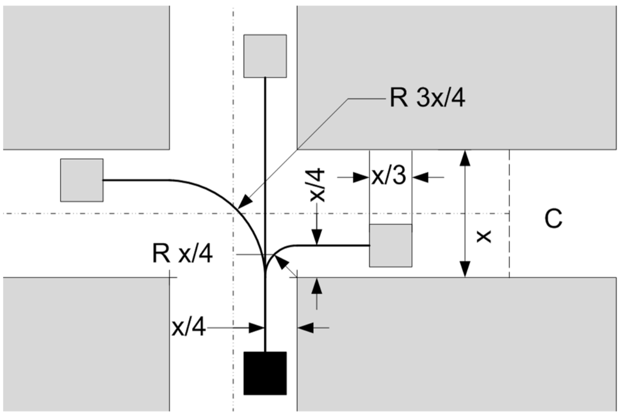

Figure 4: *Initial placement and trajectory of obstacles*

3.2 Test Course Creation Method

Based on the given use case and defined by the layout, 2DA-shapes have to be selected and their parameters have to be defined to create a test course. However, the state of the art does not contain a method to select and parameterise the 2DA-shapes of F3244-17 and the addition described in section 3.1. Hence, the test course creation method

(TCC) is developed to guide a user of F3244-17, including the presented addition through the correct selection process of 2DA-shapes based on the given layout.
The TCC is developed based on critical geometrical characteristics which influence testing the navigation capabilities of the AGV. As described in section 2.1, the AGV's navigation contains three main tasks: localisation, path planning and obstacle avoidance, and control. Concerning localisation, critical characteristics for SLAM algorithms are large, highly dynamic environments that raise the scan matching error and cause discrepancies in the mapping process. In contrast, small well-defined environments such as the 2DA-shapes are uncritical regarding the localisation. Similar to the localisation, dynamic high-speed situations are challenging the control of the actuators. Concerning the linear velocity, the 2DA-shapes, and especially the turns and intersections, are not intended to be large enough to create high-speed situations. In contrast, the rotary velocity control can be tested within the intersections. Lastly, the path planning and obstacle avoidance is limited by the available space. Particularly narrow environments and numerous dynamic obstacles with interjecting trajectories are critical for these functions. In summary, on the one hand the 2DA-shapes are suited to test the algorithms of path planning and obstacle avoidance, and, to some extent, to check the control algorithms regarding the rotary velocity. On the other hand, the 2DA-shapes are less challenging for the localisation algorithms, especially SLAM, and the linear velocity control.
Consequently, relevant critical characteristics for the selection of 2DA-shapes are based on the algorithms of path planning and obstacle avoidance. As stated, these critical characteristics include in particular the narrowness of the environment. Furthermore, dynamic obstacles with interjecting trajectories are critical, which are encountered within intersections or two-way paths. Therefore, the order of an intersection is critical because higher order intersections are more likely to feature a greater number of dynamic obstacles. Due to the possibility of dynamic obstacles, two-way paths must be considered even though they are most often wider than one-way paths.
Based on the two criteria of *narrowness* and *dynamic obstacles,* a maximum of three different test courses can be selected, while weighting *narrowness* higher than *dynamic obstacles*:

1. Narrowest one-way path with narrowest adjacent intersection
2. Narrowest two-way path with narrowest adjacent intersection
3. Highest order intersection

The test course of the narrowest path will be created including the narrowest adjacent intersection for testing the rotary velocity control. Figure 5 depicts the method for selecting the suitable 2DA-shapes and for creating test courses. The detailed use will be described in section 4. There might not always be three different test courses if criteria are overlapping. For example, if the narrowest two-way path is also the narrowest path of the layout, the one-way path is not required.

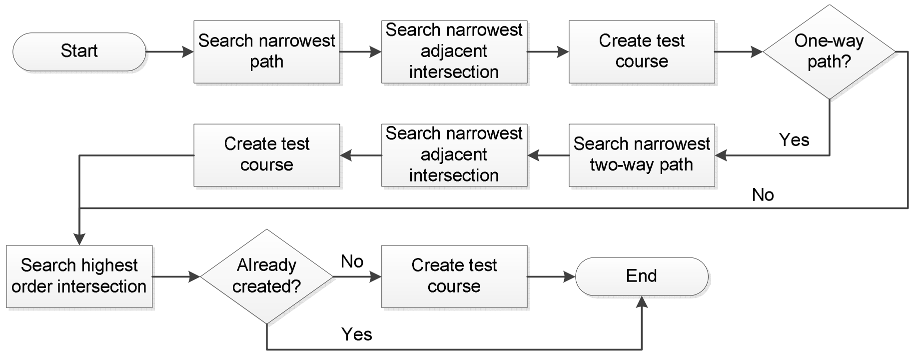

Figure 5: *Test course creation method based on a given layout*

4 Validation

The goal of this section is to demonstrate the TCC, including the extensions, and to validate whether the behaviour of the AGV's control software within deduced test courses is comparable to the original environment. For the validation, Unity3D is used to simulate the fictive intralogistics scenario (see Fig. 2) constituting as environment. Furthermore, the virtual AGVs containing the required sensors and actuators and dynamic obstacles are simulated. Additionally, a control programme is simulated which provides the AGV's control software with tasks. This control software is tested using SIL. It is connected to the Unity3D simulation, receiving tasks from the control programme and measurement data from the sensors. Then the control software is executed, including steps for localisation, path planning and obstacle avoidance and control. The resulting velocity commands are sent back to the actuators simulated in Unity3D. The setup is shown in Figure 6.

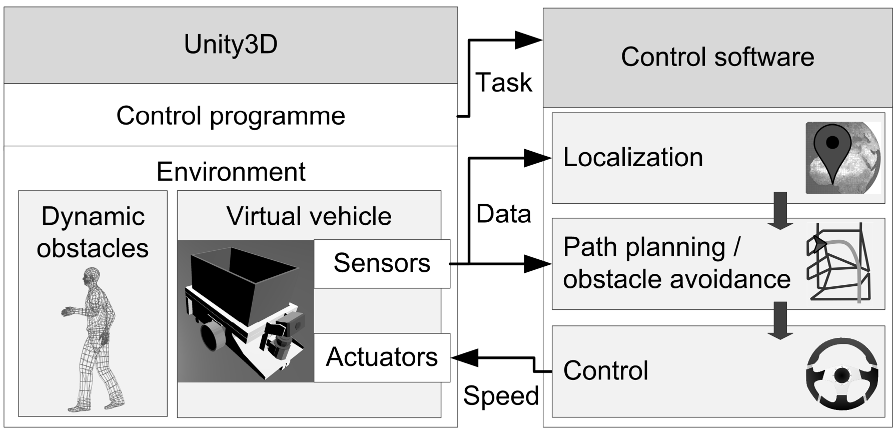

Figure 6: *Simulation setup using Unity3D*

The TCC is used to create the test courses based on the fictive intralogistics scenario. The narrowest path is found at the right side of the layout, with a width of 1.7 m, and the narrowest adjacent intersection is a triple intersection also measuring 1.7 m (dotted rectangle in Fig. 2). Consequently, the triple intersection is selected as the 2DA-shape and the test course is created with the stated parameters. Since this test course is a one-way path, the narrowest two-way path is searched and found on the left side, measuring a width of 2.7 m. Its narrowest adjacent triple intersection measures 2.7 m (dashed rectangle in Fig. 2), and consequently this second test course is created similarly to the first one. Lastly, the highest order intersection is a triple intersection, thus no additional course is required. The method results in two out of three possible courses (see Fig. 7), which are modelled in Unity3D and are used as the environments in the simulation setup.

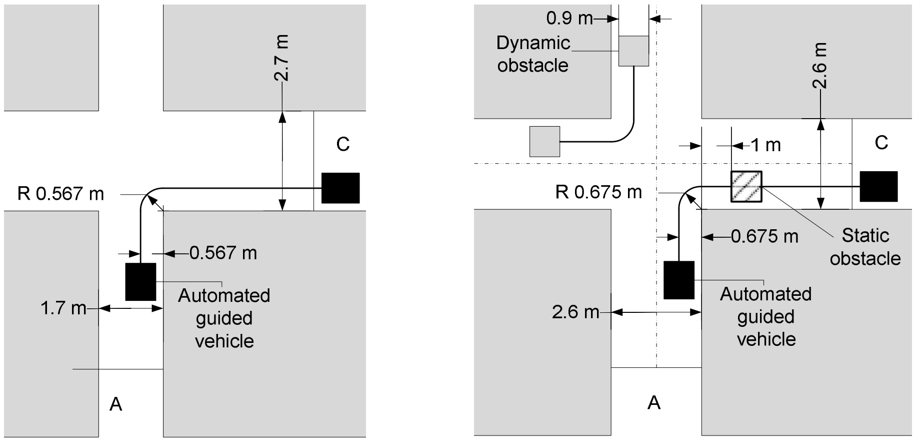

Figure 7: *Two test courses based on the fictive intralogistics environment*

The plausibility of the test courses and extensions to F3244-17 is tested in three experiments, which are conducted both in the deduced test courses and in the original environment by using the simulation setup. The experiments are chosen to test the traditional F3244-17 method as well as the newly introduced static and dynamic obstacles. Thus, the first experiment is conducted in the one-way test course without any obstacles (left side in Fig. 7). The AGV is placed behind the start line in section A. Reaching the intersection, it performs a right turn to end the experiment behind the finish line in section C. The second and third experiments are conducted in the two-way path test course (right side in Fig. 7). In these two experiments, the AGV starts in section A and finishes in section C, similar to the first experiment. In the second experiment, a static rectangular-shaped obstacle is placed on the trajectory of the AGV in front of the finish line. In the third experiment, a dynamic obstacle is placed in section D. Mirroring the AGV, it performs a right turn finishing in section B. The AGV will encounter the obstacle within the intersection, thus limiting the driving space drastically. Each experiment is conducted 22 times in the test course and in the original environment. Similar to the traditional method, the time is recorded as well as the result whether the experiment passed or failed. Table 1 shows the results, illustrating that all experiments in the test courses and in the original environment were passed and none failed while using the same parameters for the navigation algorithms. Furthermore, the recorded time indicates similarity in behaviour. In experiments 1 and 3, there is only a small difference between the mean times of the test courses and those of the original environments.

Table 1: *Experimental results of test courses vs. original environments*

	Experiment 1			Experiment 2			Experiment 3		
	Pass	Fail	t / [s]	Pass	Fail	t / [s]	Pass	Fail	t / [s]
Test course	22	0	31.83	22	0	42.53	22	0	34.68
Original	22	0	32.06	22	0	41.42	22	0	34.75

In comparison, there is a larger difference of the mean times in the second experiment, amounting to 1.1 s. The reason for this disparity is that in experiments 1 and 3, the local obstacle avoidance algorithm is not used because the AGV can navigate the given trajectory. In experiment 2, the alternative path planned in the test course is slightly different from the one in the original environment. In conclusion, the results show that experiments conducted in the test courses are reproducible and transferable to the original environment. Thus, test courses are generally suitable for testing navigation algorithms for AGVs. Special attention needs to be given to the local obstacle avoidance algorithms, since the experiments showed that the outcome of experiment 2 is the same (passed), but the difference in time indicates a slightly different behaviour.

5 Conclusion and Outlook

The paper introduces the addition of a triple intersection and obstacles to the standard F3244-17, adjusts its method for testing the navigation algorithms, and furthermore proposes a method for test course creation based on a given layout. The additional 2DA-shapes, the adjusted method and the TCC are validated using a fictive intralogistics scenario. The simulation setup using SIL embeds the control software of the AGV into a Unity3D simulation of the environment, including a virtual AGV and dynamic obstacles. The proposed method is used in the fictive intralogistics scenario to deduce test courses of the original environment. They are used to conduct experiments showing that the results achieved within the test courses are transferable to the original environment. Therefore, the extensions to F3244-17 and the proposed method to create test courses can omit time-consuming modelling of the original environment in future SIL studies. Furthermore, this indicates that tests in a physically built test course could be transferred to the target use case. Thus, the majority of tests could be conducted before commissioning the AGV at the customer's site. In future investigations the test method shall be further adjusted to include a comparison of the AGV's driven trajectory, thus enabling further comparisons of local obstacle avoidance behaviour. Moreover, this paper indicates that navigation algorithm tests within real test courses are transferable to the original environment. Therefore, this hypothesis shall be researched and verified.

Acknowledgement

The research project “FOLLOWme ILS – Intralogistics System using Automated Guided Vehicles” is funded by the German Federal Ministry of Education and Research.

References

Allmacher, C.; Dudczig, M.; Klimant, P.; Putz, M.: Virtual Prototyping Technologies Enabling Resource-Efficient and Human-Centered Product Development. Procedia Manufacturing 21 (2018), pp. 749–756.

Fitz, B.: Kopf an Kopf. Robotik und Produktion (2018) 5, pp. 66.

Bostelman, R.; Hong, T.; Cheok, G.: Navigation Performance Evaluation for Automatic Guided Vehicles (2016).

Echeverria, G.; Lassabe, N.; Degroote, A.; Lemaignan, S.: Modular open robots simulation engine: MORSE. In: 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 46–51.

F45 Committee ASTM F3244-17: Standard Test Method for Navigation: Defined Area. West Conshohocken, PA: ASTM International, 2017.

Fan, H.; Zhu, F.; Liu, C.; Zhang, L.; Zhuang, L.; Li, D.; Zhu, W.; Hu, J.; Li, H.; Kong, Q.: Baidu Apollo EM Motion Planner (2018).

Font, J.M.; Batlle, J.A.: Mobile Robot Localization. IFAC Proceedings Volumes 39 (2006) 15, pp. 340–345.

Fox, D.; Burgard, W.; Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine (1997) 1, pp. 23–33.

Koenig, N.; Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 2004, 2149-2154 vol.3.

Kohlbrecher, S.; Stryk, O. von; Meyer, J.; Klingauf, U.: A flexible and scalable SLAM system with full 3D motion estimation. In: IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2011, Kyoto, Japan, 11/1/2011 - 11/5/2011, pp. 155–160.

Li, G.; Yamashita, A.; Asama, H.; Tamura, Y.: An efficient improved artificial potential field based regression search method for robot path planning. In: International Conference on Mechatronics and Automation (ICMA), 2012, Chengdu, China, 8/5/2012 - 8/8/2012, pp. 1227–1232.

Makris, S.; Michalos, G.; Chryssolouris, G.: Virtual Commissioning of an Assembly Cell with Cooperating Robots. Advances in Decision Sciences (2012).

Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B.: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In: Eighteenth National Conference on Artificial Intelligence, 2002, pp. 593–598.

Paulo Costa, A.; José Gonçalves, B.; José Lima, B.; Paulo Malheiros, A.: SimTwo Realistic Simulator. Theory and Applications of Mathematics & Computer Science (2011), pp. 17–33.

Uttendorf, S.; Eilert, B.; Overmeyer, L.: A fuzzy logic expert system for the automated generation of roadmaps for automated guided vehicle systems. 2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bali 2016, pp. 977–981.