URN (Paper): urn:nbn:de:gbv:ilm1-2011iwk-133:3

56TH INTERNATIONAL SCIENTIFIC COLLOQUIUM
Ilmenau University of Technology, 12 Ű 16 September 2011
URN: urn:nbn:gbv:ilm1-2011iwk:5

OBSTACLE AVOIDANCE FOR MULTI-AXLE STEERED MULTI-BODY VEHICLES

Susann Beyersdorfer, Sebastian Wagner, Stephan Zipser

Fraunhofer Institute for Transportation and Infrastructure Systems IVI
Zeunerstr. 38
01069 Dresden, Germany

ABSTRACT

Manoeuvring a large heavy load vehicle through difficult scenarios is not always an easy task. This paper presents a path planning algorithm, based on the A*-Algorithm, that calculates collision free paths for multi-axle steered multi-body vehicles. So it is possible to predict whether the vehicle can pass its designated route or if the route needs to be changed. The precalculated manoeuvres can be used to assist the driver.

Index Terms— Path planning, obstacle avoidance, multi-axle steering, multi-body

1. INTRODUCTION

During the last years increased requirements on carrying capacity and efficiency have caused a trend to longer heavy load vehicles. Obviously a major drawback of long vehicles is the limited manoeuvrability. To overcome this disadvantage at least partially the vehicles where designed of multiple bodies equipped with multiple steering axles. These steering axles are actuated by a mechanical steering linkage, approximately satisfying the law of Ackermann. In modern vehicles the axles are often electronically steerable (steer-by-wire is already standard in modern agricultural machinery like tractors and combine harvesters or in heavy load vehicles). Most of this axles are controlled by fixed steering schemes. In difficult situations the axles are steered manually via a control panel. In the course of planning large heavy load transports it is actually challenging for the responsible dispatcher to decide whether e.g. a narrow crossroad can be passed or not.
For the planning task it would be easier to know if a difficult scenario can be passed before the transport has started. Furthermore advanced steering strategies based on individual electronic steerable axles can provide a much better manoeuvrability. Therefore this paper presents the calculation of individual paths for each steering axle providing a collision free passing.
To solve the problem path planning algorithms are used. The suggested algorithm is based on the well known A*-Algorithm and will be discussed further.

1.1. Related work

Collision avoidance in general is a task that has been topic of many investigations and developments. A well known example is the parking assist system for cars, e.g. [1]. Assistance systems for parking and reversing are also patented for multi-axle vehicles with front-wheel-steering [2, 3, 4].
In literature autonomous driving is described using path planning algorithms. Simulations have been made for front-wheel-steered vehicles with two or more axles, e.g. [5, 6, 7]. Path planning algorithms that where experimentally verified are introduced in [8] and [9]. In [8] path planning is used to drive a truck-trailer backwards under a swap body automatically. [9] describes a path planning algorithm for a long truck-trailer used for the transport of A380 components.
Autonomous manoeuvring for more complex vehicles, especially for vehicles with multi-axle steering that can be controlled individually, has been discussed in [10].

1.2. Path planning

The objective of path planning is to find a collision free path through an environment containing obstacles. Therefore the vehicle is presented by its configuration q. The space of all configurations is called C, and the space of all free configurations (i.e. configurations that are not in collision) is called C_{free}. Two- and multi-axle vehicles belong to the nonholonomic systems, i.e. they are subject to restrictions of the form

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

With the nonholonomic condition (coming from the roll without slipping constraint)

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

the kinematic model of the system of form

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

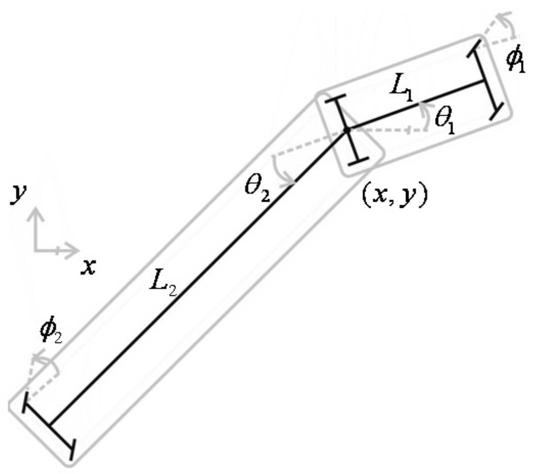


Fig. 1. Kinematic model of an articulated truck with two steering axles

can be established. Whereas x and y denote the position and θ the orientation of the vehicle. u is the control vector of the system.
In the field of path planning for vehicle-like systems graph searching algorithms have been proofed to be successful. That are search procedures, where - beginning at a given start configuration - the space C_{free} is scanned iteratively for adjacent configurations that lead to the goal configuration. New configurations are evaluated by cost functions. When the goal configuration is arrived, the planned path results by backtracking the minimum cost configurations.

2. A PATH PLANNER FOR MULTI-AXLE STEERED VEHICLES

In this section the suggested path planning algorithm for multi-axle steered vehicles is presented using the example of an articulated truck with two steering axles, Fig. 1. First, the kinematic model of the system is given in subsection 2.1. Then the algorithm and its characteristics are explained in subsections 2.2 to 2.4.

2.1. Kinematic model

The kinematic model of the articulated truck with two steering axles shows Fig. 1. Thereby x and y denote the position of the rear-axle of the truck, θ_1 describes the orientation of the truck with respect to the fixed x-axis and θ_2 is the orientation of the trailer referred to the longitudinal axis of the truck. ϕ_1 and ϕ_2 are the respective steering angles. Applying (2) for every axle, the kinematic model can be established, cf. (4).

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

Table 1. Procedure of the modified A*-Algorithm

```
Modified A*-Algorithm
----------------------------------------------------------
initialise search with start
configuration, add node to 'open
list'
WHILE(goal configuration not
achieved AND still elements in 'open
list')
  delete first node of 'open
  list'(node with minimum costs)
  and add it to 'closed list',
  sort 'open list'
  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
      'open list') AND ISNOT(in
      'closed list'))
        add node to 'open list',
        sort 'open list'
      ELSE IF(configuration is in
      'open list' AND has lower
      costs)
        update node in 'open list',
        sort 'open list'
      END IF #configuration in lists
    END IF #collision
  END FOR #adjacent configurations
END WHILE
IF(goal configuration achieved)
    backtrack nodes in 'closed list'
ELSE
    stop procedure
END IF #goal achieved
```

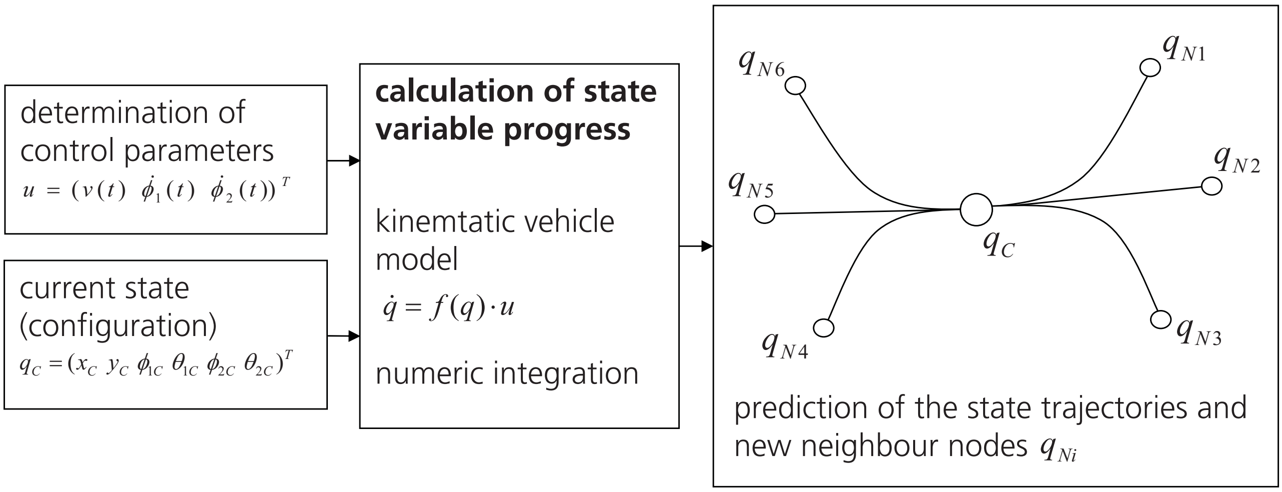


Fig. 2. Modified A*-Algorithm: Choosing adjacent configurations

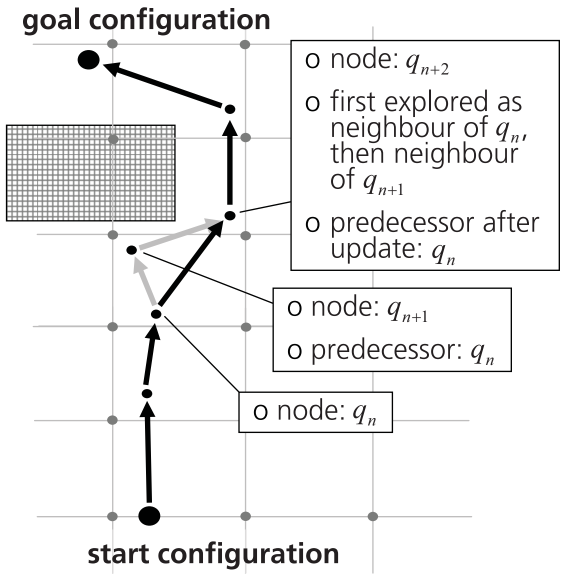


Fig. 3. Modified A*-Algorithm: Principle and grid map

2.2. Modified A*-Algorithm - Principle

The path planning algorithm for multi-axle steered multi-body vehicles is based on the classical graph search algorithm A*, introduced in [11]. Table 1 shows the procedure of the modified algorithm. As described in [11], two lists are used to sort nodes during the search procedure. A node in the search tree stores basically its configuration, its control parameters, its active costs and its predecessor. The algorithm is initialised with the vehicle's start configuration, i.e. the node that belongs to the start configuration is inserted in a list called 'open list'. 'open list' is a binary heap in which the nodes are sorted according to their costs. At the beginning this list is empty. In every iteration step the neighbour configurations are calculated (subsection 2.3) and their costs are determined (subsection 2.4). Nodes are inserted in 'open list' if they are collision free. After a node is inserted, 'open list' is sorted again. If a neighbour configuration is already represented by a node in 'open list' and has lower costs as this node, then the node is updated (esp. its predecessor is set to the predecessor of the neighbour configuration). Afterwards 'open list' is sorted. Thus the sequence of nodes of the planned path always changes during the procedure to the benefit of lower cost paths. This behaviour depicts Fig. 3: the path including the node q_{n+1} related to the light grey arrows is updated to a shorter path with lower costs (from q_n to q_{n+2}, black arrow) during the search. After every iteration step the node at the first place in 'open list' (i.e. the node with minimum costs) is chosen for further exploration (cf. q_C in Fig. 2), deleted from 'open list' and inserted in a list called 'closed list'. After deleting a node 'open list' has to be sorted. When the search is finished the resultant path can be achieved by backtracking the predecessor nodes in 'closed list', starting at the goal configuration.

2.3. Choosing adjacent Configurations

One essential aspect of the search procedure is the choice of neighbour configurations. With respect to the fact that the restrictions of the system, esp. maximum steering angle, have to be satisfied, neighbour configurations are calculated using the kinematic model of the system. In every search step a bunch of control inputs (v, $\dot{\phi}_1$, $\dot{\phi}_2$) is chosen. The trajectories of the state variables (x, y, ϕ_1, θ_1, ϕ_2, θ_2) of system (4) are computed with numerical integration, starting at the current configuration q_C, cf. Fig. 2. After every integration step a collision test for the determined configuration is done. If a collision is detected the dedicated node is discarded. A grid map of the environment is available for collision testing. Note, that within this algorithm configurations are not restricted to the resolution of the grid map, instead they can lie everywhere (as illustrated in Fig. 3). The grid is relevant only for the collision test procedure. For a given configuration collision is tested for the nearest points in the grid map. To rule out the possibility of a not detected collision due to the fact that the configuration does not lie on the grid an additional safety distance is defined.

2.4. Choosing cost functions

To decide which node is chosen next for further expansion of the search tree, nodes are evaluated by several cost functions. The performance of the algorithm is highly dependent on the choice of this functions, i.e. an inadequate cost function leads to the expansion of much more nodes.
The classical A*-Algorithm was developed for point-like robots. Two cost functions where used. The first is the cost for the length of the path that has been travelled yet, the second is the distance that remains to the goal configuration. For point-like robots this cost can be estimated (ignoring obstacles) by the euclidean distance to the goal configuration. This principle can be assigned to nonholonomic systems.
But for car-like vehicles the orientation has to be considered for an estimation of the distance to the goal. For a car an analytical solution for the shortest path between any two configurations is given in [12]. For a multi-body vehicle no such approach is known. The presented algorithm suggests a weighted euclidean distance between the actual configuration q and the goal configuration q_g as an estimate:

$$C_{goal}(q) = \Big((x_g - x)^2 + (y_g - y)^2 + (w_1(\theta_{1_g} - \theta_1))^2 + (w_2(\theta_{2_g} - \theta_2))^2 \Big)^{\frac{1}{2}} \tag{5}$$

$$\text{with} \quad w1 = \frac{L_1}{\tan(\phi_{1_{max}})}, \quad w2 = \frac{L_2}{\tan(\phi_{2_{max}})} \tag{6}$$

For the travelled distance the path length of the midpoint of the truck's rear axle has been chosen. It is calculated during numeric integration. Other choices for this cost could be sums of the path length of several steered axles or the required manoeuvring area.

From a practical point of view it seems convenient to take additional cost functions into account to improve the path shape. Further criteria leading to cost functions can be the minimisation of:

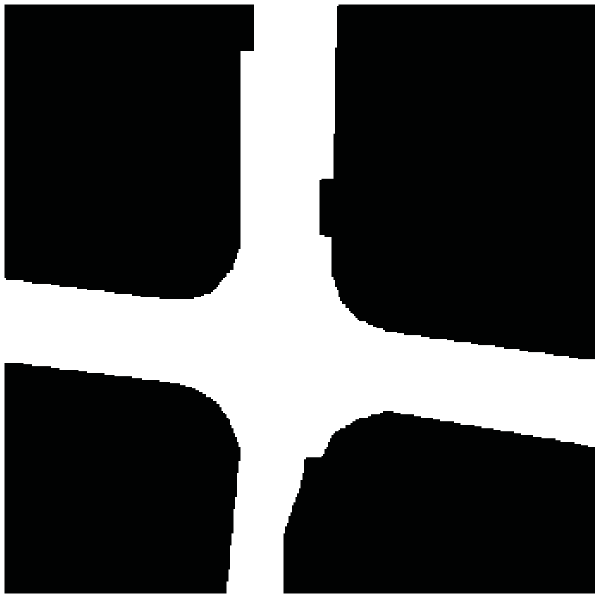
- path length
- required manoeuvring area
- added steering angle
- steering power
- driving power
- distance to obstacles
- number of cusps
- ...

A substantial analysis of the influence of several additional criteria and combinations of them is under investigation.

Table 2. Control parameters as input for numerical integration

v	$\dot{\phi}_1$	$\dot{\phi}_2$
$+v_{max}$	$+\dot{\phi}_{1_{max}}$	$+\dot{\phi}_{2_{max}}$
$+v_{max}$	$+\dot{\phi}_{1_{max}}$	0
$+v_{max}$	$+\dot{\phi}_{1_{max}}$	$-\dot{\phi}_{2_{max}}$
$+v_{max}$	0	$+\dot{\phi}_{2_{max}}$
$+v_{max}$	0	0
$+v_{max}$	0	$-\dot{\phi}_{2_{max}}$
$+v_{max}$	$-\dot{\phi}_{1_{max}}$	$+\dot{\phi}_{2_{max}}$
$+v_{max}$	$-\dot{\phi}_{1_{max}}$	0
$+v_{max}$	$-\dot{\phi}_{1_{max}}$	$-\dot{\phi}_{2_{max}}$
0	$+\dot{\phi}_{1_{max}}$	$+\dot{\phi}_{2_{max}}$
0	$+\dot{\phi}_{1_{max}}$	0
0	$+\dot{\phi}_{1_{max}}$	$-\dot{\phi}_{2_{max}}$
0	0	$+\dot{\phi}_{2_{max}}$
0	0	$-\dot{\phi}_{2_{max}}$
0	$-\dot{\phi}_{1_{max}}$	$+\dot{\phi}_{2_{max}}$
0	$-\dot{\phi}_{1_{max}}$	0
0	$-\dot{\phi}_{1_{max}}$	$-\dot{\phi}_{2_{max}}$
$-v_{max}$	$+\dot{\phi}_{1_{max}}$	$+\dot{\phi}_{2_{max}}$
$-v_{max}$	$+\dot{\phi}_{1_{max}}$	0
$-v_{max}$	$+\dot{\phi}_{1_{max}}$	$-\dot{\phi}_{2_{max}}$
$-v_{max}$	0	$+\dot{\phi}_{2_{max}}$
$-v_{max}$	0	0
$-v_{max}$	0	$-\dot{\phi}_{2_{max}}$
$-v_{max}$	$-\dot{\phi}_{1_{max}}$	$+\dot{\phi}_{2_{max}}$
$-v_{max}$	$-\dot{\phi}_{1_{max}}$	0
$-v_{max}$	$-\dot{\phi}_{1_{max}}$	$-\dot{\phi}_{2_{max}}$

(a) Labelled picture of an intersection (Google Maps)



(b) Binary obstacle map

Fig. 4. Results: Creation of a binary obstacle map

3. RESULTS

Routes for large heavy load transports are often planned also using common tools like Google Maps. With the information of those mapping tools difficult scenarios for the transport can be identified in the planning process. For simulation an image of about $50\ \mathrm{m} \times 50\ \mathrm{m}$ of a Google Maps picture was used. Passable areas and obstacles where labelled manually on the extracted picture (an intersection in a small village). Out of this segmentation a binary obstacle map for the path planning algorithm was calculated, see Fig. 4.
For the map a resolution of $0.1\ \mathrm{m}$ and a prediction horizon of $2.0\ \mathrm{m}$ (Fig. 5(a)) resp. $4.0\ \mathrm{m}$ (Fig. 5(b)) in the numeric integration has been used. Near the goal configuration the resolution is refined three times with path sections of $1.0\ \mathrm{m}$ in each search step. As control parameters for the calculation of adjacent configurations constant values where chosen, see Table 2. In addition to the costs for the travelled distance and the estimate $C_{goal}(q)$ a minimum added steering angle is required:

$$C_\phi(t) = w_3 \int_0^t |\phi_1(\tau)| + |\phi_2(\tau)|\ d\tau \qquad (7)$$

For the simulation the following values where implemented:

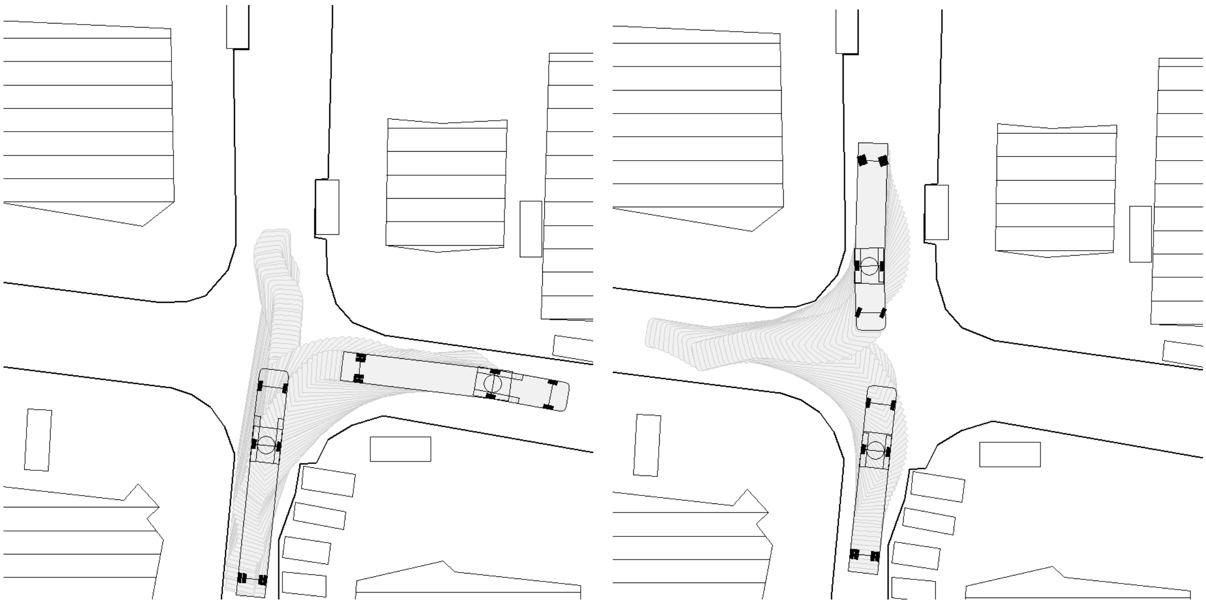
$$\begin{aligned} v_{max} &= 0.2\ \frac{\mathrm{m}}{\mathrm{s}} \\ \dot{\phi}_{1/2_{max}} &= 2.0 \frac{\pi}{180}\ \frac{\mathrm{rad}}{\mathrm{s}} \\ \dot{\phi}_{1/2_{max}} &= 4.0 \frac{\pi}{180}\ \frac{\mathrm{rad}}{\mathrm{s}} \quad \text{(near the goal)} \\ w_3 &= 0.1 \end{aligned} \qquad (8)$$

Fig. 5 depicts the simulation results for a turning-right resp. a reversing manoeuvre. The path planning algorithm calculates individual paths for both steering axles. By admitting $v = 0$ for the variation of control parameters in a search step it is possible for the vehicle to steer while it is standing. With this feature the manoeuvring effort can be reduced significantly. The required manoeuvring area is marked grey. Fig. 5 shows, that the vehicle remains inside the allowed area.

4. CONCLUSION AND FUTURE WORK

It could be shown successfully, that path planning in combination with flexible multi-axle steering can improve planning and manoeuvring for large heavy vehicles. For the first time an algorithm provides individual paths for each steering axle, ensuring a collision free passing. The algorithm is made up in a modular manner, so that it can be easily expanded for other vehicle configurations (more steering axles, more bodies, more pivot points).
Enhancements could be done regarding the following items:

- adapting map resolution as well as prediction horizon to the location of the obstacles
- introducing additional criteria to exclude nodes and speed up the search process
- accomplishing an extensive evaluation of several combinations of cost criteria (cf. subsection 2.4)
- analysing the difference between the use of a kinematic and a kinetic model



(a) Turning-right manoeuvre (b) Reversing manoeuvre

Fig. 5. Results: Manoeuvring in a narrow intersection

- ○ arriving at the goal configuration as exact as possible (currently this depends on the grid map resolution)

5. REFERENCES

[1] Continental Automotive GmbH, "Einparkhalbautomat," 2008, Patent application No. DE 10 2007 009 745 A1.

[2] DaimlerChrysler AG, "Steuerungssystem für ein Gespann," 2005, Patent application No. DE 10 2004 009 187 A1.

[3] DaimlerChrysler AG, "Rückwärtsfahrhilfesystem und Verfahren zur Unterstützung des Fahrers eines Zugfahrzeug-Anhänger-Gespanns bei einer Rückwärtsfahrt," 2007, Patent application No. DE 10 2005 043 466 A1.

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

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

[6] S. Sekhavat, P. Svestka, J.P. Laumond, and M.H. Overmars, "Multi-level 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, M. Valesco-Villa, "Artificial potential fields for trailer-like systems," in *10th Latimamerican Congress on Automatic Control Guadalajara*, 2002.

[8] A. Stopp R. Stahn, T. Stark, "Laser scanner-based navigation and motion planning for truck-trailer combinations," in *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 optimization for airbus A380 component transportation," *IEEE Robotics & Automation Magazine*, vol. 12, pp. 12–21, 2005.

[10] S. Zipser S. Beyersdorfer, S. Wagner, "Optimale Hindernisumfahrung mit mehrachsgelenkten Fahrzeugen," in *conference proceedings CD: 10. Wissenschaftstage der Hochschule Lausitz*, 2010.

[11] 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.

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