Balancing Centralised Control with Vehicle Autonomy in AGV Systems for Industrial Acceptance

Markus Bader

Vienna University of Technology
Institute of Computer Aided Automation (ACIN)
Vienna 1040, Austria
Email: `markus.bader@tuwien.ac.at`

Andreas Richtsfeld, Wolfgang Holl

DS AUTOMOTION GmbH
Technology & Product Development
Linz 4030, Austria
Email: `[a.richtsfeld, w.holl]@ds-automotion.com`

Markus Suchi, George Todoran, Markus Vincze

Vienna University of Technology
Automation and Control Institute (ACIN)
Vienna 1040, Austria
Email: `[markus.suchi, george.todoran, markus.vincze]@tuwien.ac.at`

***Abstract*—Automated Guided Vehicle (AGV) systems have to fulfil safety requirements and work reliably in order to be cost-effective and gain industry acceptance. Consumers want flexible AGV systems which require autonomous and distributed components to work, but this autonomy is often perceived as a disadvantage and a safety hazard. This work presents ongoing attempts and challenges to the distribution of knowledge and autonomy within AGV fleets while still ensuring safety and efficiency. Acceptance is gained by the integration of expert knowledge and a smoothly adjustable level of AGV autonomy which allows for a balance between centralized control and vehicle autonomy. Results are shown using a 3D physics simulation of a small production site.**

Keywords–AGV; Robotic; Industry; Safety; Planning; Autonomous; Navigation.

I. INTRODUCTION

Automated guided vehicles (AGV) are driverless mobile platforms primarily used for transportation processes as well as for flexible system solutions on assembly lines. Applications for AGV systems span from automated harbours where containers are moved around to pallet transport in warehouses. Hospitals use AGVs to automate processes, such as laundry and preparation of medication and to transport food and other goods between stations. The workspaces of humans and AGVs are normally separate but accessible to one another.

Prevention of collisions and deadlocks is imperative, and regular tasks, such as recharging or vehicle cleaning, must be managed. Reliability and safety are important issues, therefore simple and straight forward approaches are preferable. Thus, AGV systems are mainly designed off-line, with manually designed tracks, sometimes only for one specific vehicle type to make on-board planning obsolete. This is done to simplify centralised coordination and to enable an efficient overall control process.

Most AGV systems are individually designed for a certain application, which generates a market for highly specialized companies. Kiva Systems [1], with its AGVs for warehouse automation, is one of the most well-known companies. DS-Automotion [2], the project's partner, produces AGV systems with similar technology, but in contrast to Kiva Systems, upgrades a variety of vehicle types, ranging from small self-made platforms for the automotive or health-care industries to standard transportation products, such as forklifts. In addition, logistic solutions are provided for health-care, paper, and automotive industries, as well as for intra-logistic applications. Figure 1 shows some of these AGVs.

Figure 1. DS-Automotion's AGVs in action on an automotive assembly line, in a hospital and a paper factory.

In the last few years, customers have been increasingly requesting flexible and customisable solutions. They want systems to operate in environments with humans and they do not want to reconstruct their (often leased) buildings in order to accommodate an AGV system.

Normally, AGVs are not autonomous agents. This means

ISBN: 978-1-61208-405-3

that all vehicles of a fleet are guided by a centralised system which supervises the overall transport process. The agent's autonomy is limited to safety actions to ensure a safe overall process. A more flexible solution would lead to more data to process and higher computational costs. This additional data would not be manageable in real-time by a centralised system, as the bandwidth and the computational costs would be too high. As a result, control must be distributed and agents have to gain more autonomy in making decisions. However, autonomous agents are not well accepted in industrial applications and therefore a balance has to be struck among demands, flexibility and control. This paper targets exactly this problem by proposing a hybrid system which is able to scale the level of autonomy for each vehicle on demand and integrate expert knowledge into the system.

Section II describes a typical AGV system and the state of the art in mobile robotics. Our approach is presented in Section III and the challenges to face in Section IV. Results are shown in Section V, followed by a conclusion.

II. STATE OF THE ART

The structure of a classical AGV system is depicted in Figure 2 and works in the following way: The *AGV control system* (ACS) is driven by requests from the *Production Planning and Control* (PPC) module which disassembles general processes into internal processes. General processes are externally triggered processes such as customer requests, in contrast to internal processes, which describe the processes needed to fulfil externally triggered processes. Operation orders for AGVs are therefore part of internal processes and must be coordinated. The ACS assigns operation orders to the vehicles, specifies the track the vehicles have to follow and controls their speed in order to avoid collisions and deadlocks.

A. Industry

The automation industry prefers straightforward and non-complex solutions. For example, magnetic or RFID markers under the real, physical predefined track are commonly used for localization as well as for path-planning. The agent's on-board tracking control has to simply follow the *bread crumbs* of marker beacons. Such a control typically takes advantage of a flat system output [3], which in this case is the robot pose performing the tracking control. The benefit of such *bread crumb* localization is the low computational costs needed for localization and for trajectory planning. This type of navigation is sufficient for many industrial applications. Expensive safety-certified sensors and controllers are required, if there are humans in the same workspace. In this case, safety controllers must be used to override the motor controller commands in order to prevent accidents. SICK [4] produces certified laser range scanners which are able to dynamically adapt the safety areas to the vehicle's velocity. Obstacles detected within a safety area cause an emergency halt. An emergency halt means that an agent has to move itself into a safe state and cannot just stop moving, e.g.,, the system has to prevent agents from stopping in front of an emergency exit. Laser range sensors are thus mounted on AGVs in order to detect obstacles. However, the lasers are not necessarily used for navigation because of the additional complexity required. This forces every AGV to stay on the predefined tracks, therefore leaving a track in the case of an obstacle is not possible. An obstacle on the track will cause the AGV to slow down and eventually to stop. Even if localization techniques are able to deal with deviation from predefined tracks, they are usually avoided in order to keep systems simple.

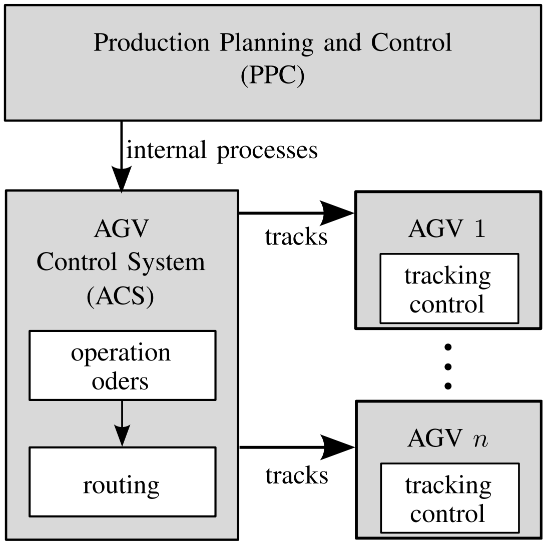


Figure 2. Modules of a classical AGV-system. A single server routes all AGVs along offline defined tracks; no path planning is involved.

DS-Automotion controls its AGV fleet by dividing the tracks into segments of arcs and lines. The control system distributes to each robot the next course segments to follow. This enables the ACS to prevent collisions by exclusively assigning only one segment at a time per agent. Due to the complexity of this problem, the time frame is limited and heuristics need to be used in order to reduce computational complexity. The goal of the approach proposed here is to decouple routing and local planning. The AGV should be able to recognize specific scenarios and should deal with them locally by adapting its trajectory planning strategy accordingly.

B. Research

The IEEE Robotics & Automation Magazine, Special Issues March 2014, recently summarised the state of the art and research done on perception and navigation for autonomous vehicles with articles on dynamic environments [5], risk analysis [6], self-localization and mapping in in- and outdoor environments [7], object recognition as well as path-planning [8] and motion-planning [9]. All of these research topics have to be combined to create an autonomous vehicle fleet. Projects, such as the DARPA [10] challenges have successfully demonstrated this, but the commercial market still lacks reliable autonomous agents.

Since 2014, Robot Operating System (ROS) has offered a software package dealing with AGVs [11]. The code collection includes drivers and simulations for an Ackermann type robot intended for logistics transport. The framework enables users to define tracks using waypoints, and the simulated AGV is able to follow these tracks. The ROS navigation stacks [12] are used to control and localise the vehicle. A logistic framework to coordinate multiple AGVs is missing, and the system is not able to deal with expert knowledge. However, we believe that this expert knowledge is vital for the commercial market and for industrial acceptance.

Similar set-ups to AGV systems can be found in RoboCup [13]. Competitions like RoboCup's Small Size League (SSL) soccer is designed to improve multi-agent cooperation through friendly competition.

The environment has similarities to an AGV system. Agents are controlled by a centralised server, and all objects on the playing field are tracked by a standardised vision system. The league has shown that it is possible to detect specific scenarios and to react quickly by adapting *plays* [14]. A play denotes a sequence of actions or behaviours according to a playbook, e.g., follow track and slow down. The playbook describes recognizable scenarios with according plays, e.g., a scenario (with an automated fork lift) in front of an elevator door ⇒ play: verify that the fork is folded before entering the lift; follow track precisely with low speed. Plays can also provide predefined plans with roles for multiple agents which can be adapted to scenarios to prevent deadlocks or collisions, for example in the following scenario: the passing of two agents ⇒ play: select leader; leader selects side for passing; follower acknowledges side; passing. Similar techniques are also used in RoboCup's Middle Size League (MSL), which has no centralised command system.

In the approach proposed here we are presenting an idea for how to integrate expert knowledge into the system to support play selection. This is done by augmenting track segments as well as areas around segments in order to simplify scenario recognition and to enable reproducible behaviour.

III. APPROACH

AGV systems currently deployed in industrial applications use manually offline designed tracks for path planning. These tracks are defined by a list of segments and distributed by the ACS to the AGVs, as shown in Figure 2. An AGVs task is to follow these segments. This system has only two planning levels:

- the overall routing on the centralised server and
- the on-board tracking control on the AGV.

Obstacles on the track always trigger an emergency halt.
We would like to present an approach which enables an AGV system to additionally:

- autonomously avoid obstacles on the track,
- solve situations without the ACS interfering, e.g., a multi-robot situation or pick and place actions,
- use optimised trajectories to be time-, energy- and/or resource-optimal (e.g., floor abrasion), and
- be easier to maintain and less expensive during system design and set-up.

This can only be realized if AGVs are able to:

- localise themselves, (even when leaving the predefined track),
- communicate with each other, and
- execute and adapt their behaviour (role play), to solve local issues without centralised intervention.

We propose that the ACS distribute segments to the AGVs, similar to before, but encapsulated with additional attributes. For demonstration purposes we will group areas into *free* or *critical*. A free area signals that an AGV is allowed to leave the track; a critical area indicates particular precaution. The additional segment attributes are used to indicate to the system what to expect or which behaviour (role play) should be selected to manage the track segment. A typical attribute would be that no stops are allowed. This is important when passing a fire door: the agent would have to verify whether there is enough space free after the critical section before entering said section and would select an appropriate motion control algorithm. In our approach, agents are able to select one of two motion control algorithm.

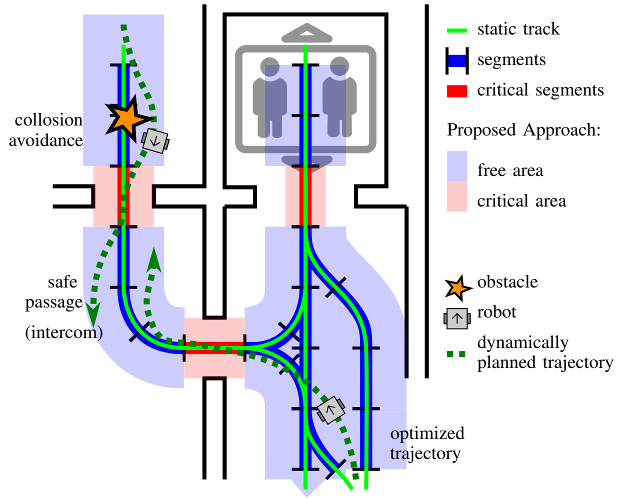


Figure 3. This figure depicts the limitations of the a classical AGV system and the advantages to be gained by the new approach proposed.

- A *Model Predictive Control* (MPC), implemented similarly but in a more advanced way than the *Dynamic Window Approach* (DWA) [15] to follow tracks, which allows the system to diverge from the track and to avoid obstacles.
- A tracking controller based on a flat system output [3], which tries to follow tracks precisely. In the presence of an obstacle, the control slows the vehicle down, eventually stopping it.

Figure 3 shows the limitations of the old approach and the advantages of the new approach: The system currently used has centralised path planning based on predefined line and arc segments (blue). An AGV has to follow the static tracks (green) routed by the control system. In the face of an obstacle, the AGV slows down and eventually stops on the track.
In contrast to the system currently used, the system proposed here uses predefined areas in which a vehicle is allowed to move freely. Obstacles can be circumnavigated and two or more vehicles are able to directly communicate in order to plan trajectories for safely passing each other. Trajectories are locally planned and can be time-, resource- or energy-optimised.

A. First Iteration – Behaviour Controller, Role Play and Playbook

The aforementioned concepts *play/role play* and *playbook* describe strategies for facing specific scenarios. A play or role play describes the parameter selection, and an interaction procedure for one or more robots for a certain length of time. The playbook holds multiple such predefined role plays for selection. The behaviour controller implemented is in charge of recognising scenarios and selecting appropriate plays. In the first iteration, simple role plays are implemented with the goal of getting a system up and running as it was before. AGVs use the predefined tracks as a basis for local path planning, but they may change their local path when an obstacle is blocking it or when indicated to by predefined segments. Agents are able to select between two tracking control types for different motion behaviours: MPC or low level tracking control. The controller parameters are selected on-demand depending on the role play executed. This enables the vehicle to behave differently in different areas, while also giving the operator the capability of restricting the system, when necessary.

B. Second Iteration – Robot-Robot interaction

The second stage will enable vehicles to plan their own paths, if permitted within the current area. Using the aforementioned playbooks with role plays for specific scenarios enables the control system to detect such scenarios and to initiate role plays with one or more agents involved, e.g., the passing of two vehicles in a hallway or passing a door. If such a scenario is recognised, the vehicles involved are allowed to communicate with each other to adapt the known role play. This allocates the control competences to the agents, thus making the system more flexible.

C. System Components

The overall architecture of the system proposed is shown in Figure 2 and in more detail in Figure 4. A PPC module coordinates the overall process and interfaces the company's accounting system, e.g., an *Enterprise Resource Planning* (ERP) system. The ACS gives transportation orders to the vehicles *(job planner)* and plans optimal routes for each vehicle *(route planner)*. Each AGV implements a *Behaviour Controller* (BC) as a state machine, which makes binary decisions for them and selects role plays. The BC module controls the AGV while autonomously solving situations based on the aforementioned playbook and communicates success or failure to the ACS. It triggers local navigation modules if a new plan needs to be computed.

IV. CHALLENGES

In our approach, we introduce two fundamental changes to the AGV system currently used by DS-Automotion. Each vehicle has a navigation module with a path-planner and a motion controller (*aka* a trajectory-generator), as well as a behaviour controller (BC) to trigger role plays.

A. Navigation

Two navigation layers in the AGV are used in the system proposed, namely *path planning* and *motion control*, which are often denoted as global and local planning, respectively. This may cause confusion because the centralised ACS also has a planning module which computes the overall *AGV routing tables* for the fleet.

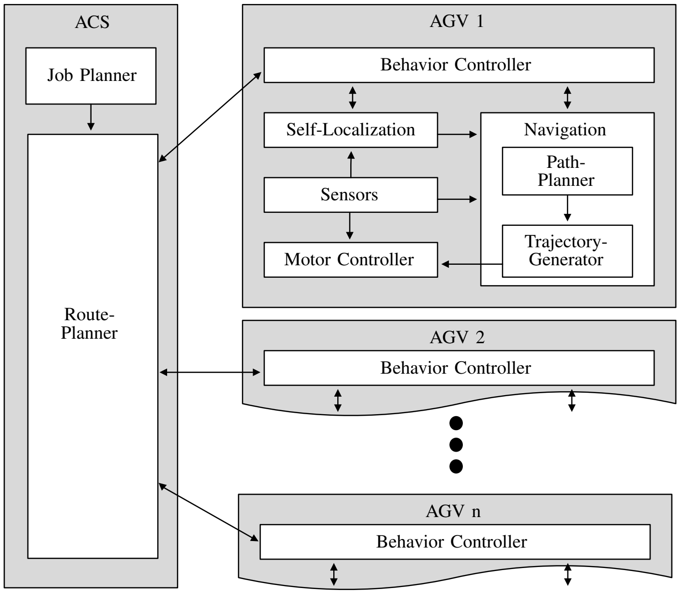


Figure 4. AGV system overview: The ACS gets orders from the PPC (see Figure 2) and distributes them to the AGVs. The ACS also supervises AGV route planning in order to optimise the execution time of all of the orders given to the system.

1) AGV routing tables: In order to take full advantage of the vehicles' capabilities, the ACS has to be adapted. The route planning has to cope with variances in execution time, as the execution time of a role play can vary due to local navigation. The ACS has to learn and adapt parameters such as execution time and success rates of role plays in order to create optimal routing tables.

2) Path Planning: The local path-planning receives routes to follow from the ACS and delivers segments to follow to the motion control. A challenge at this layer arises if a vehicle leaves the predefined tracks. The path-planner has to compute a suitable path by using known maps of the environment or to communicate an issue to the ACS. It is also the task of this planner to find paths to objects for pickup.

3) Motion Control: Virtually exact tracking control can be achieved by using, a flat output system model and track segments as splines as input , but the system proposed should be able to diverge from the track if needed. An MPC [9] generates a possible trajectory based on the current system state and weights each one based on a cost function which can include the obstacles detected. This control executes the first control sequence of the winning trajectory only until the next control iteration. Continuous updates are needed for safe and smooth motion control. It is commonly known that the most computationally intensive procedures in this cost function are collision detection and the evaluation of motion costs, but the latest research has demonstrated that the introduction of proper heuristics effects a huge performance gain [16], [17]. In our approach, we allow the system to switch between an exact tracking control and the MPC.

B. Behaviour Controller

Playbooks as used in RoboCup soccer scenarios must be developed to simplify plans, especially when multiple agents are involved. The selection of a lead agent during a multi-agent role play must be managed [18]. However, more important for the acceptance of the system are the integration of expert knowledge and reproducible behaviours.

C. Self-Localization

Another challenge occurs due to the inaccuracy of self-localization when using laser-based localization methods, such as *Adaptive Monte Carlo Localization* (AMCL) [19], which is implemented in the ROS or in the *Mobile Robot Programming Toolkit* (MRPT) [20]. The system has to deal with inaccuracies, and eventually has to adapt its behaviour to gain a better localization confidence when needed. For example, a pick and place procedure where one vehicle places a payload and another vehicle picks it up fails upon inaccurate localization.

D. Mapping

In order to be cost-effective, AGV systems with customised vehicles are usually deployed for a long period of time. During this long period of use, changes to the environment can be expected and must be dealt with. A common map layer which represents daily changes to the environment can be updated and distributed to the vehicles. Creating a sound map of multiple measurements is a difficult challenge and it is not yet clear whether this task should be performed by each agent individually or by a centralized unit, especially if loop closing is necessary.

E. Industrial Acceptance

Industry demands flexible and easily maintainable solutions. This is only manageable with a distributed system, but that inherently increases the system's complexity. It will be a challenge to find the right balance and in this study, suitable role plays with expert knowledge in order to create an acceptable system for industry.

V. RESULTS

We are currently at the first iteration level, as proposed in Section III with a simple working set-up.

A. Set-Up

We interfaced the ACS used by DS-Automotion to intercept operation orders and computed routing tables. The ACS has multiple safety features to ensure a safe process. For example, all vehicles are monitored to verify that vehicles are on the track following the assigned route. A simulated environment was created using GazeboSim [22], a freely available 3D simulation package including a physics engine. We simulated a production site within our lab with multiple vehicles. Figure 5 shows two related snapshots. The MRPT-library is used for localization and ShmFw [23] for communication and visualization of data. ShmFw is a fast dynamic framework based on the boost inter-process library [24], which uses shared memory elements for inter-process communication. ROS libraries are only used to interface the simulator by using customized ROS nodes to exchange data between ROS messages and ShmFw variables. The decision to avoid ROS in the functional code is due to the current system used by the project partner, who uses their own middleware. Another reason to exclude ROS was down and upward compatibility. Compared to the product cycles of AGVs, the release cycles of ROS are very short, and usually a company has to support products for many years. However, there is an industrial version called ROS-Industrial [25], which might be of interest for future projects.

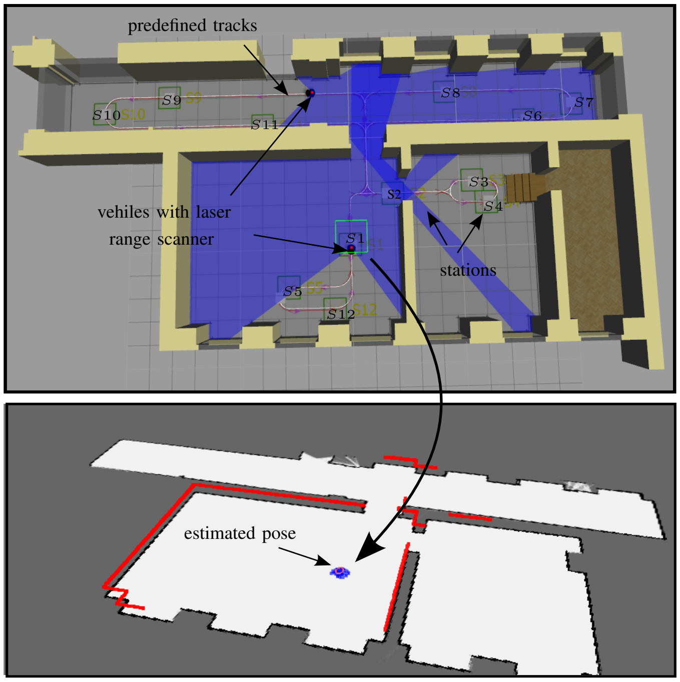


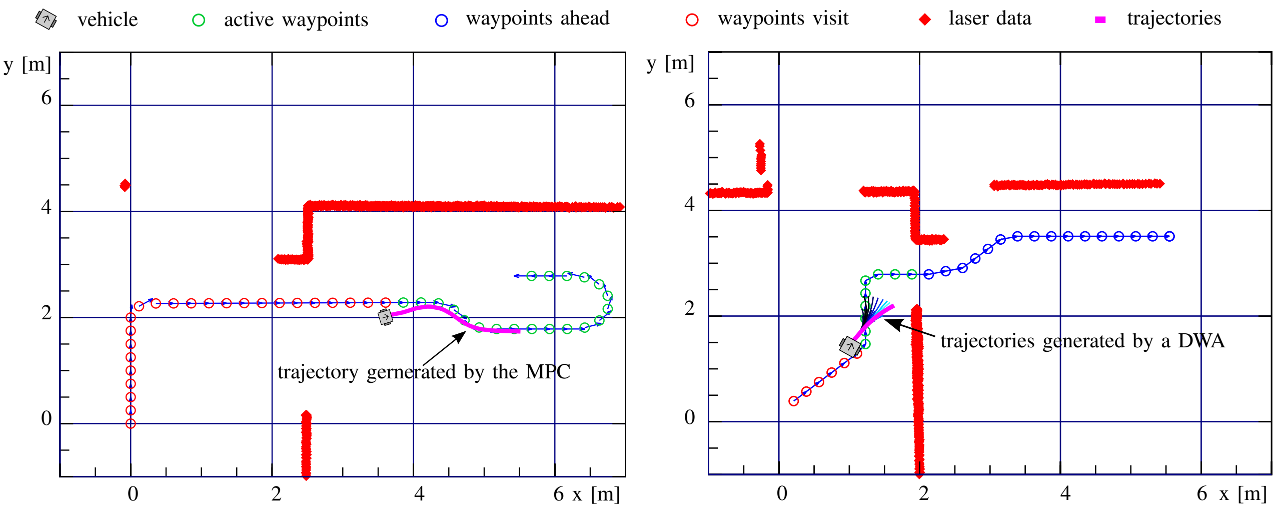
Figure 5. Top: Simulated production site with tracks, stations $S1-S12$ and two simulated vehicles with a SICK laser range scanner and rays in blue. Bottom: MRPT particle filter self-localization with an estimated robot pose on a previously-generated map using a Rao-Blackwellised particle filter SLAM, also implemented in MPRT.

B. Autonomy

At the current level we are able to start vehicles at arbitrary locations. The system uses a local path planner to find a path to the next known track to receive orders. The aforementioned complex initialisation procedure is still needed because of safety issues. All segments delivered to vehicles are augmented with additional parameters to trigger different behaviours, such as switching between the multiple implemented motion control methods with various settings. The operator is now able to predefine areas to control vehicle behaviour in advance. For example, vehicles in open areas use an MPC or DWA to cope with blocked tracks. In areas such as turns between $S3-S4$, shown in Figure 5, which are close to the stairways, the operator is able to predefine a motion control to follow the track as precisely as possible. Figure 6 shows cases with unblocked and blocked paths, as well as different tracking controls implemented. The behaviour shown in Figure 6c allows the vehicle to select trajectories to avoid collisions with obstacles next to the path, but an obstacle on the path would cause the vehicle stop. This behaviour was designed to increase the acceptance of the system and was favoured by the industrial project partner.

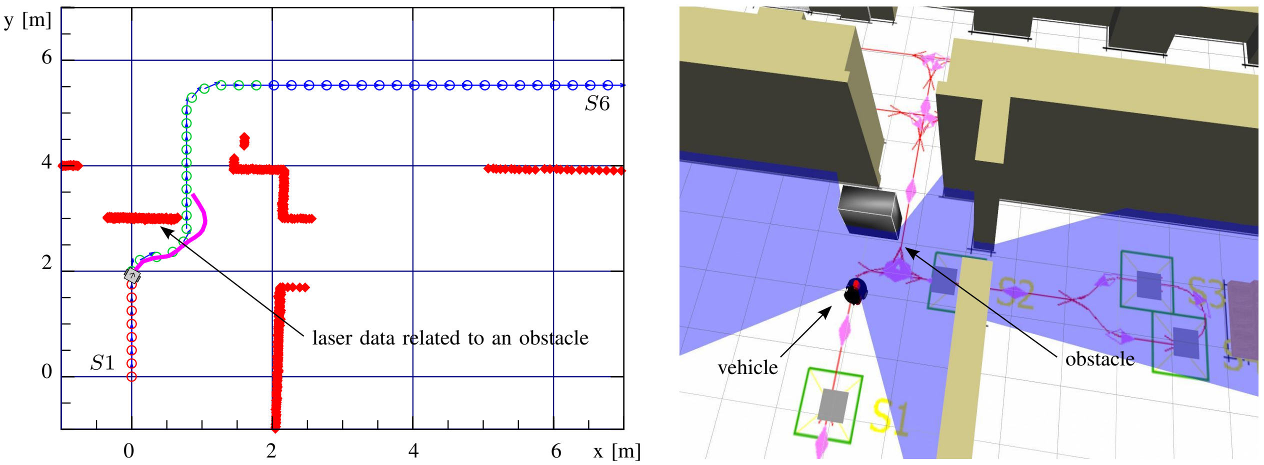
C. Path-planning

The waypoints shown in Figure 6a and 6c are based on the static predefined segments and placed at a constant distance to represent the path, in contrast to Figure 6b. In this case the AGV computed its own track by taking the previously



(a) Predefined waypoints between $S1$ and $S3$. A MPC is used to follow the path.

(b) The waypoints between $S1$ and $S3$ are computed on-line using an A-star. A DWA is used to follow the path.



(c) The MPC's cost function includes sensor readings and avoids collisions with an obstacle next to the path.

(d) Simulated environment related to Figure 6c. The obstacle next to the path would cause a collision.

Figure 6. Operation orders executed with different behaviours.

recorded map and the start and goal position into account. The trigger for doing so is based on the expert knowledge encoded in the track segments delivered by the ACS. The route is not as smooth as the one delivered by the ACS, but in this case the DWA implemented takes care of this problem and avoids obstacles by using a cost function to weight possible trajectories within a certain time window, as shown in Figure 6b. In this way the ACS does not have to take care of replanning until the AGV signals otherwise. The current system is now able to perform at the same level of efficiency as before but is also able to cope with obstacles.

VI. CONCLUSION

This work presents a recently-begun research project with the goal of transferring research knowledge from the field of mobile robotics to the industrial application of AGV systems. We proposed an approach for decentralisation of the control system in order to achieve a flexible solution. The approach entails enhancing agents with an on-board self-localization and navigation module as well as a behaviour controller for carrying out autonomous actions. The centralised control system has to deal now with autonomous agents, shifting the task from control of them to coordination of them. Expert knowledge augments, on the one hand, the map to allow or confine autonomous actions in specific areas, and on the other hand, the route delivered to the AGV to prepare the agent for scenarios. We believe that only a balanced system tuned by the human operator on site will be accepted in industrial applications, and reproducible behaviour, as well as human ability to influence autonomous behaviour are vital to this acceptance.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Austrian Research Promotion Agency (FFG) according to grant agreement No. 846094 (Integrierte Gesamtplanung

mit verteilen Kompetenzen für fahrerlose Transportfahrzeuge).

REFERENCES

[1] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses," in Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence - Volume 2, ser. IAAI'07. AAAI Press, 2007, pp. 1752–1759.

[2] "DS-Automotion," 2015, URL: http://www.ds-automotion.com [accessed: 2015-04-04].

[3] M. Liess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," International Journal of Control, vol. 61, no. 6, 1995, pp. 1327–1361.

[4] Detection and Ranging Solutions, SICK AG, Erwin-Sick-Str. 1, 79183 Waldkirch, Germany, December 2011, 8014402/2011-12-20. [Online]. Available: http://www.sick.com

[5] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized centroid and formation control for multi-robot systems," in Robotics and Automation (ICRA), 2013 IEEE International Conference on, May 2013, pp. 3511–3516.

[6] C. Laugier et al., "Probabilistic Analysis of Dynamic Scenes and Collision Risks Assessment to Improve Driving Safety," Intelligent Transportation Systems Magazine, IEEE, vol. 3, no. 4, Winter 2011, pp. 4–19.

[7] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," in Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, vol. 1, Sept 2003, pp. 842–849.

[8] S. M. LaValle, Planning Algorithms. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[9] T. Howard, M. Pivtoraiko, R. Knepper, and A. Kelly, "Model-Predictive Motion Planning: Several Key Developments for Autonomous Mobile Robots," Robotics Automation Magazine, IEEE, vol. 21, no. 1, March 2014, pp. 64–73.

[10] S. Thrun et al., "Stanley: The robot that won the DARPA Grand Challenge," Journal of Field Robotics, vol. 23, no. 9, 2006, pp. 661–692.

[11] "ROS: The agvs package," 2015, URL: http://wiki.ros.org/agvs [accessed: 2015-04-04].

[12] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The Office Marathon: Robust navigation in an indoor office environment," in Robotics and Automation (ICRA), 2010 IEEE International Conference on, May 2010, pp. 300–307.

[13] H. Kitano, M. Asada, I. Noda, and H. Matsubara, "RoboCup: robot world cup," Robotics Automation Magazine, IEEE, vol. 5, no. 3, Sep 1998, pp. 30–36.

[14] J. Bruce, M. Bowling, B. Browning, and M. Veloso, "Multi-robot team response to a multi-robot opponent team," in Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, vol. 2, Sept 2003, pp. 2281–2286.

[15] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," Robotics Automation Magazine, IEEE, vol. 4, no. 1, Mar 1997, pp. 23–33.

[16] R. Knepper and A. Kelly, "High Performance State Lattice Planning Using Heuristic Look-Up Tables," in Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, Oct 2006, pp. 3375–3380.

[17] M. Suchi, M. Bader, and M. Vincze, "Meta-Heuristic search strategies for Local Path-Planning to find collision free trajectories," in Proceedings of the Austrian Robotics Workshop (ARW-14), May 2014, pp. 36–41.

[18] N. Basilico, N. Gatti, and F. Amigoni, "Leader-follower Strategies for Robotic Patrolling in Environments with Arbitrary Topologies," ser. AAMAS '09. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 57–64.

[19] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press, 2005.

[20] "Mobile Robot Programming Toolkit (MRPT)," 2015, URL: http://www.mrpt.org [accessed: 2015-04-04].

[21] "RISC Software GmbH," 2015, URL: http://www.risc-software.at [accessed: 2015-04-04].

[22] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol. 3, Sept 2004, pp. 2149–2154.

[23] "ShmFw," 2015, URL: https://github.com/ShmFw/shmfw [accessed: 2015-04-04].

[24] "Boost.Interprocess," 2015, URL: http://www.boost.org/doc/libs/1_55_0/doc/html/interprocess.html [accessed: 2015-04-04].

[25] "ROS-Industrial," 2015, URL: http://http://rosindustrial.org [accessed: 2015-04-04].