Modular Platform for Route Guidance in the Cyber-Physical Laboratory Test Field

Marian Göllner*, Sven Jacobitz, Taihao Li, Xiaobo Liu-Henke

Institute of Mechatronics, Ostfalia University of Applied Sciences,
Salzdahlumer Str. 46/48, D-38302 Wolfenbüttel; **mar.goellner@ostfalia.de*

SNE 33(1), 2023, 45-52, DOI: 10.11128/sne.33.tn.10637
Selected ASIM SST 2022 Postconf. Publication:2023-02-01;
Received Revised Improved:2023-02-27; Accepted:2023-03-10
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper presents a domain-specific configurable, modular platform for route guidance and trajectory planning (doplar) of intelligent vehicles in different cyber-physical traffic systems, which can be used in projects of different domains. Due to its modular structure, this platform can be used both in road traffic and in the context of Industry 4.0 or Smart Home applications. The big advantage of this platform is that core modules such as the route guidance can be retained and only individual modules, e.g. for wireless communication, must be adjusted. The goal of the entire platform is to plan an optimized vehicle operation according travel time and energy consumption, incorporating dynamic environmental data available from wireless communication within the cyber-physical transport system.

Introduction

A key characteristic of a cyber-physical system is the blurring of boundaries between mechatronic components that communicate via a network infrastructure such as the internet. This steadily increasing degree of networking and functional diversity is leading to increasingly complex, distributed systems with more and more intelligent functions. Such systems are characterized by the integration of a wide variety of components with different requirements for real-time capability, safety and timing, which must all interact reliably within the framework of the CPS. Accordingly, modeling, synthesis as well as validation are complex. Real-time realization and testing in particular requires a lot of effort due to the lack of a real environment and real communication partners [1].

An intelligent vehicle must be capable of performing the four basic tasks of measuring, recognition, planning and execution in order to achieve its intrinsically set or extrinsically motivated goal. First of all, it must detect (measure) its environment with sensors and process (recognize) the sensor information for environment perception. Based on that, the autonomous system makes decisions about its behavior and its interaction with the environment (planning) and finally realizes these decisions with the help of its actuators to execute the basic movement.

Motivation

At Ostfalia University, several transdisciplinary joint projects are being carried out in which cyber-physical systems are investigated in various domains, such as road traffic, Industry 4.0 or Smart Home.

This paper presents the domain-specific configurable, modular platform for route guidance and trajectory planing (doplar) of automated transportation systems in cyber-physical traffic system, which can be applied to every of these above introduced and further projects in order to develop reuseable functions. The aim of this platform is the planning of an optimized vehicle operation with regard to travel time and energy consumption, including dynamic environmental data from wireless communication within the cyber-physical transport system. The paper will detail on the route guidance and its exemplary application for a cyber-physical laboratory test field for intelligent mobility applications [2] developed at Ostfalia.

Methodology

It is obvious that the development of cyber-physical systems in general or of the platform doplar requires a clearly structured, methodical approach due to the complex and interconnected individual functions. In order to master the overall complexity and the interdisciplinary research and development process in the fields of mechanics, electronics as well as control, information and communication technology, a systematic structuring of the entire mechatronic system is necessary.

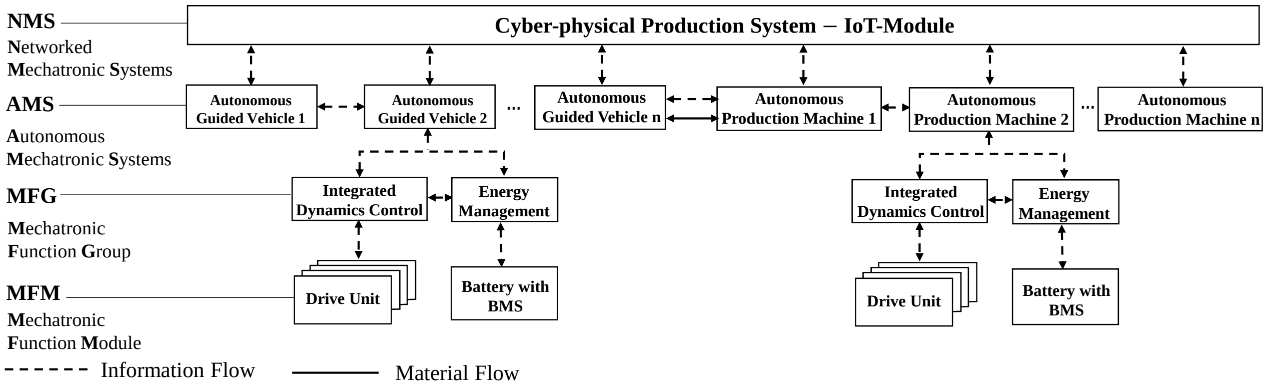

Fig. 1: Mechatronic structuring and hierarchization of a cyber-physical industry 4.0 production plant

With the help of mechatronic structuring according to [1], the entire system is structured hierarchically in a top-down process starting from the main function and subdivided into partial and/or sub-functions. The individual functions are encapsulated in modules with defined interfaces, which can be combined into groups to fulfill higher functionalities. This mechatronic structuring is carried out on four levels:

- Mechatronic Function Module (MFM): The lowest hierarchical level is made up of the MFM, consisting of mechatronic systems that cannot be further divided, including the mechanical support structure, sensors, actuators, bus communication and information processing. Each encapsulated MFM has a defined functionality and describes the dynamics of the system.
- Mechatronic Function Group (MFG): MFGs are created by coupling several MFMs and adding a higher-level information processing. MFGs enable the implementation of more sophisticated functions by using the subordinate MFMs with their actuators.
- Autonomous Mechatronic System (AMS): The overall mechatronic system forms the next hierarchical level of the AMS by combining several MFGs. After processing the available information, set values for subordinate MFGs and MFMs are generated.
- Networked Mechatronic System (NMS): If several AMSs are operated side by side, e.g. to process a customer order, a higher level of coordination is required. This coupling with information technology at the highest level is a NMS, or in this case a cyber-physical production system, which corresponds to an industrial 4.0 production line or an industrial 4.0 factory. The product information is managed by the high-level cyber-physical production system and forwarded to all relevant components of the production line, e.g. via WLAN.

Figure 1 exemplary illustrates the modular and hierarchical structure of an autonomous industrial 4.0 production plant consisting of several AGVs and production machines as focussed in the project Synus, which is the basis of the pilot application focused in this article. By adding or exchanging MFM, MFG or AMS, the NMS can be configured as required.

At the lowest level for the AGVs there are four MFMs with smart drive units for the realization of the set forces and moments and one MFM for the power supply, which consists of the battery modules and the battery management, which, among other things, monitors the battery during operation and balances the charge. The integrated dynamics control for controlled vehicle dynamics and the energy management for controlling and monitoring the energy flows are arranged hierarchically above. AGVs and production machines as mechtronic complete systems are arranged hierarchically one level higher than AMS.

The highest level of the NMS is formed by the coupling with information technology of the AGVs and production machines in the production plant. Mechatronic structuring is followed by mechatronic composition. In a bottom-up process, each module, starting with the lowest hierarchy level of the MFM, is designed, validated and successively integrated into the higher-level overall system in a model-based, verification-oriented process using the defined interfaces.

1 State of the Art

The following is an overview of the state of the art with regard to all functions necessary for the route guidance of the doplar platform.

1.1 Mapping

The basis for the route guidance are maps, which can first be divided into geometric and topological maps as well as hybrid intermediate forms. Topological maps are based on graphs and only provide nodes and weightings without directly accessible links to the real world. These maps are used for route planning [3]. Geometric maps, on the other hand, reproduce the environment exactly by projecting it e.g. onto a two-dimensional surface (grid maps such as building ground plans, [4] or three-dimensionally with elevation data as terrain maps, [5]). A special form is created by assigning features to special contours on the map (feature maps, edges and walls, etc., [6], [7]). The result of the environment perception by the vehicle sensors are environment maps, whereby the form of representation depends very much on the sensors used and the output data of the fusion structure. The maps must be converted to topological.

1.2 Route planning

Conventional navigation functions are mainly based on topological maps and rarely directly on geometric maps or hybrid hybrids. Simple approaches for direct navigation on grid maps would be the Manhattan metric [8] or the more centralistic French railway metric [9]. More general are the navigation algorithms based on topological maps using graphs on which the methods of graph theory can be applied. Navigation in this context is only a problem of the shortest paths, if the nodes and weightings of the graph have been reasonably defined in advance. The exact formulation of the problem is crucial to answer the complexity question. If the graph is set without negative weights, the Dijkstra algorithm [10], which is also regarded as the basis of map navigation systems, offers the shortest runtime. If you limit the search field with e.g. the A* algorithm [11], you get even shorter runtimes, but at the expense of the reliable identification of the shortest path. Here it is possible that a shorter way is not found due to the restriction although it can be proven. If negative weights are set up, e.g. to favour certain paths or edges or to reward them for use, the Bellman-Ford algorithm [12] achieves the shortest runtime. These algorithms all start at a start point and propagate to the end point. As soon as this is reached, the algorithm usually ends, since this is the shortest path as the abort criterion.

This is different with the algorithms of [13] and [14], which are founded on the work of [15] and are each based on finding the shortest paths between all node pairs. The approach here is that if the route between any two points is to be retrived, the individual partial paths of this route are already minimal in themselves. If the shortest paths between the respective points are known, the shortest path is composed of the already known shortest paths of the partial paths. In the ideal case, even the shortest path between the searched start and end points is already included. These methods are particularly suitable for static problems, since an initial high calculation effort is necessary, but does not have to be performed a second time, whereas they are less suitable for dynamic applications in traffic.

2 Concept of Doplar

The aim of doplar is to plan optimized vehicle operation by route and trajectory planning with regard to journey duration and energy consumption, taking into account dynamic environmental data from wireless communication and vehicle and environmental sensors.

A problem with current approaches is the largely specific development of individual functions for certain vehicles in defined domains without taking into account a transfer to related domains, which results in a high, double development effort. This problem results in the essential requirement that the functions presented in this paper should be used for different domains without much effort. The functional structure proposed to achieve the goal and the solution of the problem, the domain-specific configurable modular platform for route guidance and trajectory planning of intelligent vehicles (doplar), is shown in Figure 2 .

In order to achieve the goal of optimized vehicle operation, a route guidance function is required to determine an optimal route, which is then planned out by the trajectory generation for realization. A human-machine interface (HMI) is required for route guidance, e.g. to enable vehicle occupants or other users to set targets. Optionally, fleet management could also intervene in route planning in order to coordinate several vehicles. For the route guidance, mapping is necessary, which provides map data in the correct form and updates it,

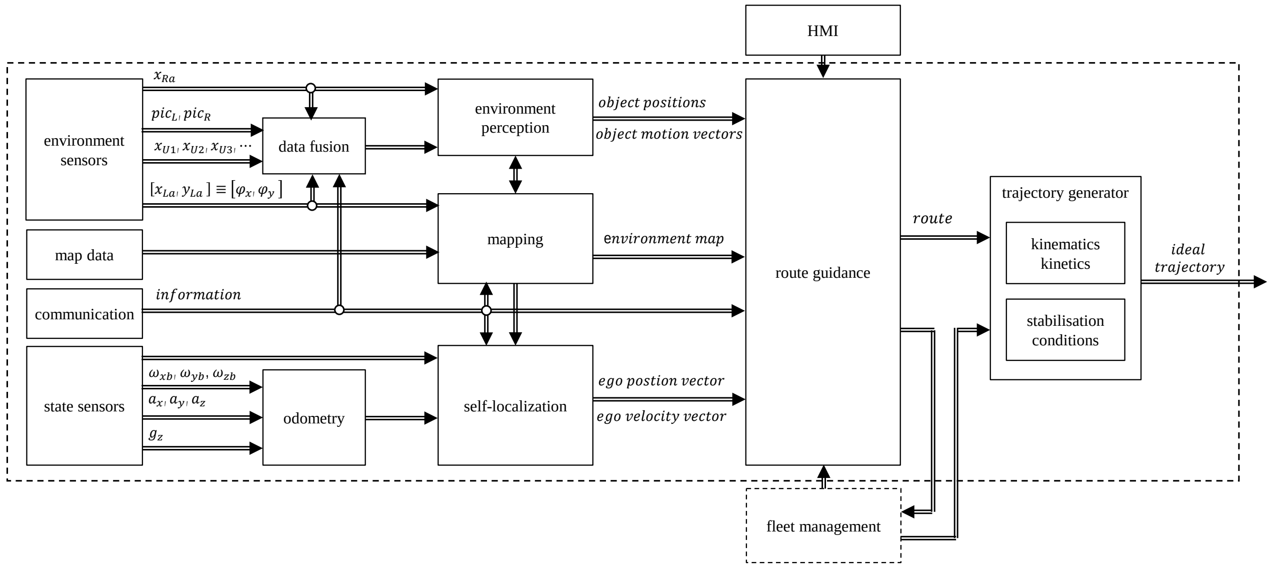

Fig. 2: Concept of the domain-specific configurable modular platform for route guidance and trajectory planning of intelligent vehicles (doplar)

self-localization to determine the current position, environment perception to determine the driving environment and possibly deviating map data and communication, which enables an exchange of information between the ego vehicle and the environment.

All these functions are encapsulated in different modules with defined interfaces, so that the exchange of individual modules is possible as long as the interfaces are maintained. This ensures that the platform can be used in different domains, since domain-specific modules, e.g. for environment perception, can be easily exchanged without having to change core algorithms such as route guidance.

3 Design of Doplar's Route Guidance

This chapter describes the design of the functions for providing map data and route guidance of doplar.

3.1 Mapping

The mapping module serves to provide and update the map material as a necessary basis for route guidance. The map data has to be described mathematically as graph $G = (N, E)$, which consists of a finite set N of nodes (x, y) and a finite set E of weighted edges. A node is a point with fixed coordinates in x and y position and an edge is the connection between two nodes. The edges thus correspond to road segments that are connected to each other by the nodes. The origin of the map data can also be domain-specific: For applications in road traffic, for example, the OpenStreetMap can be used, whereas for applications in Industry 4.0 plants or the Smart Home floor plans can be converted into graphs. A further possibility for generating or updating the map material is the use of a SLAM algorithm, which evaluates the vehicle's environment sensors and provides information about its environment.

The map forced in this paper is a geometrical (G) as well as a topological (T) hybrid form, a hybrid G+T map, which links a 2.5 dimensional grid map with a graph map and thus enables real-time, event-based online navigation. This map goes back to [16] and was adapted in the context of this work. The map is generated by splitting the fused sensor data in the vertical axis and projecting them onto a grid on the position of the vehicle's center of gravity. The following Figure 3 shows the projection on the grid map with a corresponding resolution. This map is now only insufficiently suitable to carry out the route guidance. Only an assignment of each individual grid cross as node and the execution of the Manhattan metric would be conceivable. Since this follows fixed paths, i.e. the grid, the vehicle would roll in x- and y- direction and possibly not find the shortest way, because nodes cannot be skipped.

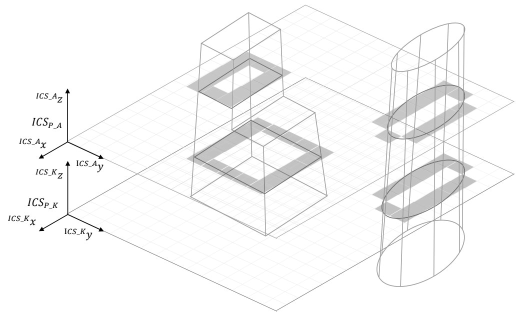

Fig. 3: Creating a 2.5-dimensional, geometric map from data of the environment perception

A logical improvement is the definition of a graph map based on the grid map by transforming the grids into nodes and connecting the nodes with weighted edges depending on the actual distances from each other. Thus simple diagonal journeys are possible. An optimal route, however, still cannot be found under the premise that the grid is evenly distributed, since the paths always run at a minimum of 45°.

Therefore, the approach chosen sorts only characteristic points to the graph map, namely those corners which describe the maximum extent of an object. The edges of the obstacles and objects are detected and pre-sorted according to relevance using the global target vector. The distance of these points can be determined by means of Euclidean distance and taken over as weighted edges between the nodes. Edges that are covered by an object get a very large weighting factor and are therefore disadvantaged by the navigation algorithm. The new map (Figure 4) can now be treated as a pure graph map and can be traced back to the grid sectors of the grid map by assigning the nodes.

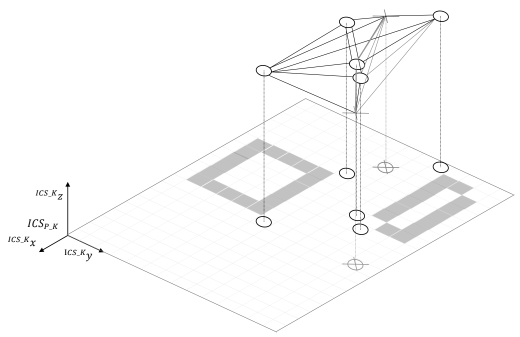

Fig. 4: Creating a topological graph map based on a 2.5-dimensional geometric map

3.2 Route guidance

The route guidance module used according to [17] aims at finding an optimal route from a start point to a destination point in a directed graph G, which describes the traffic network, related to a defined quality criterion. Route guidance is based on Dijkstra's algorithm, which belongs to the methods of width search. Using the cost function

$$J_i = g_s \cdot s_i + g_t \cdot t_i + g_E \cdot E_i \qquad (1)$$

and the weighting factors g_s, g_t and g_E, the costs J of each edge i are calculated from the information of the edge weights about distance s, duration t and energy consumption E between two nodes. Dijkstra‘s algorithm always converges to the optimal route if the graph does not contain loops or negative edge costs.

The total cost of each node is initialized to infinity with the start of the algorithm except for the start node, which is initialized with zero. The algorithm determines the costs of the unvisited neighboring nodes from the start node as the sum of its own costs plus the costs of the connecting edge and updates the total costs of the neighboring nodes if the recalculated costs are less than the previous costs. All considered neighbor nodes are added to a waiting list, the start node is marked as visited and the node with the lowest cost is selected from the waiting list. Starting from this node, the procedure described is repeated until the destination node is reached with minimal total costs (see Figure 5).

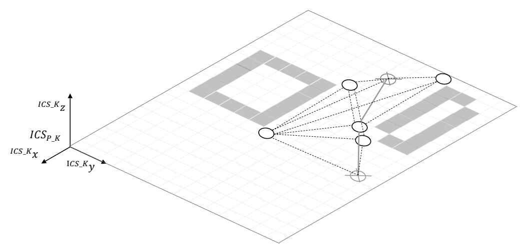

Fig. 5: Result of Dijkstra's algorithm for finding the optimal route in a topological graph map

The result of minimizing the total costs from start to destination is the route $\underline{r}$, which consists of nodes connected by edges:

$$\underline{r} = \min_{n \in N} \sum_{i=start}^{n=dest} J_i \qquad (2)$$

The integration of dynamic information from wireless communication in the cyber-physical traffic system is realized by a temporary adaptation of the edge weights in the graph. For this purpose, the position of the message is first used to determine which edge is to be assigned the information. The edge weights are then changed according to the type of information. If, for example, a closed road occurs as a result of road works, the distance, duration and energy consumption for the corresponding edge are set to infinity, so that the edge can no longer be part of the optimum route regardless of the weighting factors. If, on the other hand, a delay occurs due to congested traffic, only the duration and energy consumption of the edge are changed, since the length of the route does not change.

3.3 Trajectory planning

Trajectory planning is used to generate target polynomials for longitudinal and lateral dynamics, which are then passed on as set points to the dynamic control of an AGV in order to follow the optimum route determined in the route guidance module and execute the transport order. The return value from the route guidance is the already optimized group $\underline{r}$ of points (x_i, y_i) in order of the direction to be departed:

$$\underline{r} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} : \begin{matrix} x_0 \\ y_0 \end{matrix} \rightarrow \begin{matrix} x_1 \\ y_1 \end{matrix} \rightarrow \ldots \rightarrow \begin{matrix} x_k \\ y_k \end{matrix} ; i = 0 \ldots k \tag{3}$$

In order to take into account the ideally shortest path, which consists only of straight lines between the nodes, additional conditions are placed on the course of the polynomial. By inserting support nodes $j \supset i$a closer approximation to the optimale route is made possible. Defining start and end angles at which the polynomial runs in, guarantees an initially linear acceleration and deceleration.

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = j \cdot \begin{cases} \begin{bmatrix} x_i \\ y_i \end{bmatrix} & j \mod 2 = 1 \\ \begin{bmatrix} \frac{x_{i+1} - x_i}{2} + x_i \\ \frac{y_{i+1} - y_i}{2} + y_i \end{bmatrix} & j \mod 2 \neq 1 \end{cases} \tag{4}$$

The nodes and weighted edges of the optimal route are used to generate trajectories. Figure 6 shows the generated trajectory. As already described in chapter 2, these trajectories are polynomials of n^{th} order depending on the nodes obtained from the route guidance.

$$p_i' \begin{pmatrix} x_j \\ y_j \end{pmatrix} \doteqdot p_{i-1}' \begin{pmatrix} x_j \\ y_j \end{pmatrix} ; p_i'' \begin{pmatrix} x_j \\ y_j \end{pmatrix} \doteqdot p_{i-1}'' \begin{pmatrix} x_j \\ y_j \end{pmatrix} \tag{5}$$

First the mathematical boundary conditions for the construction, in this case the natural boundary suitability under the minimization of the computation effort, are to be included.

$$p_0'' \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = 0; p_{n-1}'' \begin{pmatrix} x_n \\ y_n \end{pmatrix} = 0 \tag{6}$$

But the shortest way is not the fastest way in most cases. In order to adapt the trajectory to this, an optimization is carried out with the help of a simple dynamic model of the vehicle. The main goal is to follow the trajectory as fast as possible, which results in deviations that in their entirety represent a fast path along the same support points.

The trajectory found in this way is transferred to the underlying model predictive trajectory control, presented e.g. in [18] and [19]. It works cascaded and transfers its setpoints to subordinate dynamic controllers, which influence the system and thus let it follow the calculated trajectory.

4 Demonstration of the Platform Doplar

This chapter shows the realization of the doplar platform on the basis of a pilot application in the domain of industry 4.0, which is focussed in the project Synus, and exemplary results.

4.1 Pilot application in the domain of Industry 4.0

In this subchapter an overview of the AGV's used in the cyber-physical laboratory test field is presented first. Figure 5 shows the system structure of the considered AGV. The vehicle sensors determine vehicle conditions and environmental information, which are used by the central information processing. The information processing also evaluates information from wireless communication and generates messages for the environment if necessary. With the help of the collected information, control algorithms are executed, which provide the set points of the vehicle actuators and are realized by them, so that the functionality of the AGV is accomplished.

On the vehicle side, the AGV has four electric drives close to the wheels, which are powered by a battery.

The realization of the AGV is shown in Figure 6. Mecanum wheels are used to transmit the drive torque to the road surface in order to perform omnidirectional driving maneuvers. Compared to transport vehicles with conventional wheels, the AGV does not require any maneuvering space and rotations around the vertical axis can be realized (Technologie-Netzwerk, 2016). Taking into account the environment-friendliness of the drive, the AGV's power supply is provided by a battery pack. The concept of active load handling is achieved by a conveyor belt with a height-adjustable lifting system.

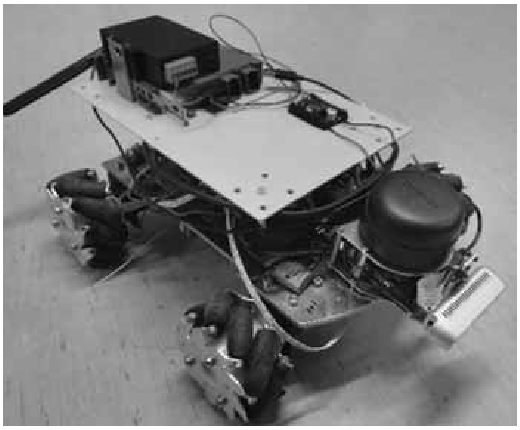

Fig. 6: Realization of the AGV

In the pilot application focussed in this paper, a spare part for a vehicle test bench (bottom left) shall be collected from an employee (top right). The AGVs are to jointly award this transport order by evaluating their optimal routes in order to achieve a minimum transport time. An optimal trajectory is to be planned for the chosen AGV to fulfill the transport order.

4.2 Exemplary results

First, the doplar was used to generate a graph from the floor plan, which enables route guidance. The fleet management initially initiates the route guidance for the three AGVs. On the basis of the routes generated, the time to complete the order is determined so that the AGV with the shortest transport time can be selected for the order. A result of the route guidance system is shown in Figure 8. The three determined routes, which also seem optically plausible, are recognizable. AGV 3 has the shortest route to the order location and therefore also requires the shortest transport time so that this AGV should be selected for the transport order.

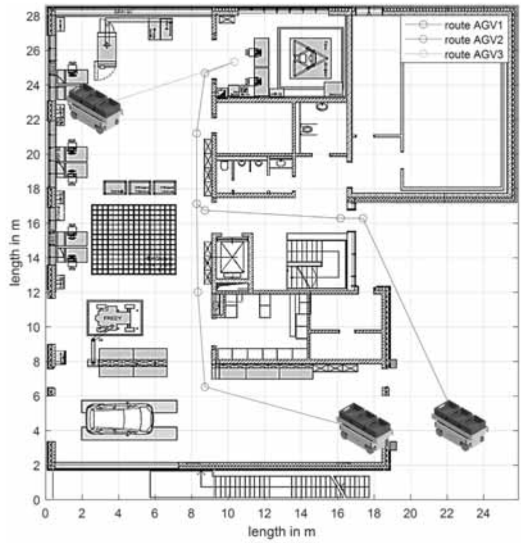

Fig. 7: Result of the route guidance for the three AGVs as basis for the fleet management for the distribution of the transport order

This pilot application shows exemplary the functionality of the platform doplar from generating a graph map from a floor plan to route guidance. further examples are concluded in [2].

5 Conclusion and Outlook

In this article the domain-specific configurable modular platform for route guidance and trajectory planing of intelligent vehicles (doplar) was presented, which can be used in different projects across domains. The great advantage of this platform is that core components such as the navigation and guidance algorithm can be retained and only individual modules, e.g. for wireless communication, have to be adapted. The aim of the entire platform is to plan optimized vehicle operation with regard to journey time and energy consumption, taking into account dynamic environmental data available from wireless communication within the cyber-physical transport system. The platform specifications serve as target values for subordinate systems such as integrated vehicle dynamics control or for awarding an order in an industrial 4.0 production plant. The doplar platform was demonstrated in a pilot application for an industrial 4.0 production plant in the context of the project Synus, in which three AGVs can be used for transport tasks. The fleet management initiates a target guidance of all available AGVs, which forms the basis for the decision to award a contract. The AGVs exchange their forecasted routes with each other and jointly allocate the order. For the AGV that has received the order, doplar generates an optimal trajectory for its completition. In the following work steps, the developed platform doplar is to be integrated into other domains and real systems, examined under real-time conditions and further optimized.

Acknowledgement

This contribution evolved from the research project Model-based Conception and Evaluation of Industry 4.0 Solutions for Networking Mechatronic Components in Production Facilities by Digitalization (MIMec) which is part of the collaborative project Methods and Tools for the Synergetic Conception and Evaluation of Industry 4.0 Technologies (Synus). It is funded by the European Regional Development Fund (EFRE: ZW 6 85012454).

References

[1] Liu-Henke X. *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab: Zugl.: Paderborn, Univ., Diss., 2005*, vol. 589 of *Fortschritt-Berichte VDI Reihe 12, Verkehrstechnik/Fahrzeugtechnik*. Düsseldorf: VDI-Verl., als ms. gedr ed. 2005.

[2] Jacobitz S, Göllner M. Seamless validation of cyber-physical systems under real-time conditions by using a cyber-physical laboratory test field. In: *IEEE International Conference on Recent Advances in Systems Science and Engineering*. 2021; .

[3] Kortenkamp D, Weymouth T. Topological Mapping for Mobile Robots Using a Combination of Sonar and Vision Sensing. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2)*, AAAI'94. USA: American Association for Artificial Intelligence. 1994; pp. 979–984.

[4] Thrun S, Burgard W, Fox D. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Machine Learning*. 1998;31(1/3):29–53.

[5] Kweon IS, Kanade T. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992; 14(2):278–292.

[6] Neira J, Tardos JD. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*. 2001; 17(6):890–897.

[7] Leonard JJ, Durrant-Whyte HF. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*. 1991; 7(3):376–382.

[8] Benkert M, Widmann F, Wolff A. The Minimum Manhattan Network Problem: A Fast Factor-3 Approximation.

[9] Walter R. Metrische Räume. In: *Einführung in die Analysis*, edited by Walter R, pp. 270–327. Berlin: de Gruyter. 2007;.

[10] Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959;1(1):269–271.

[11] Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*. 1968;4(2):100–107.

[12] Bellman R. On a routing problem. *Quarterly of Applied Mathematics*. 1958;16(1):87–90.

[13] Floyd RW. Algorithm 97: Shortest path. *Communications of the ACM*. 1962;5(6):345.

[14] Warshall S. A Theorem on Boolean Matrices. *Journal of the ACM*. 1962;9(1):11–12.

[15] Kleene SC. Representation of Events in Nerve Nets and Finite Automata. In: *Automata Studies. (AM-34)*, edited by Shannon CE, McCarthy J, Annals of Mathematics Studies, pp. 3–42. Princeton, NJ: Princeton University Press. 1956;.

[16] Golfarelli M, Maio D, Rizzi S. Elastic correction of dead-reckoning errors in map building. In: *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 2. 1998; pp. 905–911 vol.2.

[17] Scherler S, Liu-Henke X, Henke M. Predictive Energy Management for an Electric Vehicle with Fuel Cell Range Extender in Connected Traffic Systems. In: *2020 19th International Conference on Mechatronics - Mechatronika (ME)*. IEEE. 02.12.2020 - 04.12.2020; pp. 1–8.

[18] Göllner M, Zhang J, Liu-Henke X. Model predictive trajectory control for automated driving of a spherical electrical drive. In: *2018 IEEE International Systems Engineering Symposium (ISSE)*. IEEE. 10/1/2018 - 10/3/2018; pp. 1–6.

[19] Göllner M. Fusion structure for environment perception and model-predictive navigation of autonomous vehicles. 25.06.2019.