Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, October 6-9, 2013

Novel Model Based Path Planning for Multi-Axle Steered Heavy Load Vehicles

Susann Beyersdorfer[1] and Sebastian Wagner[2]

***Abstract*— Maneuvering heavy load transports on narrow roads is a challenging task that requires a precise and reliable planning in advance. This paper introduces a novel procedure that improves planning and execution of heavy load transports by using path planning algorithms. The proposed algorithm augments existing approaches significantly to enable path planning for multi-body vehicles with multi-axle steering.**

I. INTRODUCTION

A. Problem Statement

Efficient freight traffic is a central interest in modern economy, whereas many goods are transported on roads. Heavy load transports are a special type thereof used to transport large and heavy loads by long articulated vehicles, e.g. Fig. 1.

For example modern wind power plants kept enlarging for the last decade requiring suitable heavy load transport vehicles. But that vehicles are difficult to handle on ordinary road infrastructure. One major disadvantage is their bad maneuverability. To overcome this problem at least partially modern trailers are equipped with steered axles. Thereby two kinds of steering are common. Manual steering means steering via control panel. Forced steering applies fixed steering schemes that depend on the current articulation angle.

Limited maneuverability is especially problematic regarding urban infrastructure, because urban streets are not designed for heavy load vehicles. That is why, for each transport, a time-consuming planning phase is necessary to find a passable route.

Nowadays, planning routes is based on expert knowledge and is often done using simple geometric tests and freehand sketches. But this gives only rough, unreliable estimates for the passability of narrows. In turn, despite the cost-intensive planning, occasionally vehicles get stuck, resulting in high and expensive recovery efforts.

B. Related Work

Regarding the state of the art in general path planning and route planning can be differentiated.

Route planning is known from modern navigation systems. But although these systems use path planning algorithms to find an optimal route they assume a point-like vehicle and therefore cannot plan collision free paths for real life vehicles. When considering the available space becomes necessary more advanced planning procedures are required.

Fig. 1. Transportation of Fraunhofer IVI's AutoTram® Extra Grand

Automated path planning solutions for real vehicles are known from automatic parking systems for cars, where maneuver primitives are used to drive a vehicle into a parking lot, e.g. [1]. More sophisticated systems perform online path planning in the content of reversing multi-axle vehicles with one steering axle, ([2], [3], [4]). Those solutions are developed for a single path shape and cannot be used for other tasks like finding paths on narrow roads.

In literature path planning algorithms are used to calculate collision free paths for single vehicles and vehicle-trailer combinations. In the field of path planning for vehicle-like systems graph searching algorithms have been proved to be successful. These are search procedures, where - beginning at a given start configuration - the free space is explored iteratively for adjacent configurations that lead to the target configuration. A major problem within graph searching is the dimension of the configuration space, i.e. the space of all possible configurations, that has to be searched for a solution path. Adding one dimension to the configuration space (e.g. a drawbar or a steering axle at the rear of a vehicle module) extremely increases computational expenses that have to be dealt with. Simulations have been made for vehicles with one steering axle, e.g. [5], [6], [7]. Stopp et al. [8] describe a path planning algorithm tested with a real life vehicle combination and thus show that their planning procedure leads to a realizable solution. Lamiraux et al. [9] published a remarkable algorithm for a tractor-semitrailer vehicle with one steering axle that was used to transport A380 components with an overall transport length of 50 meters. This work comes closest to the presented objective. The main

[1]S. Beyersdorfer is with Fraunhofer Institute for Transportation and Infrastructure Systems IVI, Dresden, Germany `susann.beyersdorfer at ivi.fraunhofer.de`

[2]S. Wagner is with Fraunhofer Institute for Transportation and Infrastructure Systems IVI, Dresden, Germany `sebastian.wagner at ivi.fraunhofer.de`

978-1-4799-2914-613/$31.00

drawback is the lack of multi-axle steering support required for planning with modern heavy load transport vehicles. Furthermore truck-trailer combinations are not considered.

Regarding the special problem of planning routes for heavy load transports, software tools are available that simulate vehicle motions and detect collisions, e.g. 'easyTrack' offered by RZI Software GmbH. To start the simulation paths have to be defined manually. If a collision occurs during simulation, the paths have to be adapted manually. Using those tools is time consuming and beyond that possible solutions may not be found. As is the case for the other approaches multi-axle steering is not supported.

Summarizing the state of the art, there is no automatic path planning algorithm for multi-axle steered articulated vehicles available.

C. Solution Overview

This paper presents a new approach for the automated calculation of particular paths for every steering axle resulting in collision free paths for the whole vehicle combination. Therefore, a novel path planning algorithm will be introduced. Within this algorithm feasible paths are calculated using a kinematic vehicle model allowing the calculation of manual as well as forced steering for the trailer. The main objective is to develop an approach that is suitable for the most common vehicle configurations, like truck-trailer and tractor-semitrailer vehicles. Furthermore ground clearance of the vehicle modules is considered and checked against heights of over drivable obstacles during collision detection, so that almost all available space is used.

In contrast to common procedures, the presented algorithm provides guaranteed results regarding the passability of narrows. Resulting paths can further be used to assist the driver during transport. Using the described algorithm, planning and execution of heavy load transports can be made more efficient, cheaper and safer in future.

The presented algorithm is based on a kinematic vehicle model. In principle path planning can be done for truck-trailer vehicles and tractor-semitrailer vehicles. This paper concentrates on tractor-semitrailer vehicles. Chapter II describes the kinematic vehicle model. Afterwards the path planning algorithm will be discussed in detail in III.

II. KINEMATIC VEHICLE MODEL

A. Configurable vehicle

The underlying data structure for the vehicle model is based on DIN 70020 [10], which provides a consistent description of common vehicle combination dimensions. Using this data structure makes it possible to configure all kinds of real life tractor-semitrailer vehicles. Relevant parameters for the kinematic vehicle model as well as a geometric model for collision checking are calculated automatically.

B. Kinematic vehicle model

Heavy load transport vehicles contain multiple bodies and trailers with multiple steering axles. For low velocities kinematic models represent the real vehicle motion in good quality. Fig. 2 shows the kinematic parameters of a tractor-semitrailer with off axle hitching and multi-axle steering. Two- and multi-axle vehicles belong to the nonholonomic systems, i.e. they are subject to restrictions of the form

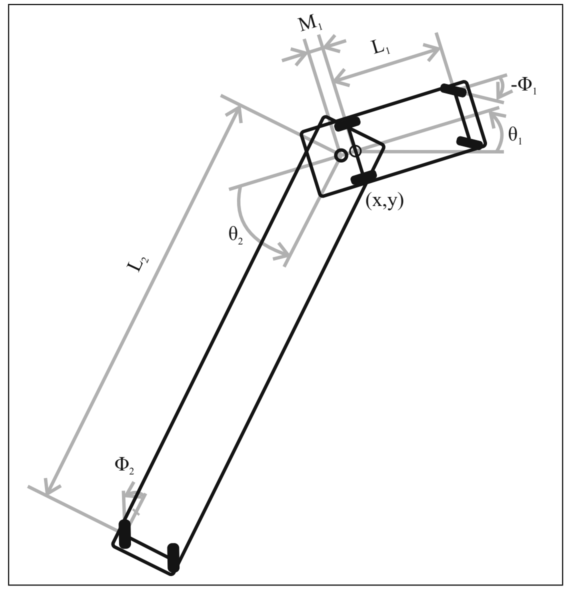

Fig. 2. Kinematic parameters of a tractor-semitrailer with off-axle hitching

$$G(q,\dot{q},t) = 0. \tag{1}$$

In (1) q is the vehicle configuration and t is time. For instance the nonholonomic condition for the truck's rear axle is

$$\dot{x}\sin(\theta_1) - \dot{y}\cos(\theta_1) = 0. \tag{2}$$

Whereas x and y denote the position of the midpoint of the truck's rear axle and θ_1 the orientation of the truck. For a tractor-semitrailer vehicle the degree of freedom is four. Thus, modeling the truck by two compensatory axles and the trailer by one compensatory axle is sufficient for simulating the behavior of a real life vehicle with multiple axles and will be exploited within developing the kinematic model. Using the concept of rolling without slipping and therefore applying (2) to every axle, the kinematic model for manual steering of form

$$\dot{q} = f(q,u) \tag{3}$$

can be determined

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi}_1 \\ \dot{\theta}_1 \\ \dot{\phi}_2 \\ \dot{\theta}_2 \end{pmatrix} = \begin{pmatrix} \cos\theta_1 \\ \sin\theta_1 \\ 0 \\ \frac{1}{L_1}\tan\phi_1 \\ 0 \\ -\left(\frac{\tan\phi_1}{L_1} + \frac{\sin(\phi_2+\theta_2)}{L_2\cos\phi_2} + \frac{M_1}{L_1 L_2}\frac{\tan\phi_1\cos(\theta_2+\phi_2)}{\cos\phi_2}\right) \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \dot{\phi}_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \dot{\phi}_2. \tag{4}$$

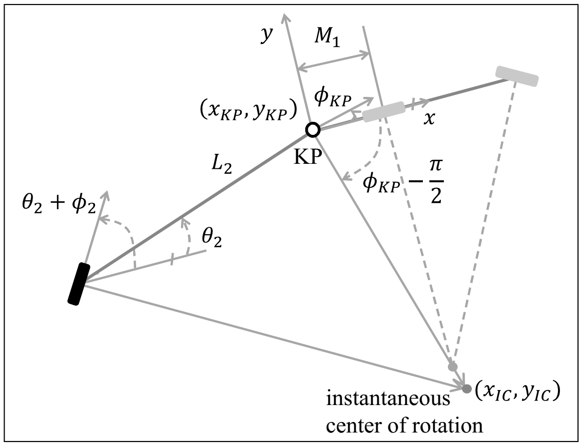

Fig. 3. Extended law of Ackermann: instantaneous center of rotation

In (4) the configuration vector q of the system is $q = (x, y, \phi_1, \theta_1, \phi_2, \theta_2)^T$, where x, y and θ_1 denote the truck parameters as in (2), θ_2 is the articulation angle and ϕ_1 and ϕ_2 are the steering angles of truck and trailer steering axles respectively. Control inputs u are the vehicle velocity v and the steering angle derivatives $\dot{\phi}_1$ and $\dot{\phi}_2$. Vehicle parameters are given by the truck wheel space L_1, the off axle hitching M_1 and the trailer wheel space L_2.

In case of forced steering ϕ_2 is always equal to θ_2, as long as the maximum steering angle is not violated. Thus $\dot{\phi}_2 = \dot{\theta}_2$ and the equation for $\dot{\phi}_2$ can be neglected.

Possibly occurring model errors can be corrected by the driver. Therefore, a remaining margin in steering angles has to be considered during path planning.

C. Calculation of trailer steering angles

Although one trailer axle is sufficient for the derivation of the kinematic vehicle model, it is necessary to verify the adherence to steering angle restrictions for the remaining axles.

These steering angles can be determined by extending Ackermann's law, considering the instantaneous centers of rotation, as shown in Fig. 3 and Fig. 4. Without loss of generality it is assumed, that the global coordinate system has its point of origin at the king pin with $\theta_1 = 0$.

At first the position of the instantaneous center of rotation (x_{IC}, y_{IC}) is determined by calculating the intersection of the lines perpendicular to the velocity vectors at the king pin and the midpoint of the trailers last axle

$$\begin{pmatrix} x_{IC} \\ y_{IC} \end{pmatrix} = -L_2 \begin{pmatrix} \cos\theta_2 \\ \sin\theta_2 \end{pmatrix} + s_{IC} \begin{pmatrix} \cos(\theta_2 + \phi_2 - \frac{\pi}{2}) \\ \sin(\theta_2 + \phi_2 - \frac{\pi}{2}) \end{pmatrix} \quad (5)$$

with

$$s_{IC} = L_2 \frac{\sin(\phi_{KP} - \theta_2 - \frac{\pi}{2})}{\sin(\phi_{KP} - \theta_2 - \phi_2)}. \quad (6)$$

In (6) ϕ_{KP} denotes the direction of velocity at the king pin (KP), that can be determined using the law of Ackermann for the truck axles (Fig.3)

$$\phi_{KP} = \tan^{-1}\left(\frac{M_1}{L_1 \tan(\phi_1 - \frac{\pi}{2})}\right). \quad (7)$$

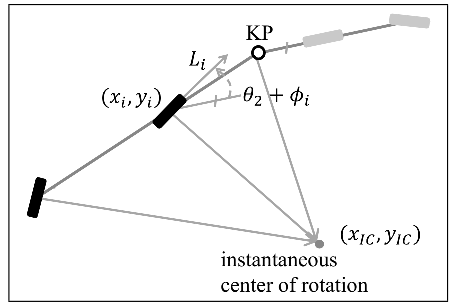

Fig. 4. Extended law of Ackermann: calculate trailer steering angles

The other parameters are chosen analog to the configuration description of (4) and as in Fig. 2. The calculation of the remaining steering angles is then given by

$$\phi_i = \tan^{-1}\left(-\frac{(x_{IC} - x_i)}{(y_{IC} - y_i)}\right) - \theta_2 \quad (8)$$

with

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = -L_i \begin{pmatrix} \cos\theta_2 \\ \sin\theta_2 \end{pmatrix}. \quad (9)$$

The index i denotes the i-th trailer axle.

To make sure that only feasible paths are planned, the steering angles as well as the articulation angle must remain within the specified interval. While checking the articulation angle requires almost no resources, checking all steering angles would result in high computational effort. Since it can be shown that the first and/or last trailer axle always has the highest amplitude in steering angles, only these two axles need to be checked.

III. A NOVEL PATH PLANNING ALGORITHM

The general objective of path planning is to find a collision free path through an environment containing obstacles. Using the vehicle model in II a novel path planning algorithm is designed to find a solution to the transportation problem. The algorithmic core of the proposed path planning approach is based on the well known A*-Algorithm, a graph based search procedure that is guaranteed to find a minimum cost solution, if a solution exists. Within the A*-Algorithm a search tree consisting of nodes is explored by iteratively adding new nodes to it. A node contains the configuration, a cost value and a reference to its predecessor node. For a detailed description of A* see [12]. In the scope of planning paths for heavy load vehicles the A*-Algorithm needs to be adapted to multi-body vehicles with multiple steering axles. Fig. 5 depicts the principle of the original A*-Algorithm and the developed adaptions. The following steps are performed subsequently:

- **Determining potential neighbor nodes:** Starting at a current node neighbor nodes are explored and added to the search tree. Within the adapted version neighbor nodes are calculated using the kinematic vehicle model to exclude not available nodes from the start.
- **Collision detection:** The neighbor nodes are checked for collisions. Within the scope of planning paths for heavy load vehicles a special obstacle map is used that

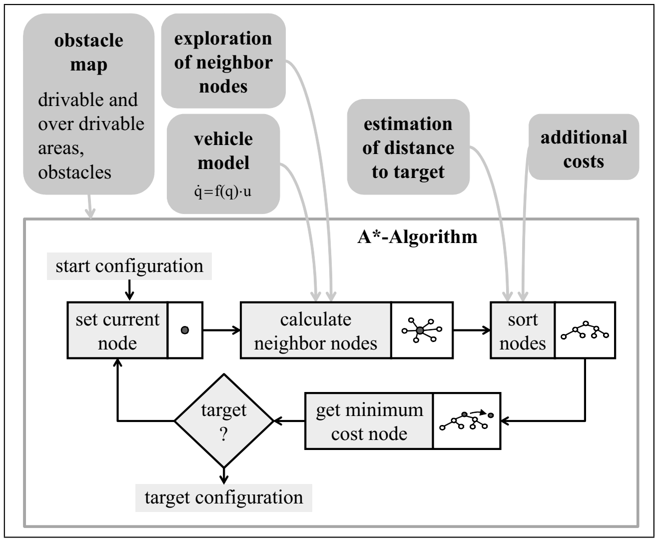

Fig. 5. Principle of path planning algorithm for multi-axle steered vehicles

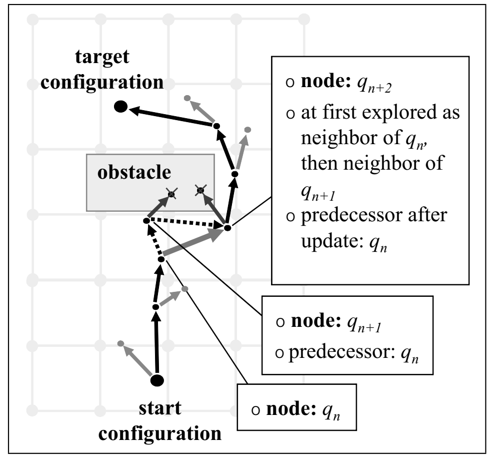

Fig. 6. Finding minimum cost paths and grid map

TABLE I
PROCEDURE OF THE MODIFIED A*-ALGORITHM

```
Modified A*-Algorithm
initialize search with start configuration,
add node to search tree
WHILE (target configuration not achieved
  AND still elements in search tree)
  delete first node of search tree (node with minimum costs),
  add it to 'closed list',
  sort search tree
  FOR(every combination of control parameters)
    calculate vehicle trajectories,
    check whether vehicle is in collision
    IF(vehicle ISNOT(in collision))
      calculate configuration costs
      IF(configuration ISNOT in search tree)
        add node to search tree,
        sort search
      ELSE IF(configuration is in search tree
        AND has lower costs)
        update node in search tree,
        sort search tree
      END IF #configuration in lists
    END IF #collision
  END FOR #adjacent configurations
END WHILE
IF(target configuration achieved)
    backtrack nodes in 'closed list'
ELSE
    stop procedure
END IF #target achieved
```

considers drivable areas and (over drivable) obstacles, see III-A.
- **Evaluation of costs:** All newly explored nodes are evaluated by heuristic cost functions. The cost functions are described in III-C. Furthermore, the usage of additional costs will be introduced in III-D.
- **Determination of cheapest successor:** After insertion of collision free nodes the search tree is sorted. Then the minimum cost node is taken from the tree and inserted in a so called '*closed list*'. This node will be the next to be explored. The paths leading to these nodes are optimal, by definition of A*.

This procedure is executed until the minimum cost node taken from the search tree is within a defined distance to the target configuration. Once, the target node is arrived, the minimum cost solution is automatically found by back tracking the predecessor nodes in '*closed list*' beginning at the target node. For a detailed description of the procedure see Tab. I and [11].

A. Obstacle map

The obstacle map is used for collision checking adapted to the problem of heavy load transports. Suitable obstacle data are available from different 2D map sources, e.g. satellite pictures (Google Maps), photographs or even hand drawings. The 2D map data will be partially expanded with height information.

On a map polygonal regions for different types of areas need to be defined in advance:
- **Drivable areas**: All vehicle parts including the wheels are allowed to drive along.
- **Over drivable obstacles**: Every over drivable obstacle gets a value for its height. This height is used to determine vehicle parts that are in collision. While wheels are generally not allowed other parts like e.g. the load or the semitrailers H-frame might be allowed to be above a certain lower height limit. Examples for over drivable obstacles are sidewalks or the isles of traffic circles.
- **Obstacles**: No part of the vehicle is allowed to be located in these areas. Some examples for obstacles are houses and trees.

Based on the polygons and height values a pixel map is calculated that provides fast collision checking. The height values will be checked against obstacles by ground clearance values of the geometric vehicle model.

B. Exploration of neighbor nodes using a vehicle model

For the exploration of neighbor nodes feasible paths are calculated using the kinematic vehicle model. Therefore,

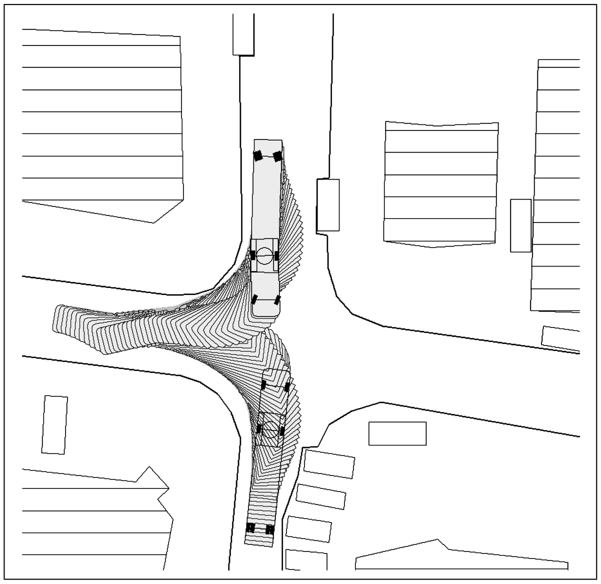

Fig. 7. Solution of path planning for a reversing maneuver

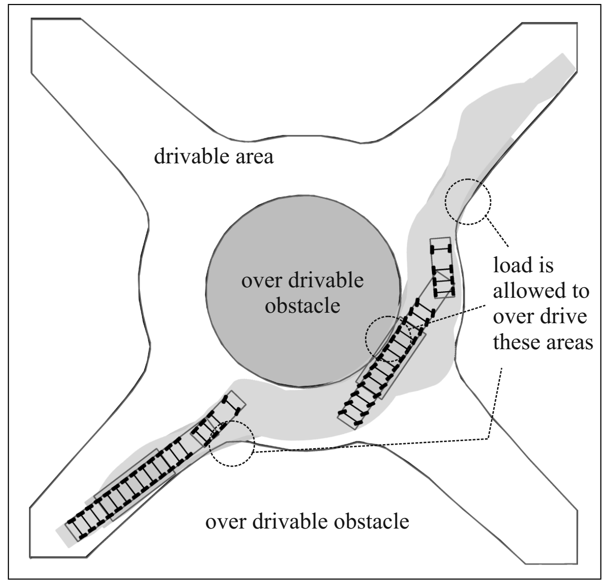

Fig. 8. Solution of path planning for passing a traffic circle

in each iteration step a set of n vehicle controls $u = (v, \dot{\phi}_1, \dot{\phi}_2)^T$ is chosen. This set is applied to the current vehicle configuration resulting in n successor configurations q_i, $(i = 1..n)$, that are calculated by numerical integration. The resulting configurations q_i belong to potential new neighbor nodes in the search graph. In the next step, each successor configuration q_i is tested in terms of collision with obstacles, resulting in non-colliding successor configurations. All non-colliding configurations are accessed by a cost function, that evaluates the configurations regarding an optimization criterion, e.g. shortest paths. Afterwards control inputs, configurations, costs and a predecessor reference are summarized in nodes and inserted in the search tree. Thus the search tree is extended for newly explored successor nodes in each iteration step. In case a node has been explored in a previous step it will replace the existing node if its costs are lower than the costs of the existing one. Thus the algorithm always decides to the benefit of lower cost paths, which is a key feature of A*. Fig. 6 depicts this behavior for a point robot with euclidean distance function: the path including node q_{n+1} related to the dashed black arrows is updated to a shorter path with lower costs directly leading from q_n to q_{n+2} during the search, due to an obstacle.

To check if a node has been explored in a previous step a grid map is used, which is a discretization of the four dimensional space spanned by $(x, y, \theta_1, \theta_2)$. In terms of saving computation time and memory steering angles are not considered. In contrast to other approaches configuration representations are not restricted to the resolution of the underlying grid map, as illustrated in Fig. 6. The grid map is solely used to decide if a node has been visited before or not.

C. Estimation of distance to target

During path planning cost functions are used to decide which node is chosen for further explorations. The classical A*-Algorithm uses the path length as cost criterion for path planning problems. For the evaluation of a node in the search tree the distance to start and an estimate for the remaining distance to the target configuration are calculated and added, ignoring obstacles. For a point robot the euclidean path length is an optimal estimate for the remaining distance to the target configuration.

For car like robots and truck-trailer vehicles finding a good estimate for the distance to the target configuration is more difficult, because the orientation of each vehicle module has to be considered. Reeds and Shepp introduced a solution for shortest paths for a car-like vehicle in [13]. For articulated vehicles no such solution is known. The presented algorithm suggests a weighted euclidean distance between the current configuration and the target configuration as an estimate for the remaining costs C_{target}, referring to [14]

$$C_{target}(q) = \Big(|x_t - x|^2 + |y_t - y|^2 + (w_1 |\theta_{1_t} - \theta_1|)^2 + (w_2 |\theta_{2_t} - \theta_2|)^2 \Big)^{\frac{1}{2}}. \qquad (10)$$

In (10) the index t denotes the target configuration and w_1 and w_2 are weighting factors. The distance to start is always calculated analog to the estimate for remaining costs, because the cost values have to be comparable.

D. Introduction of additional costs

From a practical point of view it is reasonable to take additional cost functions into account. The following criteria can be added as a cost function to alter path shapes:

- required maneuvering area
- steering effort
- articulation angle
- distance to obstacles
- number of cusps
- steering power
- driving power

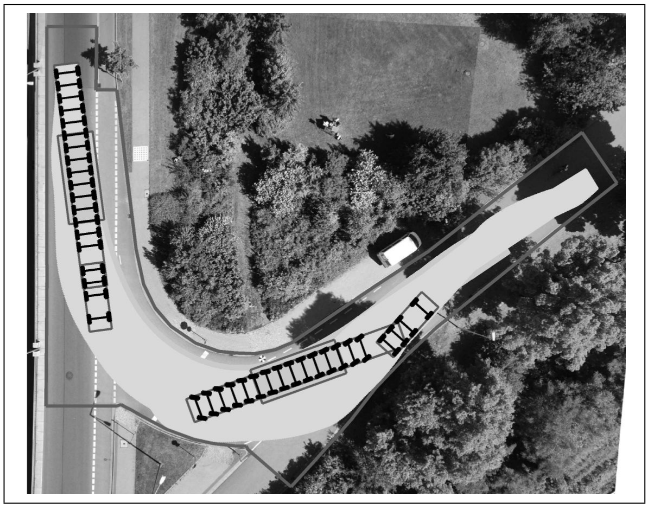

Fig. 9. Solution of path planning for a turning maneuver

E. Test scenarios

For systematic tests concerning the algorithm functionality and performance, special test scenarios have been designed. These scenarios combine real life infrastructure data and representative vehicle models. For the underlying maps drivable areas and (over drivable) obstacles are defined.

IV. RESULTS

The proposed path planning algorithm is adapted to planning heavy load transports with multi-axle steering. A framework for systematic tests is introduced, using parametric vehicle models, a multiple source map generation and test scenarios. Path planning is based on a graph search procedure, whereas all nodes in the search tree are accessible by feasible paths, thereby exploring unnecessary nodes is avoided. During the search nodes with nearly equal configurations have to be compared by their costs. In disregarding steering angles for the comparison the dimension of the introduced grid map is reduced to four. Through these adaptions, the problem of computing effort, i.e. computation time and memory, is manageable. Tests for typical heavy load vehicles and traffic environment showed that paths can be planned successfully and reliably within few minutes on an ordinary desktop computer.

Fig. 7 shows a reversing maneuver at a standard intersection which is based on a 2D representation of a narrow. Fig. 8 and Fig. 9 depict test scenarios as described in III-E. In Fig. 8 a tractor-semitrailer is driving through a traffic circle. Within this planning the load was allowed to over drive the isle in the middle of the circle and the street boundary, as the dashed circles in the figure indicate. Fig. 9 shows a turning maneuver at a highly frequented route for heavy load transports in Dresden. For this intersection no up to date map data was available. Thus Fraunhofer IVI's octocopter HORUS (www.horus.mobi/en.html) was used to take high resolution photographs as basis for path planning.

For the presented paths a complex cost function was used, that combines the criteria mentioned in III-D.

As can be seen, the path planning algorithm calculates collision free paths for each steering axle. The shown figures can be used to prove the passability of critical narrows and to assist the driver during the transport.

V. CONCLUSIONS

In the scope of heavy load transports the presented model based path planning algorithm for the first time provides feasible and collision free paths for multi-axle steered vehicles. For a correct representation of real life vehicles manual and forced steering as well as off-axle hitching is modeled for the semitrailer. The algorithm takes drivable areas and obstacles into account. For collision checks height values of obstacles as well as values for ground clearance of vehicle modules are considered. Thus, the full potential of the vehicle combination can be exploited during the planning. The algorithm is implemented modularly, so that it is expandable to other vehicle combinations like truck-trailer vehicles or e.g. collision test procedures.

Future work will concentrate on an intensive analysis and minimization of computing time. Other tasks are decreasing computational costs by adapting the simulation step size in each iteration step as well as investigating the required cost functions. Furthermore, integrating truck-trailer vehicles with a drawbar will also be in the scope of future work.

REFERENCES

[1] Continental Automotive GmbH, Semiautomatic parking machine, Patent Application No. WO 2008/104488 A1, 2008.

[2] DaimlerChrysler AG, Control system for a tractor trailer, Patent Application No. WO 2005/080179 A1, 2005.

[3] DaimlerChrysler AG, Rueckwaertsfahrhilfesystem und Verfahren zur Unterstuetzung des Fahrers eines Zugfahrzeug-Anhaenger-Gespanns bei einer Rueckwaertsfahrt, Patent Application No. DE 10 2005 043 466 A1, 2007.

[4] DaimlerChrysler AG, Verfahren zur Unterstuetzung des Fahrers eines Fahrzeugs beim Rangieren und/oder Einparken, Patent Application No. DE 10 2007 032 720 A1, 2008.

[5] J.-C. Latombe and J. Barraquand, Nonholonomic multibody mobile robots: controllability and motion planning in the presence of obstacles, IEEE Int. Conf. on Robotics and Automations, vol. 3, 1991.

[6] S. Sekhavat, P. Svestka, J.P. Laumond and M.H. Overmars, Multilevel path planning for nonholonomic robots using semi-holonomic subsystems, Int. Journal of Robotics Research, vol. 17, 1996.

[7] E. Aranda-Bricaire, T.A. Vidal-Calleja and M. Valesco-Villa, Artificial potential fields for trailer-like systems, 10th Latinamerican Congress on Automatic Control Guadalajara, 2002.

[8] A. Stopp, R. Stahn and R. Stark, Laser scanner-based navigation and motion planning for truck-trailer combinations, IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, 2007.

[9] F. Lamiraux, J.P. Laumond, C. Van Geem, D. Boutonnet and G. Raust, Trailer-truck trajectory optimization for airbus A380 component transportation, IEEE Robotics & Automation Magazine, vol. 12, pp. 12-21, 2005, DOI: 10.1109/MRA.2005.1411414.

[10] DIN 70020, Kraftfahrzeugbau-Begriffe von Abmessungen,, Berlin: Beuth Verlag, 1993.

[11] S. Beyersdorfer, S. Wagner and S. Zipser, Obstacle avoidance for multi-axle steered multi-body vehicles, 56th Int. Scientific Colloquium, TU Ilmenau, September 2011.

[12] P.E. Hart, J. Nilsson and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics, vol. SSC-4, no. 2, 1968.

[13] J.A. Reeds and L.A. Shepp, Optimal paths for a car that goes forwards and backwards, Pacific Journal of Mathematics, vol. 145, no. 2, 1990.

[14] M.M. Deza, E. Deza, Encyclopedia of Distances, Berlin Heidelberg: Springer-Verlag, 2009, p. 312.