A Planner for All Terrain Vehicles on Unknown Rough Terrains based on the MPC Paradigm and D*-like Algorithm

Adnan Tahtirović, Mehmed Brkić, Gianantonio Magnani and Luca Bascetta

Abstract—A novel conceptual design of a planner for a mobile vehicle, operating on poorly traversable unknown rough terrains, is discussed. Finding a way to include a vehicle model into the planning stage, while coping with unknown or partially known terrains, is a challenging and rarely addressed optimization setup. The main advantages of a possible solution of such a problem would be twofold. First, the planner would give trajectories which are feasible to follow by the vehicle, which is not the case in many other state of the art planning algorithms especially for large vehicle speeds. Second, those trajectories would be the optimal ones in accordance to the current vehicle states and knowledge on its environment.

We propose a solution based on an MPC planning paradigm, wherein the planner solves a constrained optimal control problem at each time instant using the current knowledge on the terrain, which is captured appropriately by an objective function. Solving an optimal control problem allows for the vehicle model being included into the planning stage, while the repeated optimization allows for taking continuously into account new terrain information. To deal with the information given beyond the sensor range and to guarantee reaching a given goal position, we have adopted a D*-like algorithm for rough terrains being used as a cost-to-go term within the optimization setup.

I. INTRODUCTION

The popularity of the research on wheeled mobile robots has been recently increasing, due to their possible use in different outdoor environments. Planetary explorations, search and rescue missions in hazardous areas [1], surveillance, humanitarian de-mining [2], as well as agriculture works such as pruning vine and fruit trees, represent possible applications for autonomous vehicles in natural environments. Differently from the case of indoor mobile robotics, where only flat terrains are considered, outdoor robotics deals with all possible natural terrains. The unstructured environment and the terrain roughness, including dynamic obstacles [3], and poorly traversable terrains, make the development of an autonomous vehicle a challenging problem.

The aim of our research is to develop an All-Terrain Mobile Robot (ATMR), based on a commercial All-Terrain Vehicle (ATV), that is suitable for a wide range of different outdoor operations. The ATMR should be able to operate in any natural environment with a high level of autonomy.

The advantage of using ATVs is represented by their good traversability potential for poorly traversable terrains and by the short time spent for reaching the goal, as well as by the possibility to operate in unsafe environments. On the other hand, the main disadvantage of ATVs is their low stability margin due to dynamic constraints, roll-over and excessive side slip [4].

An overview of motion planning algorithms has been presented in [5], [6], [7]. The interpretation of the state-of-the-art given in [7], related to the planning algorithms for mobile vehicles on rough terrains, is outlined in its compressed form in the sequel.

The research on motion planning evolved by adding the capability of taking into account the vehicle motion dynamics constraints within the well known dynamic window approach (DWA) [8], [9]. The DWA selects translational and rotational velocities by maximizing an objective function based on the vehicle heading to the goal position, distance to the closest obstacle, and velocity of the vehicle. The optimization is performed using arcs considering only reachable and safe velocities. This subject was extended to the high-speed navigation of a mobile robot in [10] by the global DWA, as the generalization of the DWA. A combination of the DWA with other methods yielded some improvements in long-term real-world applications [11]. Dubowsky and Iagnemma extended the DWA to rough terrains using the vehicle curvature-velocity space bounded by hazards as well as steering limits, wheel-terrain interaction, rollover and sideslip constraints. In this space the stability constraints of the vehicle, for instance, expressed by limit values of the roll-over and side slip indexes, can be easily described. The given algorithm was also suitable for high speed vehicles and appropriate for real-time implementation [12], [13], [14]. A convergent DWA was obtained for the unicycle mobile vehicle [15] exploiting the model predictive control combined with the direct Lyapunov function approach (MPC/CLF) [15], [16]. Even in the case when some of the aforementioned approaches might be used for unknown or partially known terrains, the continuous recomputation of the objective function term which attracts the vehicle toward the goal position, required when new information come from sensors, is limited to small scale terrains. In addition, all the approaches use the kinematic model of the vehicle, except the one presented in [15], [16], where the dynamic model of unicycle mobile vehicle has been used for planning purposes.

Sequences of motion primitives have been used to cover local planning search space since [17]. More recent works are given in [18] and [19], where the inverse trajectory...
generation was used to navigate UAV and UGV, respectively. The importance of separation in a local planning search space is discussed in [20] and it was shown that the mutual separation of a set of paths is related to the relative completeness of the motions set. The planning approach proposed in [21] generate path sets to navigate an UGV. The planner considers global guidance, satisfies environmental constraints, and guarantees dynamic feasibility by the use of a model-predictive trajectory generator.

A grid based planning approach which takes into account the vehicle differential constraints is introduced in [22]. This planner uses the vehicle model to generate the state lattices assuring the feasible paths along the cost map edges. The heuristic cost estimate, which represents cost-to-go for each node of the grid needed by the A∗ algorithm [23], is taken from a priori calculated heuristic look-up table (HLUT) [24],[25], which is based on the path length, speeding up the algorithm. In [22], the cost edges are calculated by solving two boundary value problem where the control action is parameterized converting the problem into a nonlinear programming one. The cost function represented the minimum slope-dwell performance index, selecting less difficult paths between the nodes (lattice states) that are considered in the overall optimization. Including the vehicle model into the motion planning stage provides a planner which generates trajectories that can be easily followed by a mobile robot. This especially comes to the fore when a vehicle moves with high speed and operates on rough terrains. Using a simpler planner that does not take into account the mobile vehicle model might cause a fatal error due to the difference between the planned and executed trajectories. For this reason, the gradient based algorithms such as the navigation function or a variant of the D∗ [26], [27], in our case are not considered being an acceptable solution.

The sample-based technique for robot motion planning was introduced in [28]. The first sample-based motion planners were not computationally efficient for certain environments. In [29], [30], [31] the probabilistic roadmap method (PRM) was developed for path planning in configuration spaces with many degrees of freedom. A comprehensive overview and discussion about PRM is given in [32] and [33]. PRM method has proved to work well in static well-known environments and are considered computationally efficient for car-like vehicles [34]. In [35], the authors introduced quasi-PRM and lattice roadmap (LRM) algorithms. LRM was extended in [22] to allow the state lattice to represent the differential constraints of the mobile vehicle. However, PRM may not be suitable for planning in a dynamic environment, especially because it does not take into account the vehicle dynamics and might result in very sharp turning points.

Rapidly exploring random trees (RRT) is a type of probabilistic planners originally developed to cope with differential constraints [36], [37], [38]. A significant feature of the RRT-like algorithms is that the resulting trajectories are executable by the underlying dynamical system. The RRT algorithm has been proven probabilistically complete [38], meaning that the probability of finding a solution feasible path converges to one if such a path exists. An improvement of the RRT algorithm was proposed in [39], where the obtained exponential convergence speed yielded a good performance. Several variants of the roughness-based RRTs are illustrated in [40], [41], [42], [43], while some recent results on the RRT-like planners have been introduced in [44] and [45]. Although the RRT may find a feasible solution it cannot be considered an optimal approach in accordance to the current vehicle states and knowledge on the terrain.

The RHC/CLF (Receding Horizon Control/Control Lyapunov Function) scheme developed in [46] used the concept of control Lyapunov function to obtain the stability of RHC scheme. The authors presented the generalization of the RHC/CLF scheme demonstrating its relation to the optimal controller. In [15], the authors have implemented the same scheme (MPC/CLF, Model Predictive Control/Lyapunov Function) for the navigation planning of a unicycle mobile vehicle. The approach developed and proposed in [47] used a passivity-based constraint to obtain an MPC scheme with guaranteed closed loop stability for nonlinear systems. Inspired by this control concept, a framework for mobile robot motion planning using the PB/MPC is presented both for flat and rough terrains in [48], [49], [50], where any dynamic model can be used to plan the vehicle trajectories. The main issue of this framework is computation of the cost-to-go term for the given objective function, which is required each time new information is acquired by the vehicle sensors. The idea of using optimal cost-to-go map computed by Dijsktra algorithm is presented in [51]. However, a continuous computing such a map and/or computing it for large scale terrains can be computationally very expensive, hence inappropriate for real time reactive motion planning. The idea of using a faster algorithm to compute cost map for the purpose of cost-to-go term required by the MPC planning optimization framework, has been addressed in [7]. Therein, an approximate computation, named as Roughness-based Navigation Function (RbNF), has been presented, while some of the applications of using such a cost map have been examined in [52]. Although it has not been presented in [51], using a D∗-like algorithm as a cost-to-go term has been mentioned as a possible solution. Since the D∗-like algorithms inherently deal with unknown terrains, the focus of our current research is an MPC optimization setup which uses a D∗-like algorithm for planning purposes on unknown rough terrains.

In order to use the MPC optimization setup using a D∗-like algorithm for unknown terrains, we properly extend the setup presented in [51] by defining varying state constraints for each MPC optimization cycle in accordance to the vehicle current information on the terrain. The planner may include any dynamic model into the optimization setup, and it can comply with any constraints such as those imposed on acceleration, velocity, roll and slip angles. By using D∗-like algorithm, such a planner deals with unknown terrains in a near-optimal manner.

After the description of the aim of our ongoing research (Section I), the state of the art relevant to the ATV planning
on rough terrains (Section I), the ATV which will be used in the experimental setup (Section II), the novelty of the ongoing research is presented in Section III throughout an MPC optimization setup, and it is validated within Section IV.

II. THE ATMR

The vehicle considered in this research (see Fig. 1) is a YAMAHA GRIZZLY 700, a commercial fuel powered All-Terrain Vehicle (ATV) equipped with an electric power steering (EPS). The GRIZZLY 700 is a utility ATV and is thus specifically designed for agriculture work. As a result it has a total load capacity of 130 Kg, and it is equipped with a rear tow hook. The main characteristics of the vehicle are listed in Table I.

![The Yamaha Grizzly 700 ATV](image)

For the purposes of the project, the original vehicle cover has been removed and substituted with an aluminium cover, that allows to easily accommodate for the control hardware and the sensors.

<table>
<thead>
<tr>
<th>Main characteristics of the vehicle</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine type</td>
<td>686cc, 4-stroke, liquid-cooled, 4 valves</td>
</tr>
<tr>
<td>Drive train</td>
<td>2WD, 4WD, locked 4WD</td>
</tr>
<tr>
<td>Transmission</td>
<td>V-belt with all-wheel engine braking</td>
</tr>
<tr>
<td>Brakes</td>
<td>dual hydraulic disc (both f/r)</td>
</tr>
<tr>
<td>Suspensions</td>
<td>independent double wishbone (both f/r)</td>
</tr>
<tr>
<td>Steering System</td>
<td>Ackermann</td>
</tr>
<tr>
<td>Dimensions (LxWxH)</td>
<td>2.065 x 1.180 x 1.240 m</td>
</tr>
<tr>
<td>Weight</td>
<td>296 Kg (empty tank)</td>
</tr>
</tbody>
</table>

TABLE I

**VEHICLE CHARACTERISTICS**

Control system and software architectures used on the ATMR have been described and presented in [51].

III. AN MPC-BASED PLANNER ON UNKNOWN TERRAINS

In this work we extend the idea presented in [51] from the ATV planning on known to unknown terrains. We use $D^*$-like algorithm, which inherently deals with unknown terrains, as an input to form a properly differentiable function, providing for a cost-to-go term required by the MPC optimization setup. In order to count for the new information continuously coming into the vehicle sensor range, we define a local term of an objective function as a measure of roughness in the same way as in [51]. However, to locally deal with the new obstacles, we now define a varying constraint set for the MPC optimization setup.

The MPC optimization problem can be expressed as an initial value optimal control problem (OCP) with an end-free position eqs. (1-4). The task of this optimization is to find the input $u$ of the vehicle (the velocity and steering angle for a kinematic model, or the acceleration and steering momentum for a dynamic model) along the optimization horizon $t \in (t_0, t_0 + T)$, that is over all potential candidate paths, by minimizing the cost function $J(u)$ given in (1).

The integrand $\gamma(x,u)$ represents a differentiable function as a measure of the local terrain roughness estimated by the vehicle within the sensor range. Many different terrain roughness measures can be used [53], [7] to compute the roughness cost map. To get such a differentiable function, we interpolate roughness map over the sensor range domain for each MPC optimization cycle.

The cost-to-go term $\Gamma(t_0 + T)$ represents a differentiable function as a measure of the estimated cost-to-go map in accordance with the current vehicle information on the terrain. This cost-to-go map is computed by $D^*$ using both obstacle and roughness maps acquired by the vehicle prior the beginning of each MPC optimization cycle. Hence, $\Gamma(t_0 + T)$ gives the cost of traversing the path on rough terrain from a position at the terminal optimization time, $t_0 + T$, to a goal position, by using $D^*$ algorithm. To get such a differentiable function, we now interpolate the cost-to-go map over the sensor range domain.

Eqs. (2-4) represent optimization constraints including the differential constraint related to the vehicle model (2), state constraints and control constraints (4). State constraints vary with each MPC optimization horizon such that they prevent the vehicle terminal state, $x(t_0 + T)$, being beyond the sensor range. This constraint is important since functions $\gamma(x,u)$ and $\Gamma(t_0 + T)$ are interpolated using the discrete values from the roughness and cost-to-go maps over that range, respectively. Obstacles, which are located within the sensor range, are also appropriately included in eq. (3). Other constraints such as those preventing the vehicle from the sideslip and rollover can also be easily accommodated into the optimization setup.

$$J(u) = \int_{t_0}^{t_0 + T} \gamma(x,u) dt + \Gamma(t_0 + T)$$  \hspace{1cm} (1)

$$\frac{dx}{dt} = f(x,u)$$  \hspace{1cm} (2)
\[ x(t) \in X, \quad (3) \]
\[ u(t) \in U \quad (4) \]

The optimization is solved by using an OCP software GPOPS I [54], providing the planner eqs. (1-4) being nearly optimal due to the 'optimality principle' since the D* algorithm is a near optimal estimator of the cost-to-go map. This is equally true both for the known and unknown terrains. For the latter case, D* algorithm estimates the cost-to-go map nearly optimal in accordance to the vehicle current belief on the environment.

IV. SIMULATIONS

We illustrate two simulation setups in order to show two different ways of using the proposed planner on unknown terrains. Since the optimization framework allows for using any vehicle model, we use, without loss of generality, a double integrator as a model of the vehicle. For the purpose of simplicity, we assume that the vehicle perfectly follows the planning path for each MPC control horizon.

The first setup, illustrated with Figs. 2 and 3, utilizes a rough terrain with obstacles. The path depicted with the blue line is generated by the planner given by eqs. (1-4) (complete planner), while the one depicted with the red line is generated without using local objective function term \( \gamma(x, u) \) (D* gradient based planner). Namely, in the latter case, we obtain a planner which is the result only of the cost-to-go term, \( \Gamma(t_0 + T) \), meaning that the planner follows the gradient of the D* map at the end of the current optimization horizon, \( t_0 + T \). Such a planner mimics the D* algorithm and it also includes the model of the vehicle into optimization setup. However, the model without using local objective function term does not take into account the terrain roughness within the sensor range. This fact can be seen in Figs. 2 and 3, where Fig. 2 illustrates the paths on the contour plot of the roughness cost map, and Fig. 3 depicts the paths on the contour plot of the cost-to-go map obtained by D* algorithm once the terrain is completely known. Blue: the complete planner. Red: D* gradient based planner. Initial position: (15,15). Goal position: (90,90).

The second simulation setup, illustrated with Figs. 4, 5 and 6 shows a different implementation possibility. In this case, the obstacles are considered as highly rough terrain parts, as shown in Fig. 4. Figs. 5 and 6 show the generated paths both for the complete and the D* gradient based planner. In all simulated cases, the proposed planner appropriately guides the vehicle through unknown terrain.

V. CONCLUSIONS

This paper describes part of the work devoted to the development of an All-Terrain Mobile Robot, based on a commercial All-Terrain Vehicle, for riding on unknown difficult terrains. Besides the developments of control and
A thorough statistical analysis and experimental validation, for preventing the vehicle from the sideslip and rollover. A wide range of constraints such as obstacles and those the vehicle model during the planning stage and impose the optimal planning technique based on an MPC combined with our current research is the development of an ATV reactive covered in our previous work, one of the main issues of software architecture for such a vehicle, which have been covered in our previous work, one of the main issues of our current research is the development of an ATV reactive planner for unknown terrains. We propose a novel near optimal planning technique based on an MPC combined with the D* algorithm. An MPC based planner can account for the vehicle model during the planning stage and impose a wide range of constraints such as obstacles and those for preventing the vehicle from the sideslip and rollover. A thorough statistical analysis and experimental validation, using the vehicle described in Section II, is our ongoing research.

REFERENCES


Fig. 5. The generated paths depicted on the roughness cost map, where the obstacles are considered as highly rough part of the terrains. Blue: the complete planner. Red: D* gradient based planner. Initial position: (50,35). Goal position: (30,90).

Fig. 6. The generated paths depicted on the contour plot of the cost-to-go map obtained by D* algorithm once the terrain is completely known. Blue: the complete planner. Red: D* gradient based planner. Initial position: (50,35). Goal position: (30,90).