Real-time motion planning for an autonomous mobile robot with wheelground adhesion constraint

Jiuchun Gao, Fabien Claveau, Anatol Pashkevich, Philippe Chevrel, Ramdane Abdessamed

Real-time motion planning for an autonomous mobile robot with wheel-ground adhesion constraint

Jiuchun Gao*, Fabien Claveau, Anatol Pashkevich, Philippe Chevrel

The Department of Automation, Production and Computer Sciences, IMT-Atlantique, Nantes, France

The paper proposes a new real-time motion planning technique for an autonomous mobile robot that is able to find a collision-free path and a corresponding time-optimal trajectory while taking into account limits on the actuator capacities and constraints from the wheel-ground adhesion. The proposed technique includes two sub-modules. The first one, an optimal path planner, is based on the discretization of the robot workspace and dynamic programming principle. It allows finding the shortest path in the robot environment to avoid obstacles and reach the robot target. The second one, an optimal trajectory generator, operates with the discretized robot state-space and also employs dynamic programming techniques. It produces a time-optimal motion along the obtained path, which may include numerous regular and singular trajectory sections caused by the simultaneous application of actuator and wheel-ground adhesion constraints. To adapt this technique to real-time implementation, a moving window strategy is presented that allows regularly updating the robot motion profile in a dynamic environment. The advantages of the developed technique and its suitability for real-time control are illustrated by experimental studies implemented on a car-like mobile robot.

Keywords: autonomous mobile robot; real-time motion planning; wheel-ground adhesion; time-optimal trajectory; dynamic programming

1. Introduction

At present, the autonomous mobile robot market is exploding, such types of robots are continuously improved and already widely used in numerous fields like aerospace, military, goods delivery, and medical services [1]. Their further development focuses on providing a higher degree of autonomy and greater mobility/speed, which motivates new developments in mobile robot motion planning and control because usual classical techniques can be hardly applied if the motion speed exceeds some physical limits. One of perspective research direction is related to the wheel-ground interaction whose modelling normally assumes rather high adhesion and excludes such phenomena as excessive slipping and skidding. Similar topics are also significant for self-driving cars whose mechanics is very close to four-wheel mobile robots.

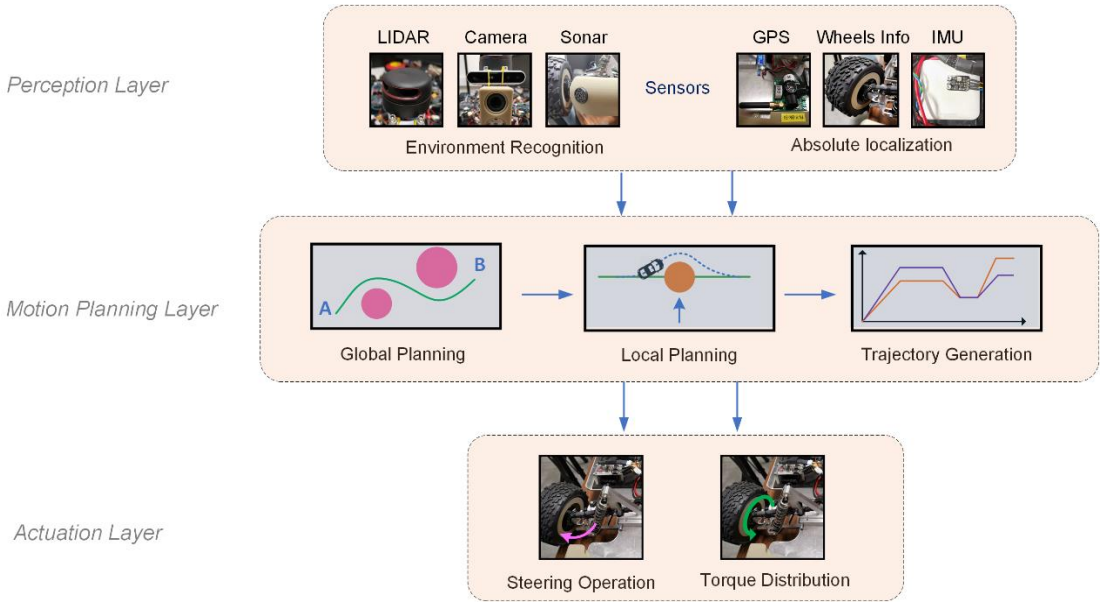

Figure 1 General schematic for autonomous mobile robot control

Generally, control of an autonomous mobile robot contains three basic layers as shown in Fig. 1, the main purpose of motion planning module is to provide the mobile robot with a proper and collision-free trajectory towards its destination while considering the

constraints from the mobile robot kinematics/dynamics, manoeuvre capabilities in the presence of obstacles, and the environment boundaries. Motion planning for autonomous mobile robots can be divided into three hierarchical classes [2]: (i) global planning, (ii) local planning, and (iii) trajectory generation. Global planning is more like routing, which is concerned with selecting the best global path from an origin to a destination. The local planning is a locally correcting phase that handles the dynamic obstacle avoidance while taking into account the vehicle kinematics, obstacles, road geometry, and traffic interactions in the case of the self-driving cars. And finally, the trajectory generation outputs a time-velocity profile corresponding to the previously obtained geometric path while considering some constraints from the robot kinematics and dynamics. In most of the related works, the main attention is paid to the path planning while the velocity profile generation, especially with the minimum-time objective, received less attention.

In literature, graph-based techniques are widely used to generate a global collision-free path in robot workspace. It presents the robot workspace in the form of a 2D graph, which allows to present the desired robot motion as a shortest path on the graph connecting the initial and the final vertices. The most representative algorithms are Dijkstra and A*. Together with a PID-based Path Follow module, they were adopted in [3, 4] for the DARPA Urban Challenge 2007. Another known technique of this group, state lattice algorithm uses a discrete representation of the planning area with a grid of vehicle states [5, 6]. But similar to the above two, the principal limitation of these techniques is the difficulty of taking into account constraints related to the robot kinematics/dynamics and wheel-ground interaction in real-time. Sampling-based approaches consist of randomly sampling the configuration space and searching for feasible connectivity inside it [7]. The most widely used two are the Probabilistic Roadmap Method and the Rapidly-exploring Random Tree [8, 9]. However, this family usually produces non-smooth and jerky paths that should be avoided in time-optimal

control of the autonomous mobile robot [2]. Combinatorial planning techniques use a polyhedral representation of the environment as input and augments it with additional edges or vertices, such as Visibility Graph and Voronoi Diagram [10-13]. But this technique concentrates on geometric aspects mainly. Local path planning techniques are usually applied to avoid collisions in a dynamic environment [14]. The Potential Field Method and its modification Vector Field Histogram are commonly used because of the elegant mathematical analysis and simplicity [15, 16]. Despite numerous advantages, these field-based methods act as a fastest descent optimization procedure, it may get stuck at a local minimum [17]. In contrast to field-based methods, the velocity-space techniques are derived directly from the mobile robot kinematics, allowing taking into account the constraints on the robot maximum velocity/acceleration [18]. A typical example is the Dynamic Window Approach (DWA) [19]. It generates the velocity commands directly, but still with difficulties in taking into account limitations arising from the wheel-ground interactions.

Trajectory generation is for producing the optimal motion along the obtained path. Existing techniques mainly rely on the classical phase-plane methods came from control theory [20] in industrial robotics. For wheeled mobile robots, a time-optimal velocity profile should be produced under dynamic constraints that are due to the motor physical capacities and also must ensure the wheel-ground adhesion [21]. In the frame of the phase-plane method, this adhesion limits can be presented as a non-linear constraint imposed on the velocity and acceleration, which leads to multi-switching of the time-optimal control [22, 23]. Such multi-switching makes difficult for algorithmization in real-time and can be hardly implemented in mobile robot controllers. The wheel-ground adhesion constraint was partially taken into account in [24], where a phase-plane based algorithm was proposed and the maximum centripetal acceleration limit was considered. Another technique was presented in [25], where the cornering trajectory planning algorithm was developed and implemented for a

differential-wheeled mobile robot. It takes into account constraints on the wheel maximum longitudinal force and lateral force in order to avoid sliding, similarly in [26, 27]. However, the longitudinal force and lateral force were considered separately there instead of a resultant one within the friction circle. In [28], a simplified elliptical friction model was considered via limiting the angular acceleration to a predefined maximum value when generating a minimum-time velocity profile for a differential-wheeled mobile robot. It is also worth of mentioning some recent work based on NMPC [29] that allows taking into account the Pacejka's tire model in urban driving conditions, however the high-curvature path and related wheel-ground adhesion issues was out of the authors' attention.

To our knowledge, few works combine global/local path planning with trajectory generation while taking into account the constraints arising from the mobile robot kinematics, dynamics, and wheel-ground interaction that are in the focus of the paper. It is an extension and integration of our previous works [30, 31]. The main contribution of the paper is the development of a new real-time motion planning technique, implemented as two sub-modules. The first of them, the path planner is based on the dynamic programming principle applied to the discretized robot workspace. It allows finding the shortest path in the robot environment with multiple obstacles. Generally, the architecture design of the path planner depends on the application. In particular, for autonomous driving on the road, a hierarchical architecture including global path planning and local path planning is required. Whereas for goods transportation in small spaces, the global and local path planners can be merged into a general one, which is implemented in our experimental mobile robot. The second submodule, the trajectory generator operates with the discretized robot state-space and also employs dynamic programming technique. It produces a time-optimal motion along the obtained path, using the maximum capabilities of the actuators. The obtained fastest motion allows achieving higher efficiency in such applications as goods deliveries. An essential advantage

of this work is that the wheel-ground adhesion condition is directly included in the trajectory generation sub-module. It is worth mentioning that the friction limits of the wheels are widely studied in the automotive field, and precise wheel dynamics models are used in vehicle dynamics control. However, there is no existing work considering this issue in the time-optimal trajectory generation/planning for autonomous mobile robots. It should be also noted that in classical time-optimal control, the velocity and acceleration constraints are usually assumed to be independent yielding rather simple trajectories with two switching only. In contrast here, the velocity and acceleration constraints are coupled and corresponding time-optimal trajectories include multiple switching points that can be hardly generated using conventional phase plane methods. But the developed technique is able to treat even such difficult cases in real-time.

2. The Problem of Real-Time Motion Planning

The problem of real-time motion planning here is divided into two sub-problems, such as (i) path planning and (ii) trajectory generation, as shown in Fig. 2. It should be mentioned, in our case, the mobile robot moves in a quite small indoor area, around 30 m^2, and it is reasonable to merge the global planning and local planning into a general path planner in this work.

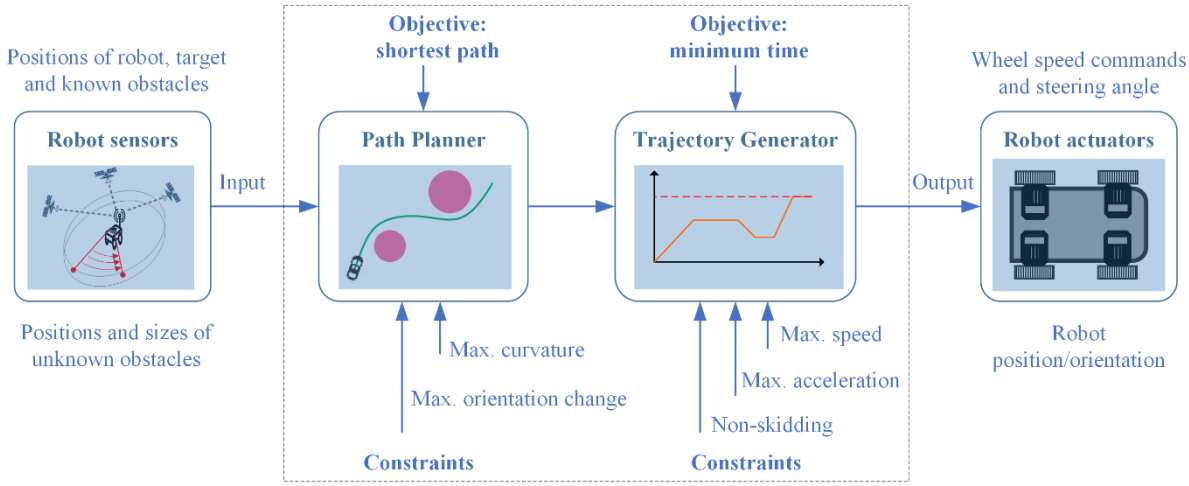

Figure 2 General schema of motion planning for autonomous mobile robot

2.1. Path Planning with Geometric/Kinematic Constraints

It is assumed that the path planner obtains the coordinates of the desired target location and the known obstacle locations from the indoor GPS system, as shown in Fig. 3a. The current position coordinates and orientation angle of the robot ${}^{G}\boldsymbol{p}_0 = ({}^{G}x_0, {}^{G}y_0, {}^{G}\psi_0)$ with respect to the global frame F_G are measured in real-time (see Fig. 3b). Let us describe the desired target location by ${}^{G}\boldsymbol{p}_t = ({}^{G}x_t, {}^{G}y_t, {}^{G}\psi_t)$, where the target orientation ${}^{G}\psi_t$ is assumed to be not constrained in this work. Also, let us denote the center locations of the known obstacles as $\{ {}^{G}\boldsymbol{p}_{obs}^{(k)} = ({}^{G}x_{obs}^{(k)}, {}^{G}y_{obs}^{(k)}) \mid k = 1, 2, \ldots n\}$ where *k* is the obstacle number. For the unknown obstacles, we are using the same notation but assume that the similar coordinates ${}^{R}x_{obs}^{(k)}, {}^{R}y_{obs}^{(k)}$ are measured online by the LiDAR with respect to the robot frame F_R, whose definition is shown in Fig. 4. The obstacles in this work are approximated by circles whose dimensions are slightly modified by adding the equivalent radius of the robot to their original values, as $\{ r_{md}^{(k)} = r_{rob} + r_{obs}^{(k)}; \mid k = 1, 2, \ldots n\}$, where r_{rob} is the equivalent radius of the robot. It should be mentioned that the unknown obstacles are detected by the LiDAR and also roughly approximated as circles for simplicity. Using the data from the LiDAR, a distance between the LiDAR center and the obstacle *d*, the obstacle width *w*, as well as an angle with respect to the LiDAR frame Θ, can be obtained. Thus, an approximated circle with the radius *w*/2 can be located at the detected point whose distance to the LiDAR center is *d*. Although the approximated circle may not completely cover the obstacle, in practice, no collision may happen as such circle and the locally planned path should be updated at each time step.

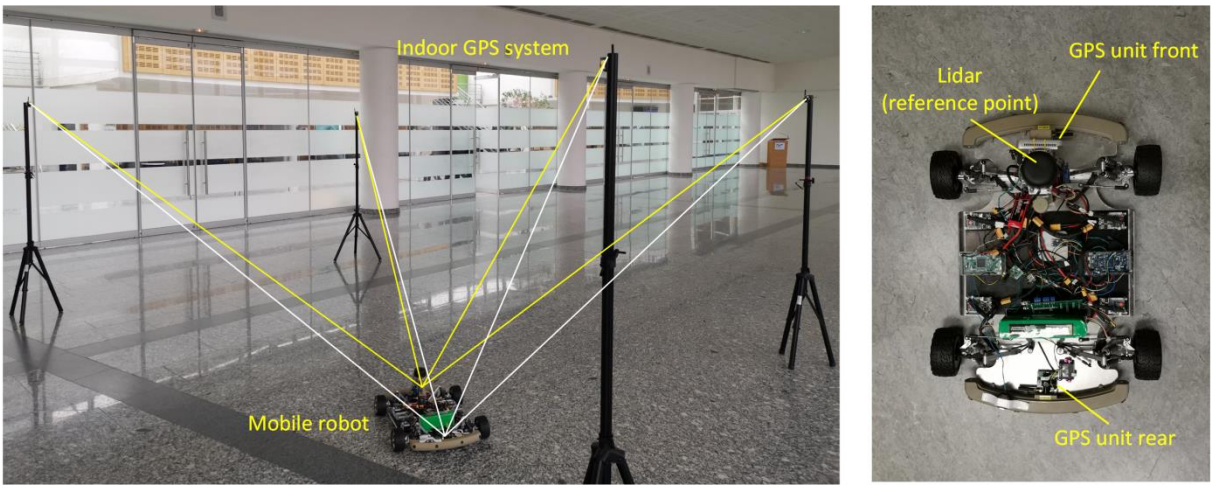

Figure 3 Estimation of robot position and orientation from the sensors (indoor GPS system and LiDAR)

Using the above definitions and notations, the considered path planning task here can be presented as finding a shortest smooth 2-dimensional curve connecting ${}^{G}\boldsymbol{p}_0$ and ${}^{G}\boldsymbol{p}_t$, which satisfies the geometric/kinematic constraints that are presented in detail below. For further convenience, let us assume that the desired path is presented in the parametric form and is described by two continuously differentiable functions

$$\{ {}^{G}x(s),\ {}^{G}y(s)\,;\, s \in [0,\, s_{\max}] \} \tag{1}$$

where s is the distance from the origin to the current state (treated here as the parameter), and the following boundary conditions are satisfied ${}^{G}x(0) = {}^{G}x_0;\ {}^{G}y(0) = {}^{G}y_0$ and ${}^{G}x(s_{\max}) = {}^{G}x_t;\ {}^{G}y(s_{\max}) = {}^{G}y_t$. In addition, let us define the path curvature described by its radius function $\{ c(s), s \in [0, s_{\max}] \}$ which can be easily computed from $x(s)$, $y(s)$ using standard formulas.

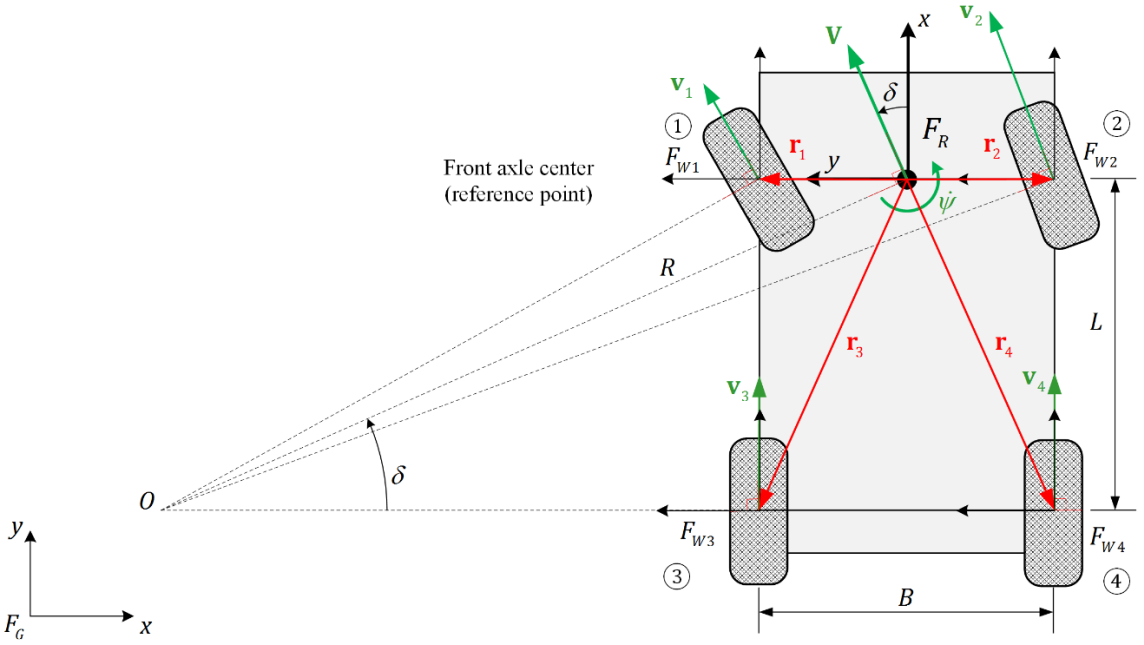

Figure 4 Kinematic structure of the mobile robot and its parameters

The principal constraints imposed on the desired shortest path can be presented in the following way. First, the robot wheel angle δ cannot exceed certain values $[-\delta_{\max}, \delta_{\max}]$, which in accordance with Fig. 4 imposes the following constraints on the path curvature

$$c(s) \le \frac{\sin\delta_{\max}}{L}; \qquad \forall s \in [0, s_{\max}] \tag{2}$$

where L is the wheelbase. Besides, considering the initial orientation of the robot ${}^{G}\psi_0$ cannot be changed instantly, it is necessary to introduce additional boundary condition on the path tangent (robot moving direction) at the starting point, which produces additional constraints of the following form

$$\left.\frac{\mathrm{d}\,{}^{G}y(s)}{\mathrm{d}\,s}\right|_{s=0} = \tan\left({}^{G}\psi_0\right)\cdot\left.\frac{\mathrm{d}\,{}^{G}x(s)}{\mathrm{d}\,s}\right|_{s=0} \tag{3}$$

And finally, the collision constraint can be expressed via the distance from the path to the obstacle center and its increased radius

$$\sqrt{\left[{}^{G}x(s)-{}^{G}x_{obs}^{(k)}\right]^2+\left[{}^{G}y(s)-{}^{G}y_{obs}^{(k)}\right]^2}\geq r_{md}^{(k)};\quad k=1,2,3... \tag{4}$$

Therefore, the path planning problem for the autonomous mobile robot studied here is presented as finding the shortest path $\{{}^{G}x(s);\,{}^{G}y(s)\,|\,s\in[0,s_{\max}]\}$ connecting the current location $({}^{G}x_0,\,{}^{G}y_0,\,{}^{G}\psi_0)$ and the target position $({}^{G}x_t,\,{}^{G}y_t)$, which satisfies the path curvature and collision constraints (2)-(4). In a formal way, this problem can be summarized as follows

$$\begin{cases} \text{find} & \{{}^{G}x(s);\,{}^{G}y(s)\,|\,s\in[0,s_{\max}]\} \\ \text{minimizing} & \displaystyle\int_0^{s_{\max}}\sqrt{\left(\mathrm{d}\,{}^{G}x(s)/\mathrm{d}s\right)^2+\left(\mathrm{d}\,{}^{G}y(s)/\mathrm{d}s\right)^2}\cdot\mathrm{d}s\rightarrow\min \\ \text{s.t.} & \\ & c(s)\leq\sin\delta_{\max}/L \\ & \sqrt{\left[{}^{G}x(s)-{}^{G}x_{obs}^{(k)}\right]^2+\left[{}^{G}y(s)-{}^{G}y_{obs}^{(k)}\right]^2}\geq r_{md}^{(k)} \\ \text{and} & \\ & {}^{G}x(0)={}^{G}x_0;\qquad {}^{G}y(0)={}^{G}y_0 \\ & {}^{G}x(s_{\max})={}^{G}x_t;\quad {}^{G}y(s_{\max})={}^{G}y_t \\ & \mathrm{d}\,{}^{G}y(s)/\mathrm{d}s=\tan({}^{G}\psi_0)\cdot\mathrm{d}\,{}^{G}x(s)/\mathrm{d}s,\ s=0 \end{cases} \tag{5}$$

2.2. Trajectory Generation with Dynamic Constraints

After obtaining a smooth path $\{{}^{G}x(s);\,{}^{G}y(s)\,|\,s\in[0,s_{\max}]\}$ in the Cartesian space, it is necessary to find a time-velocity profile ensuring the fastest motion along this path. This motion is described by the time-displacement function $s(t)$ where $t\in[0,T]$, which minimizes the traveling time objective

$$\int_0^T 1\cdot\mathrm{d}t\rightarrow\min \tag{6}$$

from the initial state $(s=0,\ \left.\frac{\mathrm{d}s}{\mathrm{d}t}\right|_{t=0}=0)$ to the final state $(s=s_{\max},\ \left.\frac{\mathrm{d}s}{\mathrm{d}t}\right|_{t=T}=0)$, and satisfies certain constraints on derivatives

$$\left|\mathrm{d}\,s(t)/\mathrm{d}\,t\right| \le v_{\max};\quad \left|\mathrm{d}^2\,s(t)/\mathrm{d}\,t^2\right| \le a_{\max} \tag{7}$$

that describe the robot's capability to achieve desired velocities and accelerations during the motion. Besides, for fast motions of wheeled robots, there is an additional specific constraint arising from the wheel-ground interaction and excluding the wheel skidding when the robot enters the high curvature path segments with high speed. As known from the vehicle dynamics [32], the adhesion force acting between the wheel and the ground is limited by a certain value that depends on the road condition and should be inside of the so-called friction ellipse/circle, as shown in Fig. 5. The latter can be expressed in the form of the following inequality

$$\sqrt{F_x^2 + F_y^2} \le \mu \cdot F_z \tag{8}$$

where F_x, F_y, F_z are the longitudinal, lateral, and vertical forces respectively, μ is the friction coefficient that is assumed as a constant here. When the robot of mass *m* goes into a cornering manoeuvre with the velocity *v* (see Fig. 5), the lateral force provides the centripetal acceleration $F_y = m \cdot v^2 / R$ where $R = 1/c(s)$ is the curvature radius. However, if the tangent speed is too high, the robot is desired to decelerate slightly, which requires the longitudinal force $F_x = m\,a$. In addition, the vertical force can be computed as $F_z = m\,g$. So, after simplification, a non-skidding condition (8) can be expressed via the derivatives of the function $s(t)$ as follows

$$\left\{\left[\mathrm{d}\,s(t)/\mathrm{d}\,t\right]^2 \cdot c(t)\right\}^2 + \left[\mathrm{d}^2\,s(t)/\mathrm{d}\,t^2\right]^2 \le (\mu g)^2 \tag{9}$$

where $c(t) = c\left[s(t)\right]$ is the path curvature.

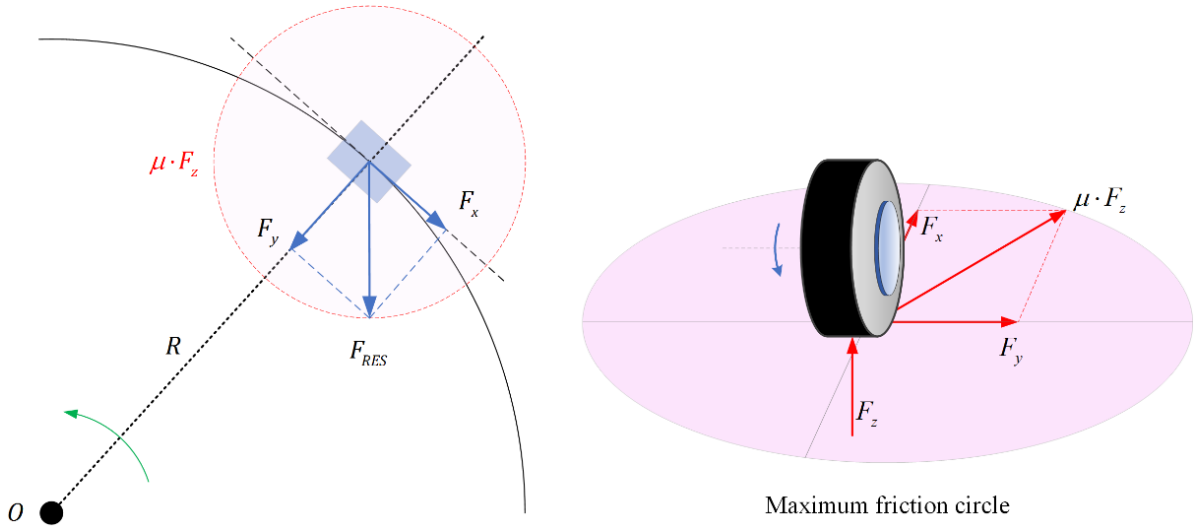

Figure 5 Friction ellipse/circle

It should be mentioned that the non-skidding condition (9) was derived assuming that both the velocity and the acceleration are computed for the robot center of gravity, i.e. the robot is presented as a material point. However, this constraint can be easily generalized by considering the interaction with the ground for all wheels separately. It can be proved that for the four-wheel case, the non-skidding condition produce four constraints with similar structures

$$\left\{\left[\mathrm{d}s_i(t)/\mathrm{d}t\right]^2 \cdot c_i(t)\right\}^2 + \left[\mathrm{d}^2 s_i(t)/\mathrm{d}t^2\right]^2 \le (\mu g)^2; \quad \forall i = 1,2,3,4 \tag{10}$$

where $c_1(t) = 1/\sqrt{(c(t)^{-1} \cdot \cos\delta(t) - 0.5B)^2 + L^2}$, $c_2(t) = 1/\sqrt{(c(t)^{-1} \cdot \cos\delta(t) + 0.5B)^2 + L^2}$,

$c_3(t) = [c(t)^{-1} \cdot \cos\delta(t) - 0.5B]^{-1}$, $c_4(t) = [c(t)^{-1} \cdot \cos\delta(t) + 0.5B]^{-1}$ and $\delta(t) = \arcsin\left(L \cdot c(t)\right)$,

$\mathrm{d}s_i(t)/\mathrm{d}t = \left[\mathrm{d}s(t)/\mathrm{d}t\right]/\left[R \cdot c_i(t)\right]$, $\mathrm{d}^2 s_i(t)/\mathrm{d}t^2 = \left[\mathrm{d}^2 s(t)/\mathrm{d}t^2\right]/\left[R \cdot c_i(t)\right]$.

It should be also noted that for high-speed driving, full dynamics of the vehicle and wheel/tire usually should be considered in the motion control. However in this work, there are several simplifications. First of all, it is assumed that the robot drives on a 2D plane, so the pitch and roll motion are ignorable, and also the friction coefficient is almost constant because

of unchanging road condition. For this reason, for planning the desired time-optimal trajectory under these assumptions, a kinematic vehicle model plus the tire dynamic model should be sufficient. For computational convenience, a simplified tire dynamic model based on TMeasy is used here. According to the generalized tire characteristics presented in [33], there are three modes: 1) adhesion mode where the combined slip ratio $S \in [0, S_{\max}]$, and $F \in [0, F_{\max}]$, $S_{\max}$ corresponds to the maximum friction $F_{\max}$; 2) adhesion/sliding mode where $S \in [S_{\max}, S_{slide}]$ and $F \in [F_{\max}, F_{slide}]$, S_{slide} corresponds to the sliding friction F_{slide}; and 3) sliding mode where $S \in [S_{slide}, 1]$ and $F = F_{slide}$. The adhesion constraint proposed in the paper ensures the resultant tire force F is always within the friction limits, never beyond $F_{\max}$, which leads to the combined slip belonging to $[S_0, S_{\max}]$. Therefore, it is reasonable to use this model in this work.

Finally, taking into account both the boundary conditions and all the constraints (7)-(9), the problem of time-optimal trajectory generation for the given geometric path $\{ {}^{G}x(s); {}^{G}y(s) \,|\, s \in [0, s_{\max}]\}$ can be summarized as

$$
\begin{cases}
\text{find} & \{ s(t); \mathrm{d}s(t)/\mathrm{d}t \;|\; t \in [0, T]\} \\
\text{minimizing} & \displaystyle\int_0^T 1 \cdot \mathrm{d}t \rightarrow \min \\
\text{s.t} & \\
 & \left|\mathrm{d}s(t)/\mathrm{d}t\right| \le v_{\max}; \quad \left|\mathrm{d}^2 s(t)/\mathrm{d}t^2\right| \le a_{\max} \\
 & \left\{\left[\mathrm{d}s(t)/\mathrm{d}t\right]^2 \cdot c(t)\right\}^2 + \left[\mathrm{d}^2 s(t)/\mathrm{d}t^2\right]^2 \le (\mu g)^2 \\
\text{and} & \\
 & s(0) = 0; \qquad \mathrm{d}s(t)/\mathrm{d}t\big|_{t=0} = 0 \\
 & s(T) = s_{\max}; \quad \mathrm{d}s(t)/\mathrm{d}t\big|_{t=T} = 0
\end{cases}
\tag{11}
$$

It is worth mentioning that the optimization problem includes some particularities compared to the classical time-optimal control problem for the second-order system that was intensively studied in the literature [34], where the constraints were applied to the velocity

and acceleration independently. In contrast, in our work, there is a non-linear adhesion constraint (9) where the velocity and acceleration are coupled, which does not allow straightforwardly applying previously developed methods. Relevant techniques for the considered time-optimal trajectory generation in real-time are proposed in the following section.

3. Motion Planning Algorithms and the Real-Time Implementation

In order to solve the above-stated problems, a combinatorial optimization based methodology is proposed in this section. It includes the discretization of the robot workspace (for path planning) and state space (for trajectory generation). Further, to find the desired shortest path and time-optimal trajectory satisfying all static/dynamic constraints, the dynamic programming principle is applied to create relevant motion planning algorithms.

3.1 Shortest Path Search in Discretized Robot Workspace

To present the robot path planning problem in a discrete way, let us use the double presentation of the workspace, via the Cartesian coordinates (x, y) and the polar coordinates (ρ, θ), and discretize the allowable domain of the polar angle $\theta \in [\theta_{\min}, \theta_{\max}]$ and the distance $\rho \in [0, \rho_{\max}]$ with the steps $\Delta\theta$ and $\Delta\rho$ respectively

$$\begin{aligned} \theta_k &= k \cdot \Delta\theta; \quad k = 0,1,...m \\ \rho_i &= i \cdot \Delta\rho; \quad i = 0,1,...n \end{aligned} \tag{12}$$

where $\Delta\theta = (\theta_{\max} - \theta_{min})/m$ and $\Delta\rho = \rho_{\max}/n$. It is assumed that the pole of the polar system coincides with the origin of the robot frame F_R and the polar axis is directed along the Cartesian X-axis of F_R. In such settings, it is reasonable to limit the range of the polar coordinate ρ by the distance from the origin to the target point, $\rho_{\max} = \sqrt{{}^R x_t^2 + {}^R y_t^2}$, as shown in Fig. 6. Further, in order to avoid excessive discretization and to reduce the real-time

computational effort, the range of the polar angle $\pm\pi$ should be reduced in the following way:

$$\theta_{\max} = \max\left\{\frac{\pi}{2}, \operatorname{atan2}({}^{R}y_t, {}^{R}x_t)\right\}$$
$$\theta_{\min} = \min\left\{-\frac{\pi}{2}, \operatorname{atan2}({}^{R}y_t, {}^{R}x_t)\right\} \quad (13)$$

where $({}^{R}x_t, {}^{R}y_t)$ is the target location coordinates with respect to the robot frame. However, it is worth mentioning that the proposed technique allows doing a new search with the full range of the polar angle if there is no solution found in the reduced admissible area. This may happen when the numbers and dimensions of the obstacles are quite huge.

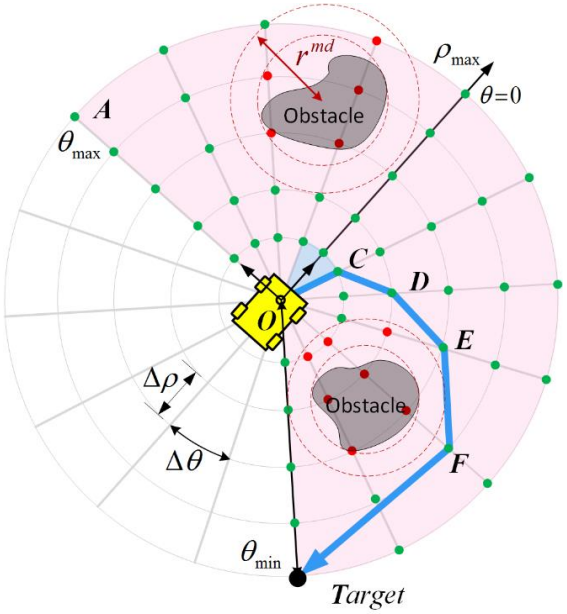

Figure 6 Discretization of robot workspace for path planning: admissible and non-admissible nodes

Such discretization allows us to replace the continuous workspace of the robot by limited numbers of nodes $\mathbf{S}_{k,i}$ that can be described by either the Cartesian coordinates $\{x_{ki}, y_{ki}\}$ or the polar coordinates $\{\rho_i, \theta_k\}$ where $x_{ki} = \rho_i \cdot \cos(\theta_k)$ and $y_{ki} = \rho_i \cdot \sin(\theta_k)$. Further, assuming that the distance between the mobile robot and the target point is

sequentially reducing, the desired optimal path can be presented as the following sequence of the nodes

$$\{\mathbf{S}_{k_1,1}\} \to \{\mathbf{S}_{k_2,2}\} \to ... \to \{\mathbf{S}_{k_{n-1},n-1}\} \to \{\mathbf{S}_{k_n,n}\} \tag{14}$$

where $\{k_i \,|\, i = 1,2,...n\}$ are certain indices obtained from the optimization routine. Corresponding distances between the subsequent nodes can be evaluated via the Cartesian coordinates as

$$\operatorname{dist}(\mathbf{S}_{k_i,i}, \mathbf{S}_{k_{i+1},i+1}) = \left\| \begin{pmatrix} x_{k_i,i} \\ y_{k_i,i} \end{pmatrix} - \begin{pmatrix} x_{k_{i+1},i+1} \\ y_{k_{i+1},i+1} \end{pmatrix} \right\| \tag{15}$$

where $(x_{k_i,i}, y_{k_i,i})$ is the Cartesian coordinates of the i^{th} path node, and $(x_{k_{i+1},i+1}, y_{k_{i+1},i+1})$ is the potential coordinates of the $(i+1)^{\text{th}}$ path node. The latter allows us to present the objective function to be minimized as follows

$$D = \sum_{i=1}^{n-1} \operatorname{dist}(\mathbf{S}_{k_i,i}, \mathbf{S}_{k_{i+1},i+1}) \to \min_{k_i} \tag{16}$$

Further, to take into account the constraint on the path curvature, all subsequences $\mathbf{S}_{k_{i-1},i-1} \to \mathbf{S}_{k_i,i} \to \mathbf{S}_{k_{i+1},i+1}$ should be verified with respect to the inequality (2), which in the frame of the adopted notations can be presented as

$$\operatorname{curv}[(k_{i-1}, i-1), (k_i, i), (k_{i+1}, i+1)] \le L^{-1} \cdot \sin\delta_{\max} \tag{17}$$

where the curvature function $\operatorname{curv}[\cdot,\cdot,\cdot]$ is easily obtained by the arc fitting of the following three points $\{x_{k_{i-1},i-1}, y_{k_{i-1},i-1}\}, \{x_{k_i,i}, y_{k_i,i}\}, \{x_{k_{i+1},i+1}, y_{k_{i+1},i+1}\}$. For computational convenience, the path curvature at the node $\mathbf{S}_{k_i,i}$ can be also expressed via the robot orientations $\psi_{k_i,i}$, $\psi_{k_{i+1},i+1}$ at the segments $[\mathbf{S}_{k_{i-1},i-1} \to \mathbf{S}_{k_i,i}]$ and $[\mathbf{S}_{k_i,i} \to \mathbf{S}_{k_{i+1},i+1}]$ respectively, which yields the revised approximated expression for the constraint (2)

$$\left| \psi_{k_{i+1},i+1} - \psi_{k_i,i} \right| \le \operatorname{dist}(\mathbf{S}_{k_i,i}, \mathbf{S}_{k_{i+1},i+1}) \cdot L^{-1} \cdot \sin\delta_{\max} \tag{18}$$

where $\psi_{k_i,i} = \operatorname{atan}2(y_{k_i,i} - y_{k_{i-1},i-1},\, x_{k_i,i} - x_{k_{i-1},i-1})$; $\psi_{k_{i+1},i+1} = \operatorname{atan}2(y_{k_{i+1},i+1} - y_{k_i,i},\, x_{k_{i+1},i+1} - x_{k_i,i})$. The collision constraints can be taken into account via the verification of the validity of the inequality

$$\operatorname{dist}(\mathbf{S}_{k_i,i}, \hat{\mathbf{S}}_j) \geq r_j^{md} \tag{19}$$

where $\hat{\mathbf{S}}_j$ represents the center of the j^{th} obstacle and r_j^{md} is the modified obstacle radius.

Thus, the original robot path planning problem is converted to a combinatorial one, dealing with the shortest path search on the graph. In contrast to classical formulations, there are specific constraints here (feasible curvature and collision-free) applied to the graph nodes and triples of subsequent nodes, which make difficulties in applying known graph-based approaches. For this reason, a dynamic programming (DP) technique is used here to find the desired optimal path that satisfies all constraints imposed by the environment and geometry.

The developed DP-based algorithm breaks down the full-size problem presented above into a set of sub-problems, aiming at finding the shortest path from the initial node $\mathbf{S}_{0,0}$ to the arbitrary node $\{\mathbf{S}_{k_i,i}, \forall k_i\}$ belonging to the i^{th} layer with the polar coordinate $\rho_i = \Delta\rho \cdot i$. To present the basic idea of this algorithm, let us denote $d_{k,i}^*$ as the length of the shortest path connecting the initial node $\mathbf{S}_{0,0}$ to the current node $\mathbf{S}_{k_i,i}$. Then, taking into account the additivity of the objective (16), the shortest path for the nodes belongs to the next layer $\{\mathbf{S}_{k,i+1}, \forall k\}$ can be found by combining the optimal solutions for the previous layer $\{\mathbf{S}_{p,i}, \forall p\}$ and the distances between the nodes belonging to the layers i and $i+1$. The latter yields the formula

$$d_{k,i+1}^* = \min_p \left\{ d_{p,i}^* + \operatorname{dist}(\mathbf{S}_{p,i}, \mathbf{S}_{k,i+1}) \right\} \tag{20}$$

that is applied sequentially starting from the second layer, i.e. $i = 1, 2, \ldots n-1$. Finally, after the selection of the minimum objective $d^*_{k,i+1}$ of the nodes belonging to the final layer and applying the backtracking, one can get the desired optimal path in the considered graph. It is described by the recorded indices $\{k_1, k_2, k_3, \ldots k_n\}$. It should be noted that the fusion of the optimal path $[\mathbf{S}_{0,0} \rightarrow \ldots \rightarrow \mathbf{S}_{k_i,i}]^{opt}$ and a new segment $[\mathbf{S}_{k_i,i} \rightarrow \mathbf{S}_{k_{i+1},i+1}]$ is allowed if and only if all the above-mentioned constraints are satisfied, which are verified using the node subsequence $[\mathbf{S}_{k_{i-1},i-1} \rightarrow \mathbf{S}_{k_i,i} \rightarrow \mathbf{S}_{k_{i+1},i+1}]$ extracted from the considered path. As the DP principle is applied to both path planning and trajectory generation, a more detailed explanation and pseudo-code for DP implementation are given in Subsection 3.2.

3.2 Time-Optimal Trajectory Generation in Discretized Robot State Space

For the problem of trajectory generation with time-optimal control structure, as discussed in Subsection 2.2, the conventional phase plane technique has the difficulty in including the non-linear wheel-ground adhesion constraint (9). An alternative approach was first proposed in our previous work [30, 31], based on the discrete dynamic programming (DP). Its main advantage is the universality allowing to take into account all considered constraints in a similar way and to generate optimal trajectories of complex structure including multiple switching points. In this paper, an extended and more complete DP-based algorithm is presented.

First, let us apply sampling to the allowable domain of the velocity $v \in [0, v_{\max}]$ and displacement $s \in [0, s_{\max}]$ with the steps Δv and Δs respectively

$$\begin{aligned} v_k &= k \cdot \Delta v; \qquad k = 0,1,\ldots m \\ s_i &= i \cdot \Delta s; \qquad i = 0,1,\ldots n \end{aligned} \tag{21}$$

where $\Delta v = v_{\max}/m$ and $\Delta s = s_{\max}/n$. Further, for each path point s_i it is possible to generate a number of the possible states (v_k, s_i) that differ in velocities, i.e. to produce a mapping from the geometric path to the robot state space (v, s) allowing to describe motion in time

$$s_i \to \{\mathbf{C}_{k,i} = (v_k, s_i) \,|\, k = 0,1,...m\} \tag{22}$$

Taking into account that the path points are naturally ordered in time as $s_0 \to s_1 \to ... \to s_n$, the search space including all possible trajectories corresponding to the given path can be presented as a directed graph shown in Fig. 7. In particular, it is clear that the robot velocity at the starting and end points is known, so only a single state $\mathbf{C}_{k_0,0}$ should be used to represent the starting point and a single state $\mathbf{C}_{0,n}$ for the target point, see the graph in Fig. 7. Another particularity of this graph caused by the time-irreversibility is that the allowable connections between the nodes are limited to the subsequent configuration states $\mathbf{C}_{k_i,i} \to \mathbf{C}_{k_{i+1},i+1}$, while the edge weights correspond to the traveling time defined below.

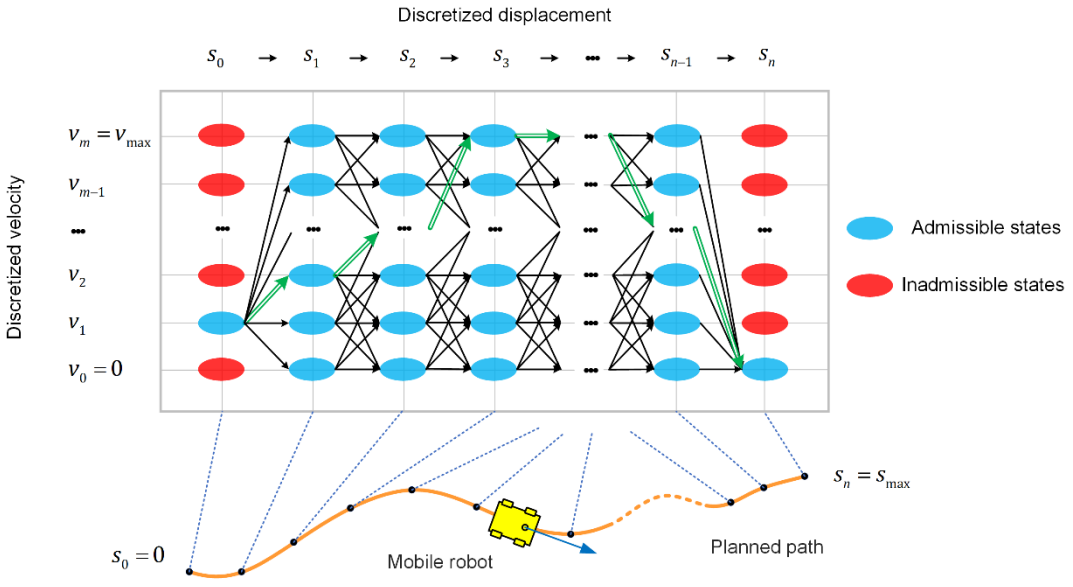

Figure 7 Graph-based presentation of the discretized state space for optimal motion generation along a given path

Using the discrete search space, the considered problem can be transformed to the searching of the shortest connection on the above presented directed graph, which can be represented as the following node sequence

$$\mathbf{C}_{k_0,0} \to \mathbf{C}_{k_1,1} \to ... \to \mathbf{C}_{0,n} \tag{23}$$

where the distance between subsequent nodes is corresponding to the motion time and is computed in the following way

$$\text{time}(\mathbf{C}_{k_i,i}, \mathbf{C}_{k_{i+1},i+1}) = 2(s_{i+1} - s_i)/(v_{k_{i+1}} + v_{k_i}) \tag{24}$$

The latter allows us to present the objective function to be minimized as follows

$$T = \sum_{i=0}^{n-1} \text{time}(\mathbf{C}_{k_i,i}, \mathbf{C}_{k_{i+1},i+1}) \to \min \tag{25}$$

It is clear that the method of state-space discretization (21) applied above automatically takes into account the velocity constraint in (7), but the relevant acceleration constraints $|a_i| \le a_{\max}; \forall i = 1,2,...n-1$ must be examined for each candidate subsequence $\mathbf{C}_{k_i,i} \to \mathbf{C}_{k_{i+1},i+1}$, using an approximated expression

$$a_i = (v_{i+1} - v_i)/\Delta t_i \tag{26}$$

where $\Delta t_i = \text{time}(\mathbf{C}_{k_i,i}, \mathbf{C}_{k_{i+1},i+1})$. In addition, the non-skidding condition (9) should be also verified using the following inequality

$$\sqrt{a_i^2 + (v_i^2/R_i)^2} \le \mu \cdot g \tag{27}$$

where $R_i = 1/c_i$ is the radius of path curvature corresponding to the i^{th} sampling point. Hence, the original trajectory generation problem is converted to a combinatorial one, which allows taking into account the wheel-ground adhesion limits and the constraints applied to the node subsequences that limit the robot acceleration.

The DP principle is adopted here too, breaking down the full-size problem into a set of sub-problems and aiming at finding all optimal sequences from the initial node $\mathbf{C}_{k_0,0}$ to the current ones $\{\mathbf{C}_{k_i,i}, \forall k_i\}$ ensuring the minimum of the motion time (25) for which the accelerations computed via the subsequences $\{\mathbf{C}_{k_i,i} \rightarrow \mathbf{C}_{k_{i+1},i+1} \,|\, \forall i = 0,1,...n-1\}$ do not exceed allowable limits.

Let us denote $\tau_{p,i}$ as the length of the optimal connection on the state-space graph from $\mathbf{C}_{k_0,0}$ to $\mathbf{C}_{p,i}$. Then, the optimal connection for the nodes belonging to the next layer $\{\mathbf{C}_{k,i+1}, \forall k\}$ can be found by combining the optimal solutions for the previous layer $\{\mathbf{C}_{p,i}, \forall p\}$ and the distances between the nodes with the indices *i* and *i*+1. The latter corresponds to the recursive formula

$$\tau_{k,i+1} = \min_{p} \left\{ \tau_{p,i} + \text{time}(\mathbf{C}_{p,i}, \mathbf{C}_{k,i+1}) \right\} \tag{28}$$

where the index *p* must satisfy the following condition

$$p \in \left\{ \left| \text{accel}(\mathbf{C}_{p,i}, \mathbf{C}_{k,i+1}) \right| \le \min \left[a_{\max}, \sqrt{(\mu \cdot g)^2 - (v_p^2 / R_i)^2} \right] \right\} \tag{29}$$

in which the acceleration $a_i = \text{accel}(\mathbf{C}_{p,i}, \mathbf{C}_{k,i+1})$ is computed using expression (26) and the velocity $v_p = p \cdot \Delta v$ corresponds to the node $\mathbf{C}_{p,i}$. This formula is applied sequentially starting from the second layer producing the matrix of the motion times $[\boldsymbol{\tau}_{k,i}]$. When this matrix is created, the minimum value $\tau_n^{\min}$ is selected from the last column corresponding to the robot time-optimal motion from the initial state to the target point. Finally, by applying the backtracking, one can get the desired optimal connection in the state-space graph. It is described by relevant indices $\{k_1, k_2, ... k_{n-1}, k_n\}$.

Algorithm: Time-optimal trajectory generation along the path

***Input*:**	Matrix of states – C(k,i) of size m×n
	Array of radius – R(i) of size 1×n
***Output*:**	Minimum path length – D_{min}
	Optimal path indices – k^0(i), i = 1,2,…n
***Notations*:**	Distance matrix – *D*(k,i) of size m×n
	Pointer matrix – *P*(j,i-1) of size m×n
***Functions*:**	Distance between nodes – *dist* ($C(k_1,i_1)$, $C(k_2,i_2)$)
	Adhesion test for a node – *adhs* (C(k,i), R(i))
	Acceleration test for nodes – *acc* ($C(k_1,i_1)$, $C(k_2,i_2)$)

(1) **Set** D(k,1):=0; P(k,1):=null; ∀k=1,2,…m
(2) **For** i = 2 to n **do**
  For k = 1 to m **do**
    For j = 1 to m **do**
  (a) **If** (*acc* (C(k,i), C(j,i-1)) = 0) &
    (*adhs* (C(k,i), R(i))=0) & (*adhs* (C(j,i-1), R(i-1))=0)
      r(j) := *D*(j,i-1)+*dist* (C(k,i), C(j,i-1))
    else
      r(j) := Inf
    end
  (b) **Set** *D*(j,i) := *min*(r); *P*(j,i) := *argmin*(r);
(3) **Set** D_{min} := *min*(*D*(k,n)); k^0(n) := *argmin*(r)
(4) **For** i = 2 to n **do**
    Set k^0(i-1) := P(k^0(i), i)

In more detail, an outline of the developed algorithm is presented below in the form of pseudo-code. The input includes the state matrix $\{C(k,i)\,|\,k=0,1,...m;\, i=0,1,...n\}$ containing information on the velocity and displacement, as well as the array of path curvatures $\{R(i)\,|\,i=0,1,...n\}$. The algorithm operates with two tables $D(k,i)$ and $P(k,i)$ that include the minimum distances for the sub-problem of lower size (for the path $\mathbf{C}_{k_0,0} \to ... \mathbf{C}_{k_i,i}$) and the pointers to the previous state $C(k_{i-1}, i-1)$ respectively. The procedure is composed of four basic steps. The first step (1) initializes the distance and pointer matrices. In step (2), the recursive formula (28) is implemented. The computing starts from the second layer, and it tries all possible connections between the states in the current layer and the previous one while verifying the wheel-ground adhesion and acceleration limits in the sub-step (2a). The

sub-step (2b) finds the optimal trajectories from all states $\{C(k,i),\forall k\}$ belonging to the i^{th} layer and records the references to $\{C(j,i-1),\forall j\}$ into the pointer matrix $P(k,i)$. In steps (3) and (4), the optimal solution is finally obtained and the corresponding path is extracted by means of backtracking.

3.3 Real-time Implementation of Developed Algorithms

For real-time implementation, a 'moving window' strategy was developed. This developed strategy breaks down the full-size problem into a set of lower dimension sub-problems corresponding to shorter displacements. As is shown in Fig. 8, a moving window is applied segment by segment. For each window, the developed DP-based algorithm from Subsection 3.2 is used to generate a trajectory segment, i.e. a local time-optimal motion from the current state to some intermediate (goal) states with maximum velocity and some ahead displacement. This procedure is periodically repeated while the current state is approaching the target one. It is implemented in real-time, generating each new segment slightly in advance while the controller is carrying out the motion profile from the previous window. The latter allows us to execute the motion generation and implementation simultaneously.

4. Experimental Validation

The proposed methodology was validated via the experimental studies dealing with a mobile robot in an indoor environment, around 30 m^2. The following subsections present some details concerning the robot technical parameters.

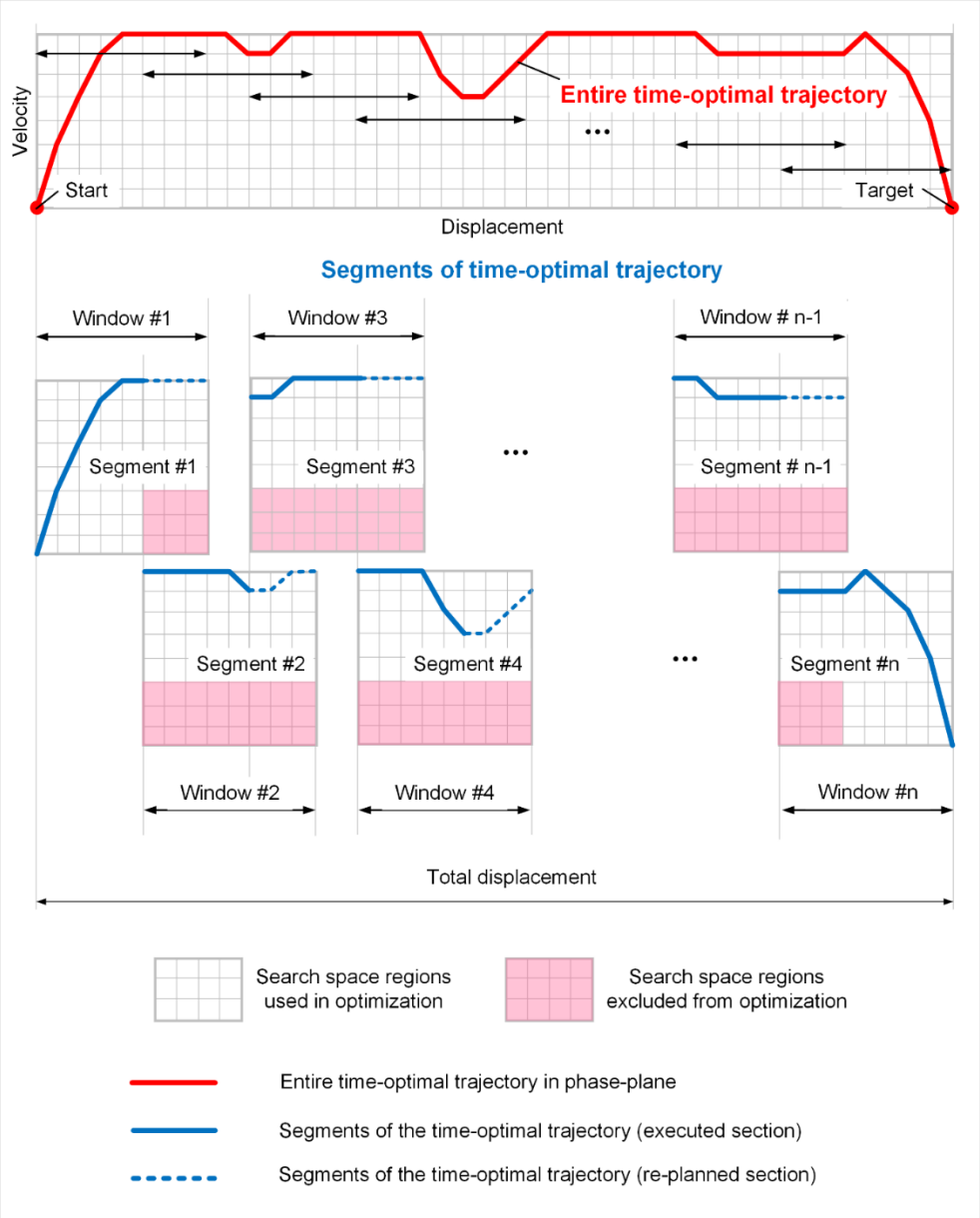

Figure 8 Multi-segment strategy of real-time optimal motion generation for the mobile robot

4.1 Robotic Platform and Experimental Setup

The mobile robot (see Fig. 3) is a car-like system with four-wheel driving and front steering, whose wheelbase is 0.450 m, and the length of the front/rear track is 0.482 m. The robot is localized by means of indoor GPS. Odometer, available in each in-wheel motors equipped with build-in hall-effect sensors, is used to measure and regulate the velocity of the wheels only. A LiDAR makes the environment scanning for local vision construction. More details concerning the sensors are given in Table. 1.

Table 1 Robot sensors, their parameters, and locations

	Sensor parameters
Indoor GPS Marvel-Mind solution	Accuracy ±2cm Detection area ~30 m^2 Frequency 4Hz (average value)
LiDAR scanner RPLIDAR A2	Range 0.5-6 m Accuracy: 0.9º resolution (200 points for a 180º arc) Frequency: 2000 Hz
Odometer	14-pole motor

The robot is actuated using four in-wheel motors (velocity control) and a steering motor (position control). The dynamic capacities and specifications of the motors are presented in Table. 2.

Table 2 Motors parameters and control modes

	Basic parameters
Servo motor Savox SB-2290SG	Maximum speed: 8.04 $rad \cdot s^{-1}$ Rotating range: [-33°,+33°]
In-wheel motors MTO7052-HBM-60-HA	Maximum speed: 23.92 $rad \cdot s^{-1}$ (3200 ERPM)

The computation layer is based on an Intel Atom processor N270 (1.6 GHz 1 GB DDR2 533 MHz RAM). The proposed algorithms were created on an external PC first and then downloaded to the robot CPU via Simulink Real Time 2017. This hardware configuration ran with the sampling time of 0.20 sec and utilizes the discrete solver from

Simulink. It should be mentioned that the sampling time used here is rather large to simplify the experiment monitoring but for future industrial applications, it will be essentially reduced.

The experimental environment is shown in Fig. 9. The robot's initial location and the target position were randomly selected by the experimenters. Within the test area, there is a no-go area (also treated as a static obstacle). A dynamic obstacle (box) was moved manually during the robot navigation.

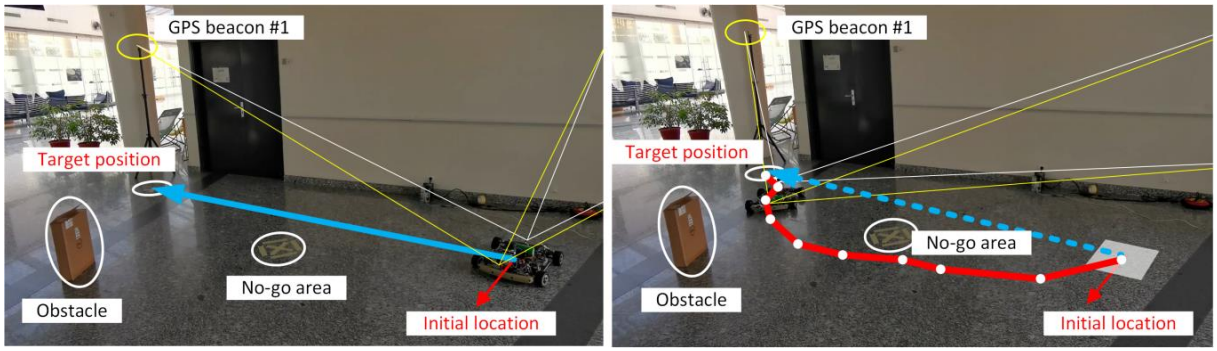

Figure 9 Experimental environment for indoor navigation

4.2 Experiment: Implementing High-Curvature Robot Paths

For the experiment, the maximum velocity was set to 0.50 m·s^{-1}, and the maximum acceleration was 0.30 m·s^{-2}. The robot initial location was placed at (1.886, 5.376, -39.586º), and the target was defined by its position coordinates (0.645, 0.579). The no-go area was presented as a circle of radius 0.350 m centered at (1.770, 3.96). The detecting range of the LiDAR was set to 1.0 m for moving obstacles.

In the implemented motion planning algorithm, the robot workspace (ρ, θ) was discretized with $n = 10$ samples for the polar radius ρ, and angle step $\Delta\theta = 1°$. It was converted into a directed graph composed of from 1800 to 2700 nodes after collision checking, which was used as an input to the developed DP-based algorithm producing a collision-free path. Then, the proposed procedure of the time-optimal trajectory generation was applied. Here, the robot state space $(s, \mathrm{d}s/\mathrm{d}t)$ was discretized with the number of the

speed increments $v = \mathrm{d}s/\mathrm{d}t$ set to 30, and the number of the distance increments $n = 10$ was the same as for the ρ discretization in the path planning. Thus, the robot state space was converted into a directed graph composed of 300 nodes. Finally, a time sequence of the speeds $v(t_i)$ for the robot gravity center was obtained and transformed into the angular speeds of the wheels.

An experimental result confirming the validity of the proposed approach is presented in Fig. 10, which shows the robot motion and its planned path evolution caused by changing of the environment. At the beginning of the experiment, the robot was placed at its initial location, and the planned path was obtained taking into account the static obstacle only (see case t = 0.0 sec, when the path was expected to be on the left side of the static obstacle). Slightly later (see case t = 5.80 sec), when the robot posture changed, the route was modified and located on the other side of the obstacle to meet the shortest path objective. The main reason for this change is that the maximum steering constraint (2) is satisfied during the initial planning whereas it is actually violated in practice. That is caused by the inaccuracies coming from the approximate discrete presentation of constraint (2), as shown by equation (18). Such violation may appear when the system status is close to the limit border. As shown in the figure, the initial path at the time t = 0.00 sec is quite curved, already at the limits of the maximum steering angle. It is also worth mentioning that the corresponding route segment at t = 5.80 sec was close to a straight line, and the robot speed reached almost the maximum value, about 0.48 $\mathrm{m \cdot s^{-1}}$. But, when the robot was avoiding the static obstacle (at the time t = 8.80 sec), the path curvature increased causing the speed reduction down to 0.40 $\mathrm{m \cdot s^{-1}}$. Further, at the time instant t = 10.00 sec, when the robot ‘saw’ a dynamic obstacle, the path was essentially re-planned to avoid any collisions while moving to the target point. While moving in the dynamic obstacle neighbourhood, the path curvature was very low allowing the robot to use its maximum speed of 0.50 $\mathrm{m \cdot s^{-1}}$, see case t = 11.20 sec. At the final segment,

when the robot was reaching its target position (see time t = 16.80 sec), the path curvature increased, and the speed decreased down to 0.29 $m \cdot s^{-1}$. In more detail, the evolution of the robot location recorded by GPS and the estimated displacement along the path are shown in Fig. 11.

To demonstrate the impact of the wheel-ground adhesion constraints on the robot motion, Fig. 12 presents the time profiles of the planned speeds and the corresponding path curvature to the total displacement. As can be seen from this figure, the path curvature is very low at the time instance t = 5.6 sec, 8.0 sec, and 12.0 sec, so the robot speed is planned to be maximum (or slightly lower because of the coarse discretization). In contrast, for other time segments, the path curvature cannot be ignored, and the robot speed is reduced in order to satisfy the wheel-ground adhesion constraint. It is worth mentioning that the proposed constraint can be applied either to four wheels or to the virtual wheel located at the robot's center of gravity for simplicity. Considering the computational capacity of the hardware, in the experimental study, the proposed constraint was applied to the center of gravity. As we assume the robot’s center of gravity and geometric centroid are overlapped, so the velocities of two wheels on the same side should be the same. Such simplification is reasonable here because the robot wheelbase of 0.450 m is rather small and even shorter than its wheel track of 0.482 m. This simplification is also justified by Fig. 12 where the optimal time profiles for both the robot center and wheels are presented.

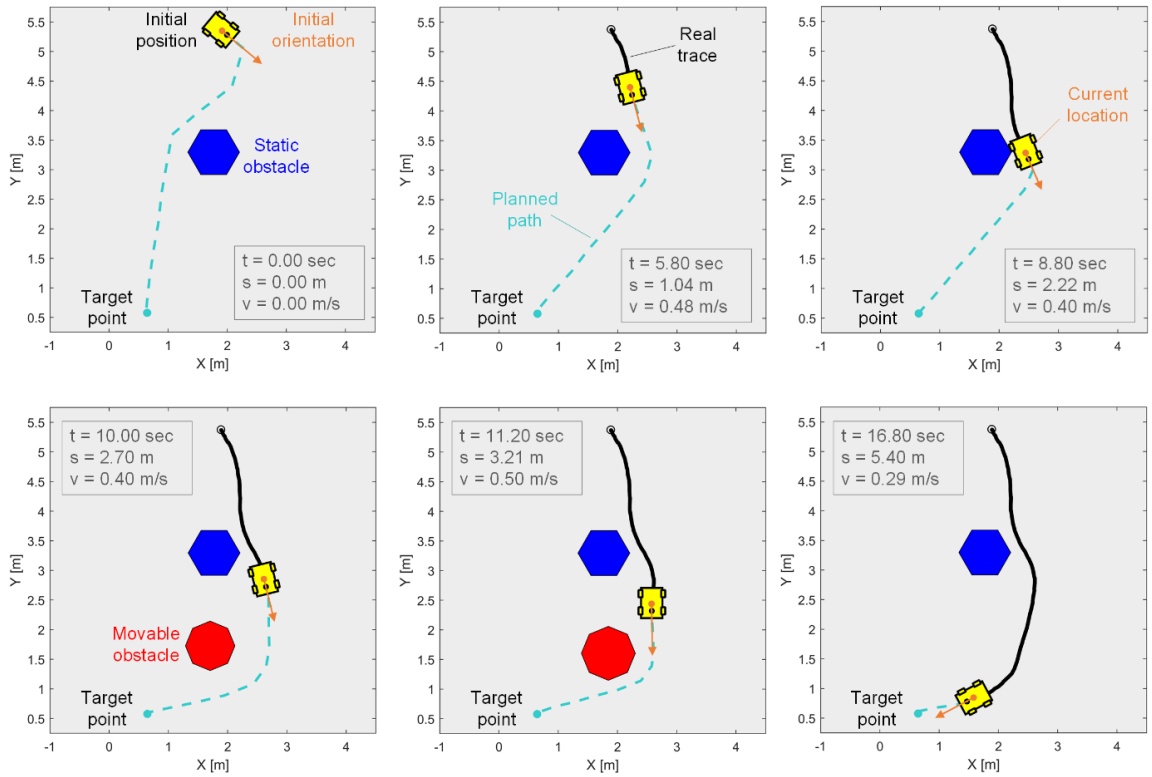

Figure 10 Evolution of the robot location and its planned path in the dynamic environment

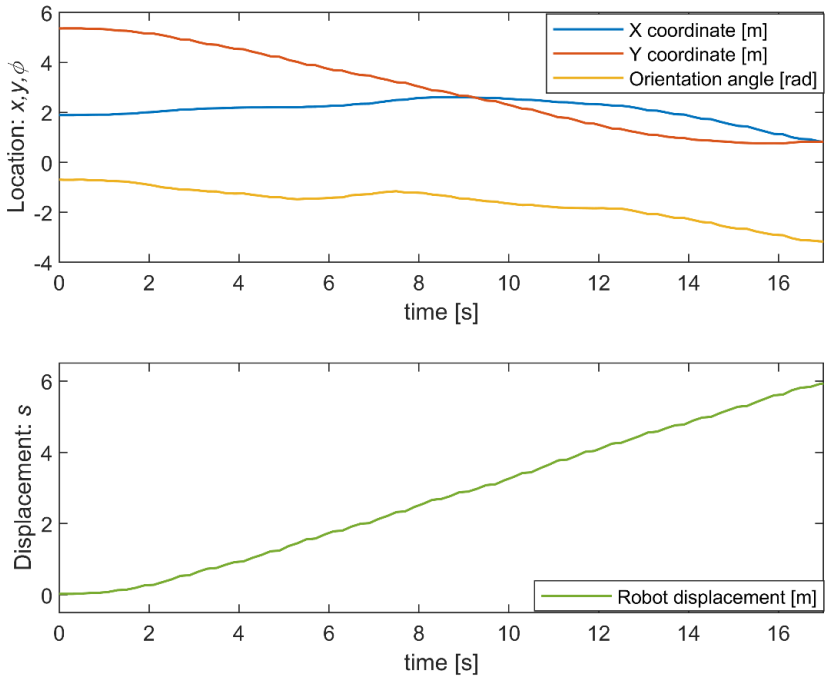

Figure 11 Time profiles of robot location coordinates and displacement

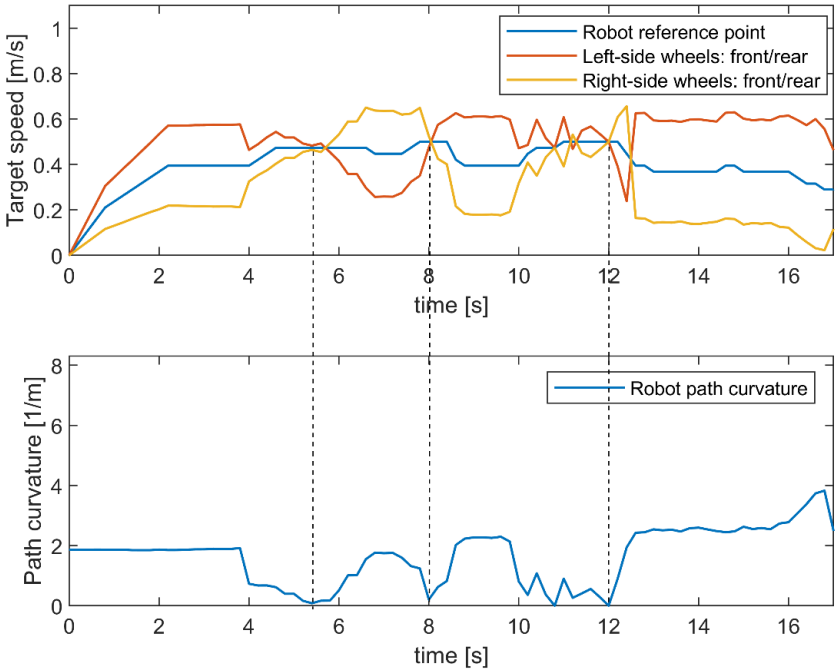

Figure 12 Time profiles of planned robot speed and corresponding path curvature

To show the validity of the proposed adhesion constraint, Fig. 13 presents the time profiles of the planned longitudinal and lateral accelerations, as well as their resultant acceleration (see the left hand side of equation (9), the simplification of the resultant tire force). As can be seen from this figure, at the time instances t = 8.8 sec and 12.5 sec, the resultant acceleration reached its limit (or slightly lower because of the coarse discretization). It is also shown in Fig. 12 that at those two time instances, the robot was trying to enter the high curvature path segment with relative high speed. This result confirms that by applying the proposed constraint, the resultant acceleration is always within the friction limit, and the adhesion condition is always satisfied.

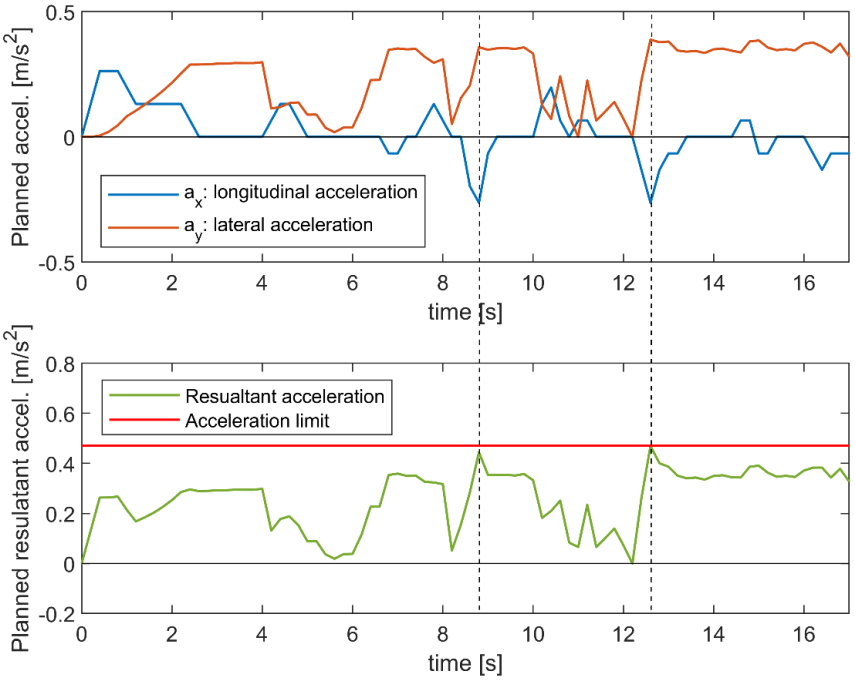

Figure 13 Time profiles of planned longitudinal, lateral, and resultant accelerations

To demonstrate the effectiveness of the proposed technique, the obtained time-optimal trajectory has been also compared with the one from the conventional phase plane method which does not allow taking into account the non-linear adhesion constraint (9) where the velocity and acceleration are coupled. As can be seen from Fig. 14, using the conventional phase plane method without wheel-ground adhesion constraint, the resultant acceleration of the robot will be out of the friction limits when the path curvature and the robot speed are both relative high, whereas the proposed solution ensures the wheel-ground adhesion for the entire robot motion.

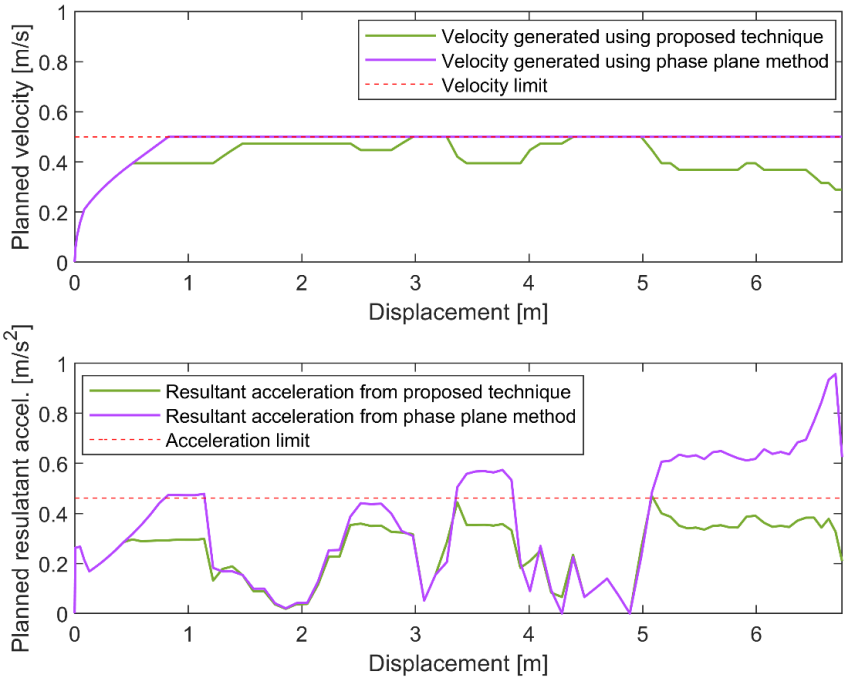

Figure 14 Comparison of motions obtained from the proposed approach and conventional phase plane method

In our experiments, to implement the planned path and generated trajectory along this path, the PI speed controllers were employed for all wheels. Corresponding time profiles are presented in Fig. 15, which shows some small differences between the planned and real wheel speeds (as well as between the speeds of the front and rear wheels) that are not essential for the global motion and are compensated due to periodical path/trajectory re-planning in accordance with the current robot location and its environment. Therefore, the obtained experimental results confirm the validity and practical reasoning of the proposed robot motion planning technique.

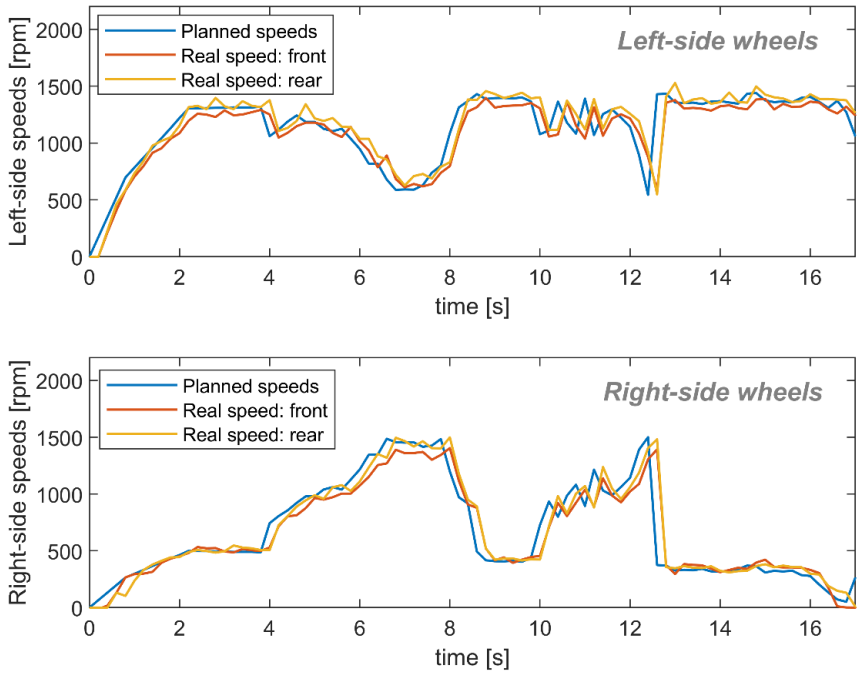

Figure 15 Experimental time profiles of the wheel speeds (PI-controlled)

5. Discussion

In spite of the obvious advantages of the proposed technique (practical benefits confirmed by the experimental study, the capability to take into account the velocity/acceleration limits, and the wheel-ground adhesion constraints), there are a number of limitations and open questions to be discussed. The first of them is related to the selection of the discretization step for the path planning and trajectory generation. It is clear that a smaller discretization step may provide faster and smoother motions but require extremely high computational efforts that may be beyond the system hardware capacity. On the other side, too excessive discretization does not yield significant improvement of our engineering objective that is focusing on generating in real-time the smooth, collision-free, and fast motions. In our experiments, the distance was discretized with the maximum step 0.6 m and the speed was sampled with the largest step 0.017 $m \cdot s^{-1}$, which were sufficient to obtain in real-time fast and smooth robot

motions within a workspace of 5×6 m^2. Clearly, it is a trade-off between the robot motion quality and our hardware capacities. However, in future implementations, it is reasonable to adjust the discretization steps for both the path planning and trajectory generation in order to have a better adaptation to the robot working environment. It also should be mentioned that when the robot works in a wider workspaces like warehouses or factories, a hierarchical architecture will be applied to the path planner which includes a global one and a local one. The global path planner is event-based, not updated at each time step, and the local one uses the same moving window as the trajectory generator, whose computing time remains the same. Therefore, even in big workspaces, the computational effort of the proposed technique will not increase significantly.

It is also worthy of mentioning that the main particularity of this work (dealing with the wheel-ground interaction modelling) is based on a number of assumptions simplifying the robot motion planning. For instance, the wheel-ground friction coefficient μ was assumed to be known and constant for all wheels while in the real working condition it can vary essentially. In practice, a simple way of selecting μ is to use the table including peak friction coefficient according to different road conditions, which is widely used in automotive field [35]. For example, on the dry concrete road, maximum μ can be 0.9, whereas 0.1 on the ice road. They are experimentally based, and sufficient for common road surfaces. A more accurate way is to estimate such coefficient in real-time while using speed and force sensors installed on wheel. It is usually implemented on vehicle electronic systems as ABS or ESP. In addition, in the current work, the wheel slip angles and the robot chassis sideslip angles are always assumed to be ignorable. Hence, it is reasonable in the future to expand the proposed technique by including a more precise tire model.

6. Conclusion

This paper proposes a new method to generate the fastest collision-free motion for an autonomous mobile robot in real-time. In contrast to the existing motion planning algorithms, the proposed one takes into account not only constraints from the robot environment and its kinematics/dynamics, but also some limits caused by the wheel-ground interaction to avoid excessive skidding. An experimental study confirms the validity and practical reasoning of the proposed technique.

In the future, the developed approach could be enhanced by executing the perception and motion planning procedures with different time frequencies, which allows reducing the computational efforts and producing smoother trajectories. The circular expression of collision-free constraints for robots can be improved by using more accurate approximations, for example in [36]. Also, more accurate approximations of the known/unknown obstacles would be necessary, in order to have a better adaptation to various obstacles, such as walls or tables, which are currently approximated by different small circles at each time step.

References

[1] Ben-Ari M, Mondada F. Robots and Their Applications. Elements of Robotics. Cham: Springer International Publishing; 2018. p. 1-20.

[2] González D, Pérez J, Milanés V, Nashashibi F. A review of motion planning techniques for automated vehicles. IEEE Transactions on Intelligent Transportation Systems. 2016;17(4):1135-45.

[3] Bacha A, Bauman C, Faruque R, Fleming M, Terwelp C, Reinholtz C, et al. Odin: Team victortango's entry in the darpa urban challenge. Journal of field Robotics. 2008;25(8):467-92.

[4] Bohren J, Foote T, Keller J, Kushleyev A, Lee D, Stewart A, et al. Little ben: The ben franklin racing team's entry in the 2007 DARPA urban challenge. Journal of Field Robotics. 2008;25(9):598-614.

[5] McNaughton M, Urmson C, Dolan JM, Lee J-W, editors. Motion planning for autonomous driving with a conformal spatiotemporal lattice. 2011 IEEE International Conference on Robotics and Automation; 2011: IEEE.

[6] Ziegler J, Stiller C, editors. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2009: IEEE.

[7] Elbanhawi M, Simic M. Sampling-Based Robot Motion Planning: A Review. IEEE Access. 2014;2:56-77. doi: 10.1109/ACCESS.2014.2302442.

[8] Kuwata Y, Teo J, Fiore G, Karaman S, Frazzoli E, How JP. Real-Time Motion Planning With Applications to Autonomous Urban Driving. IEEE Transactions on Control Systems Technology. 2009;17(5):1105-18. doi: 10.1109/TCST.2008.2012116.

[9] Mohammed H, Romdhane L, Jaradat MA. RRT* N: An efficient approach to path planning in 3D for Static and Dynamic Environments. Advanced Robotics. 2021;35(3-4):168-80.

[10] Motonaka K, Miyoshi S. Obstacle avoidance using buffered voronoi cells based on local information from a laser range scanner. Advanced Robotics. 2022:1-14.

[11] Xiong C, Chen D, Lu D, Zeng Z, Lian L. Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization. Robotics and Autonomous Systems. 2019;115:90-103. doi: https://doi.org/10.1016/j.robot.2019.02.002.

[12] Magid E, Lavrenov R, Afanasyev I, editors. Voronoi-based trajectory optimization for UGV path planning. 2017 International Conference on Mechanical, System and Control Engineering (ICMSC); 2017 19-21 May 2017.

[13] Arnaoot HM, Abdin HA, editors. Visibility Graph-Based Path Planning Algorithm Safety Evaluation and Optimization. 2022 International Telecommunications Conference (ITC-Egypt); 2022: IEEE.

[14] Kunchev V, Jain L, Ivancevic V, Finn A, editors. Path Planning and Obstacle Avoidance for Autonomous Mobile Robots: A Review2006; Berlin, Heidelberg: Springer Berlin Heidelberg.

[15] Malone N, Chiang H, Lesser K, Oishi M, Tapia L. Hybrid Dynamic Moving Obstacle Avoidance Using a Stochastic Reachable Set-Based Potential Field. IEEE Transactions on Robotics. 2017;33(5):1124-38. doi: 10.1109/TRO.2017.2705034.

[16] Minguez J, Lamiraux F, Laumond J-P. Motion Planning and Obstacle Avoidance. In: Siciliano B, Khatib O, editors. Springer Handbook of Robotics. Cham: Springer International Publishing; 2016. p. 1177-202.

[17] Ge SS, Cui YJ. Dynamic Motion Planning for Mobile Robots Using Potential Field Method. Autonomous Robots. 2002;13(3):207-22. doi: 10.1023/a:1020564024509.

[18] Kim TH, Kon K, Matsuno F. Region with velocity constraints: map information and its usage for safe motion planning of a mobile robot in a public environment. Advanced Robotics. 2016;30(10):635-51.

[19] Molinos EJ, Llamazares Á, Ocaña M. Dynamic window based approaches for avoiding obstacles in moving. Robotics and Autonomous Systems. 2019;118:112-30. doi: https://doi.org/10.1016/j.robot.2019.05.003.

[20] Kwakernaak H, Sivan R. Linear optimal control systems: Wiley-interscience New York; 1972.

[21] Weiguo W, Huitang C, Peng-Yung W, editors. Optimal motion planning for a wheeled mobile robot. Robotics and Automation, 1999 Proceedings 1999 IEEE International Conference on; 1999: IEEE.

[22] Shen P, Zhang X, Fang Y. Complete and Time-Optimal Path-Constrained Trajectory Planning With Torque and Velocity Constraints: Theory and Applications. IEEE/ASME Transactions on Mechatronics. 2018;23(2):735-46. doi: 10.1109/TMECH.2018.2810828.

[23] Petrinić T, Brezak M, Petrović I. Time-optimal velocity planning along predefined path for static formations of mobile robots. International Journal of Control, Automation and Systems. 2017;15(1):293-302. doi: 10.1007/s12555-015-0192-y.

[24] Consolini L, Locatelli M, Minari A, Piazzi A. An optimal complexity algorithm for minimum-time velocity planning. Systems & Control Letters. 2017;103:50-7. doi: https://doi.org/10.1016/j.sysconle.2017.02.001.

[25] Kim J, Kim BK. Cornering Trajectory Planning Avoiding Slip for Differential-Wheeled Mobile Robots. IEEE Transactions on Industrial Electronics. 2019:1-. doi: 10.1109/TIE.2019.2941156.

[26] Xiong L, Fu Z, Zeng D, Leng B. An optimized trajectory planner and motion controller framework for autonomous driving in unstructured environments. Sensors. 2021;21(13):4409.

[27] Diachuk M, Easa SM. Motion Planning for Autonomous Vehicles Based on Sequential Optimization. Vehicles. 2022;4(2):344-74.

[28] Okuyama I, Maximo MR, Afonso RJ. Minimum-time trajectory planning for a differential drive mobile robot considering non-slipping constraints. Journal of Control, Automation and Electrical Systems. 2021;32(1):120-31.

[29] Micheli F, Bersani M, Arrigoni S, Braghin F, Cheli F. NMPC trajectory planner for urban autonomous driving. Vehicle System Dynamics. 2022:1-23.

[30] Gao J, Pashkevich A, Claveau F, Chevrel P, editors. Optimal Motion Generation for Mobile Robot with Non-Skidding Constraint. 2019 IEEE International Conference on Mechatronics (ICM); 2019 18-20 March 2019.

[31] Gao J, Pashkevich A, Claveau F, Chevrel P. Real Time Motion Generation for Mobile Robot. IFAC-PapersOnLine. 2019;52(13):265-70. doi: https://doi.org/10.1016/j.ifacol.2019.11.179.

[32] Jazar RN. Vehicle dynamics: theory and application: Springer; 2017.

[33] Rill G, editor TMeasy--A Handling Tire Model based on a three-dimensional slip approach. Proceedings of the XXIII International Symposium on Dynamic of Vehicles on Roads and on Tracks (IAVSD 2013), Quingdao, China; 2013.

[34] Bobrow JE, Dubowsky S, Gibson J. Time-optimal control of robotic manipulators along specified paths. The international journal of robotics research. 1985;4(3):3-17.

[35] Jin H, Zhou M, editors. On the road friction recognition based on the driving wheels deceleration. 2014 IEEE Conference and Expo Transportation Electrification Asia-Pacific (ITEC Asia-Pacific); 2014: IEEE.

[36] Ito N, Okuda H, Suzuki T. Configuration-aware model predictive motion planning for Tractor–Trailer Mobile Robot. Advanced Robotics. 2022:1-15.