# Hierarchical Motion Planning with Kinodynamic Feasibility Guarantees: Local Trajectory Planning via Model Predictive Control

Raghvendra V. Cowlagi and Panagiotis Tsiotras

***Abstract*— Motion planners for autonomous vehicles often involve a two-level hierarchical structure consisting of a high-level, discrete planner and a low-level trajectory generation scheme. To ensure compatibility between these two levels of planning, we previously introduced a motion planning framework based on multiple-edge transition costs in the graph used by the discrete planner. This framework is enabled by a special local trajectory generation problem, which we address in this paper. In particular, we discuss a trajectory planner based on model predictive control for complex vehicle dynamical models. We demonstrate the efficacy of our overall motion planning approach via examples involving non-trivial vehicle models and complex environments, and we offer comparisons of our motion planner with state-of-the-art randomized sampling-based motion planners.**

## I. Introduction

Motion planning for autonomous vehicles [1], i.e., the problem of finding control inputs that enable vehicles to satisfy high-level task specifications, is often solved over two hierarchical levels. The higher level *geometric path planner* typically uses a discrete representation of the vehicle's workspace (such as workspace cell decompositions) and deals with the satisfaction of the task specifications (such as obstacle avoidance). The lower level *trajectory planner* deals with the vehicle's kinematic and dynamic constraints.

The hierarchical approach described above suffers from a lack of "consistency" between the two planners, in that the geometric path may be infeasible or unacceptably suboptimal when the vehicle dynamical constraints are considered at the trajectory planning level. To address this problem, we introduced in [2] a motion planning framework based on assigning costs to *multiple* edge transitions in the graphs associated with cell decompositions. In particular, we introduced in [2] the so-called *tile motion planning* problem which facilitates an interaction between the two planners. In [2], [3], we discussed the solution of the tile motion planning problem, using purely geometric constructions, for the Dubins car [4] kinematic model. In this paper, we present a general scheme, based on the well-known model predictive control paradigm, for implementing the tile motion planner for complex vehicle dynamical models.

Model predictive control (MPC) is a popular approach for control design in the presence of state- and input constraints [5], [6], and MPC-based approaches for trajectory generation and motion planning have previously appeared in the literature. For instance, a mixed integer linear programming (MILP) formulation of the motion planning problem, which involves the introduction of several binary decision variables to obtain a linear program that encodes obstacle-avoidance constraints, has been developed [7], [8], and applied for planar path planning [9] and three dimensional path planning [10] for UAVs in cluttered environments while including minimum turn radius constraints. Similarly, receding horizon path planning has been investigated in the contexts of vision-based navigation [11]; of obstacle avoidance for a bicycle model [12]; of trajectory generation for wheeled vehicles moving over rough terrain [13]; of path planning for environments involving both static and moving obstacles [14]; and of robust path planning [15], [16].

A serious problem associated with MPC-based motion planning is the unavoidable presence of non-convex state constraints arising from the obstacle-avoidance requirement. The MILP formulation discussed in [7] is one approach to alleviate this difficulty. In the tile motion planning scheme proposed in this paper, we use the idea of *effective target sets* [17] to transform non-convex state constraints on the MPC optimization problem into convex constraints, along with a special boundary condition.

The main contributions of this paper are as follows. The tile motion planner discussed in this paper is a crucial component of the overall motion planning framework described in [2]. This motion planning framework is powerful in that it allows the discrete and continuous facets of motion planning to be separated from one another while maintaining a guarantee of "consistency" between the two planners. In this paper, we provide concrete examples of applications of this motion planning framework for non-trivial vehicle dynamical models. Also, we demonstrate that the idea of effective target sets (which may be computed offline) can be used to reduce the complexity of local trajectory generation. In light of the limited on-board computational resources of autonomous vehicles, the proposed method of trajectory generation requires the solution of a simpler online problem with fewer variables and constraints, as compared to a nonlinear programming formulation or a MILP formulation. Finally, we demonstrate via numerical simulation results that the overall motion planner, which is enabled by the local trajectory generation scheme discussed in this paper, results in trajectories of significantly lower-cost in comparison to state-of-the-art randomized sampling-based motion planners.

The rest of this paper is organized as follows. In Section II, we describe briefly the tile motion planning problem introduced in [2]. In Sections III and IV, we discuss the applica-

R. V. Cowlagi is a Postdoctoral Associate at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 30332, USA. rcowlagi@mit.edu

P. Tsiotras is with the Faculty of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. tsiotras@gatech.edu

tion and the computation of effective target sets for solving the tile motion problem using MPC. Finally, in Section V, we provide numerical simulation results demonstrating the efficacy of the overall motion planner.

## II. $H$-COST MOTION PLANNING

Workspace cell decompositions [1, Ch. 5], which partition the workspace into convex regions called *cells*, are frequently used in path planning. A graph $\mathcal{G} = (V, E)$ is associated with the decomposition, such that each cell corresponds to a unique vertex in $V$ and each pair of geometrically adjacent cells corresponds to a unique edge in $E$. We will denote by $\mathsf{cell}(j)$ the cell associated with the vertex $j \in V$.

It has been noted in several previous works [9], [18], [19], including ours [2], that single-edge transition costs cannot capture adequately the vehicle's kinematic and dynamic constraints. In light of this observation, we discussed a motion planning framework [2] based on the solution of the so-called *$H$-cost shortest path problem*, which is defined as the problem finding a path of least cost in a graph where transition costs are defined on *multiple* successive edges (called *$H$-histories*). In this motion planner, the transition costs on $H$-histories are assigned by solving a low-level trajectory generation problem described next.

We consider a vehicle model described as follows. Let $(x, y, \theta) \in \mathcal{C} := \mathbb{R}^2 \times \mathbb{S}^1$ denote the position coordinates of the vehicle in a pre-specified Cartesian axis system, and let $\psi$ denote any additional state variables required to describe the state of the vehicle. We assume that $\psi \in \Psi$, where $\Psi$ is a $n$-dimensional smooth manifold. The state of the vehicle is thus described by $\xi := (x, y, \theta, \psi) \in \mathcal{D} = \mathcal{C} \times \Psi$. Let $U \in \mathbb{R}^m$ denote the set of admissible control values; and for $t > 0$, let $\mathcal{U}_t$ denote the set of piecewise continuous functions defined on the interval $[0, t]$ that take values in $U$. We assume that the evolution of the vehicle state $\xi$ over a given time interval $[0, t_\mathrm{f}]$ is described by the differential equation $\dot{\xi}(t) = f(\xi(t), u(t))$ for all $t \geqslant [0, t_\mathrm{f}]$, where $u \in \mathcal{U}_{t_\mathrm{f}}$ is an admissible control input, and $f$ is sufficiently smooth to guarantee global existence and uniqueness of solutions. We denote by $\xi(\cdot\ ; \xi_0, u)$ the state trajectory that is the unique solution to the preceding differential equation with initial condition $\xi(0) = \xi_0$. Finally, we denote by $\mathsf{x}(\xi)$ the projection of a state $\xi$ on $\mathbb{R}^2$.

We define a *tile* as the sequence of cells associated with a $H$-history $(j_0, \ldots, j_{H+1})$, where $j_k \in V$ for each $k = 0, \ldots, H+1$, and $(j_k, j_{k+1}) \in H$ for each $k = 0, \ldots, H$. A *tile motion planner* (TILEPLAN) is any algorithm that determines if a given tile may be feasibly traversed. A precise and general description of TILEPLAN is given in Fig. 1.

## III. MPC-BASED TILE MOTION PLANNING

The implementation of TILEPLAN is difficult mainly because (1) imposes a non-convex constraint on the state trajectory. To alleviate this difficulty, we take advantage of the fact that each cell in the sequence of cells associated with a tile is a convex region, using the idea of *effective target sets* introduced in [17].

**Tile Motion Planning Algorithm (TILEPLAN)**

```
Determine if there exist $t_\mathrm{f} \in \mathbb{R}$ and admissible control input $u \in \mathcal{U}_{t_\mathrm{f}}$ such that $\xi(\cdot\ ; \xi_0, u)$ satisfies
    $\mathsf{x}(\xi(t; \xi_0, u)) \in \bigcup_{k=1}^{H} \mathsf{cell}(j_k), \ t \in (0, t_\mathrm{f})$,   (1)
    $\mathsf{x}(\xi(t_\mathrm{f}; \xi_0, u)) \in \mathsf{cell}(j_H) \cap \mathsf{cell}(j_{H+1})$   (2)
if $\exists t_\mathrm{f}$ and $\exists u$ then
    Find $t_1$ such that
        $\mathsf{x}(\xi(t_1; \xi_0, u)) \in \mathsf{cell}(j_1) \cap \mathsf{cell}(j_2)$   (3)
    Return $t_1$, $u_{[0,t_1]}$, $\xi_1 := \xi(t_1; \xi_0, u)$, and
        $\Lambda := \int_0^{t_1} \ell(\xi(t; \xi_0, u), u, t)\ \mathrm{d}t$   (4)
else
    Return $\Lambda = \infty$
```

Fig. 1. General form of the tile motion planning algorithm.

The concept of effective target sets is informally described as follows. Consider a discrete-time dynamical system described by $\xi(k+1) = f_\mathrm{d}(\xi(k), u(k)),\ k \in \mathbb{N}$. Let $\xi_0 = \xi(0)$ be the initial state of the system, and let a horizon $N \in \mathbb{N}$ and a target set $\mathcal{X}_N \subseteq \mathcal{D}$ be pre-specified. Consider now the problem of finding a sequence of $N$ control inputs such that $\xi(N) \in \mathcal{X}_N$. Suppose that such a control input sequence exists, and consider the set $\mathcal{X}_{N-1} \subseteq \mathcal{D}$ defined by

$$\mathcal{X}_{N-1} := \{\xi \in \mathcal{D} : \exists u_{N-1} \in U \text{ s.t. } f_\mathrm{d}(\xi, u_{N-1}) \in \mathcal{X}_N\}.$$

It follows that $\xi(N-1) \in \mathcal{X}_{N-1}$. In other words, the original problem can be reduced to the problem of finding a sequence of $N-1$ inputs with the constraint $\xi(N-1) \in \mathcal{X}_{N-1}$. Continuing recursively, we may define sets $\mathcal{X}_k$ by

$$\mathcal{X}_k \quad := \quad \{\xi \in \mathcal{D} : \exists u_k \in U \text{ s.t. } f_\mathrm{d}(\xi, u_k) \in \mathcal{X}_{k+1}\},$$

for $k = 1, \ldots, N-2$, and then reduce the original problem of finding a sequence of $N$ inputs to the problem of finding a single admissible input $u(0)$ such that $f(\xi(0), u(0)) \in \mathcal{X}_1$.

### A. Definitions of Effective Target Sets for TILEPLAN

Consider the tile associated with the $H$-history $(j_0, \ldots, j_{H+1})$. We define a sequence $\{\mathcal{X}_k\}_{k=1}^{H+1}$ of subsets of the vehicle state space, called *effective target sets*, as follows. Let $\mathcal{X}_H := (\mathsf{cell}(j_H) \cap \mathsf{cell}(j_{H+1})) \times [-\pi, \pi] \times \Psi$. For each $k = 1, \ldots, H-1$, we define the effective target set $\mathcal{X}_k$ as the set of all states $\xi_k \in \mathcal{D}$ such that $\mathsf{x}(\xi_k) \in \mathsf{cell}(j_k) \cap \mathsf{cell}(j_{k+1})$ and such that there exists $t_{k+1} \in \mathbb{R}_+$ and an admissible control input $u_{k+1} \in \mathcal{U}_{t_{k+1}}$ such that the state trajectory $\xi(\cdot\ ; \xi_k, u_{k+1})$ satisfies

$$\mathsf{x}(\xi(t; \xi_k, u_{k+1})) \in \mathsf{cell}(j_{k+1}), \ t \in (0, t_{k+1}), \tag{5}$$

$$\xi(t_{k+1}; \xi_k, u_{k+1}) \in \mathcal{X}_{k+1}. \tag{6}$$

Now suppose there exist a time $t_1$ and a control $u_1 \in \mathcal{U}_{t_1}$

such that the resultant state trajectory $\xi(\cdot\,; \xi_0, u_1)$ satisfies

$$\begin{aligned} \mathsf{x}(\xi(t;\xi_0,u_1)) &\in \mathsf{cell}(j_1),\ t\in(0,t_1)\,, && (7)\\ \xi_1 := \xi(t_1;\xi_0,u_1) &\in \mathcal{X}_1. && (8)\end{aligned}$$

Because $\xi_1 \in \mathcal{X}_1$, it follows by (5)-(6) that there exists a $t_2 \in \mathbb{R}_+$ and a $u_2 \in \mathcal{U}_{t_2}$ such that

$$\mathsf{x}(\xi(t;\xi_1,u_2)) \in \mathsf{cell}(j_2), \quad \xi(t_2;\xi_1,u_2)\in\mathcal{X}_2, \quad t\in(0,t_2)\,.$$

In other words, the admissible control input $u_{1-2}$ defined as the concatenation of the inputs $u_1$ and $u_2$ by

$$u_{1-2}(t) \ := \ \begin{cases} u_1(t), & t\in[0,t_1)\,,\\ u_2(t), & t\in[t_1,(t_1+t_2)]\,,\end{cases}$$

enables the vehicle's traversal through the cells corresponding to the vertices $j_1$ and $j_2$. Continuing recursively the preceding arguments, it follows that for each $H \geqslant 2$, there exist $t_{k+1} \in \mathbb{R}_+$ and inputs $u_{k+1} \in \mathcal{U}_{t_{k+1}}$, for $k = 1,\ldots,H-1$, such that the admissible input $u$ defined by

$$u(t) := u_k(t),\ t\in[T_{k-1},T_k)\,, \quad T_k := \textstyle\sum_{m=1}^{k} t_m, \qquad (9)$$

for $k = 1,\ldots,H$, solves the tile motion planning problem.

Thus, if the effective target sets $\mathcal{X}_k$, the corresponding times of traversal $t_{k+1}$ and the control inputs $u_k$ in (9) are known for each $k = 1,\ldots,H$, then the tile motion planning problem is equivalent to the problem of finding $u_1$ and $t_1$ as described above. Crucially, (7) constrains the position components of the state trajectory to lie within a convex set. Furthermore, we may replace $\mathcal{X}_1$ in (8) by an interior convex approximating set $\tilde{\mathcal{X}}_1 \subset \mathcal{X}_1$ thus transforming the tile motion planning problem into the problem of finding $u_1$ and $t_1$ subject to convex constraints.

### B. MPC Problem Formulation

In the MPC formulation of TILEPLAN, we first approximate the vehicle dynamical model by the linear system $\dot{\xi} = A\xi + B_1 u + B_2$, where $A := \frac{\partial f}{\partial \xi}\Big|_{(\xi_0,u_0)}$, $\quad B_1 := \frac{\partial f}{\partial u}\Big|_{(\xi_0,u_0)}$, and $B_2 := f(\xi_0,u_0) - A\xi_0 - B_1 u_0$, and then consider the corresponding discrete-time linear system. We denote by $H_{\mathrm{P}}$ the prediction horizon, by $\tilde{\ell} : \mathcal{D}\times U \to \mathbb{R}_+$ a pre-specified incremental cost function, and by $\tilde{\Lambda}_{\mathrm{f}} : \mathcal{D} \to \mathbb{R}_+$ a pre-specified terminal cost function. The MPC problem is then described as follows:

$$\min_{H_{\mathrm{P}}\in\mathbb{N},(u(0),\ldots,u(H_{\mathrm{P}}))} \left\{ \tilde{\Lambda}_{\mathrm{f}}(\xi(H_{\mathrm{P}})) + \sum_{k=0}^{H_{\mathrm{P}}-1} \tilde{\ell}(\xi(k),u(k)) \right\},$$

$$\begin{aligned}&\text{subject to } \xi(H_{\mathrm{P}}) \in \tilde{\mathcal{X}}_1, \qquad \xi(k)\in\mathsf{cell}(j_1), && (10)\\ &\text{and } u(k)\in U,\ \text{ for each } k\in\{0,\ldots,H_{\mathrm{P}}-1\}.\end{aligned}$$

Note that the incremental cost $\tilde{\ell}$ in (10) need not be the same as the incremental cost $\ell$ in (4): the role of TILEPLAN in the overall motion planning framework is that of ensuring *feasibility* of traversal of tiles, while it is the higher-level discrete planner that searches for an optimal sequence of cell transitions. To implement TILEPLAN, the MPC-problem (10) is solved; the first input of the resulting input sequence is chosen and applied to the actual (nonlinear) vehicle model; the linearization is performed about the new state [6]; and the preceding steps are repeated.

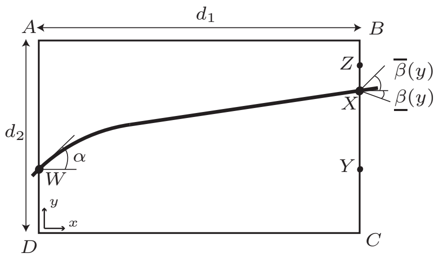


Fig. 2. Setup for Problem 3.

## IV. COMPUTATION OF EFFECTIVE TARGET SETS

In this section, we discuss the construction of the effective target sets that were previously defined for simplifying the MPC implementation of TILEPLAN. First, we consider the computation of the intersections of the effective target sets with the configuration space $\mathcal{C} = \mathbb{R}^2 \times \mathbb{S}^1$. To this end, we define the *effective target configuration sets* by $\mathcal{C}_k := \mathcal{X}_k \cap \mathcal{C}$, and, in what follows, we outline a geometric scheme of computing the sets $\mathcal{C}_k$.

*Assumption 1:* The geometric curves in the plane that can be feasibly traversed by the vehicle satisfy a local upper bound on their curvatures.

We will comment on the validity of Assumption 1 in Section IV-A. First, we use this Assumption to compute the sets $\mathcal{C}_k$ by solving the following problems in plane geometry.

Let $ABCD$ be a rectangle. We attach a Cartesian axes system as shown in Fig. 2. Let the dimensions of the rectangle be $d_1$ and $d_2$, and let $r > 0$ be fixed.

*Definition 2:* Let $\underline{\beta}(x), \overline{\beta}(x)$, $x\in[0,d_2]$ be functions such that $-\frac{\pi}{2} \leqslant \underline{\beta}(x) \leqslant \overline{\beta}(x) \leqslant \frac{\pi}{2}$. Let $Y = (d_1, y), Z = (d_1, z)$ be points on the segment $BC$ with $y \leqslant z$. A path $\Pi$ is a *Type 1 admissible path* if it satisfies the following properties:

1) The curvature at any point on $\Pi$ is at most $r^{-1}$,
2) $\Pi$ intersects the segment $BC$ in exactly one point $X = (d_1, x)$ such that $x\in[y,z]$, and it may intersect segment $AB$ and/or $CD$ in at most one point each, and
3) $\Pi'(X) \in \left[\underline{\beta}(x), \overline{\beta}(x)\right]$,

where $\Pi'(X)$ is the angle of the tangent to $\Pi$ at $X$. A Type 2 admissible path is defined analogously for traversal across adjacent edges. Next, we state two geometric problems as follows. Let $\underline{\beta}, \overline{\beta}, Y$, and $Z$ be as in the preceding definitions. Let $W = (0, w)$ and $r > 0$ be fixed.

*Problem 3 (resp. Problem 4) - Traversal across parallel (resp. adjacent) edges*: Find $\underline{\alpha}, \overline{\alpha}$ such that for all $\alpha\in[\underline{\alpha},\overline{\alpha}]$, there exists a Type 1 (resp. Type 2) admissible path with initial configuration $(W, \alpha)$.

Problems 3 and 4 appear in the recursive computation of effective target configurations as follows. Suppose that the effective target configuration set $\mathcal{C}_{k+1}$ is known. We may then express $\mathcal{C}_{k+1}$ as the product set of the line segment $\mathsf{cell}(j_{k+1}) \cap \mathsf{cell}(j_{k+2})$ with a set of allowable orientations on this line segment. We may then solve Problem 3 or 4, as applicable for the cell $\mathsf{cell}(j_{k+1})$, for each point on the line segment $\mathsf{cell}(j_k) \cap \mathsf{cell}(j_{k+1})$ to obtain allowable orientations for each point on this line segment, and thus construct $\mathcal{C}_k$.

The solutions to Problems 3 and 4 are outlined in [3], and a detailed analysis of these problems appears in [20]. In this paper, we focus on the application of these solutions to TilePlan for different vehicle models. Note that the computation of effective target sets over rectangular channels also enables motion planning for vehicles with a finite size. Specifically, one may constrain allowable trajectories through a given tile to a shruken channel within the tile. This shrunken channel will itself be a rectangular channel, thus allowing direct application the preceding analysis.

### A. Computing Curvature Bounds on Feasible Paths

We may characterize as follows the curvature of the geometric paths corresponding to projections on $\mathbb{R}^2$ of feasible state trajectories. Note that the following kinematical equations relate the inertial position coordinates $x, y$ to the orientation $\theta$ irrespective of the vehicle dynamical model:

$$\dot{x}(t) = v(t)\cos\theta(t), \quad \dot{y}(t) = v(t)\sin\theta(t). \qquad (11)$$

The curvature of the planar curve $p(t) = (x(t), y(t))$ is [21]:

$$\kappa(t) = \sqrt{(\langle\dot{p},\dot{p}\rangle\langle\ddot{p},\ddot{p}\rangle - \langle\dot{p},\ddot{p}\rangle^2)/\langle\dot{p},\dot{p}\rangle^3} = \left|\dot{\theta}/v\right|, \qquad (12)$$

by (11). In the context of the vehicle dynamical model, the curvature of feasible paths is related to the set of admissible control values via the term in the numerator of (12), and the upper bound $\kappa^{\max}$ on the curvature of a feasible path over a given time interval of interest $[0, t_{\mathrm{f}}]$ is

$$\kappa^{\max} \leqslant \min_{t\in[0,t_{\mathrm{f}}]} \max_{u(t)\in U} \left|\dot{\theta}(\xi(t), u(t))/v(t)\right|. \qquad (13)$$

### B. Illustrative Example: Particle Dynamical Model

Consider a vehicle dynamical model described by

$$\begin{array}{ll} \dot{x}(t) = v(t)\cos\theta(t), & \dot{y}(t) = v(t)\sin\theta(t), \\ \dot{\theta}(t) = \omega(t), & \dot{v}(t) = a(t), \end{array}$$

where $v > 0$ is the forward speed of the vehicle; $u_1 = a$ is the acceleration input, and $u_2 = \omega$ is the steering input. The speed $v$ is constrained to lie within pre-specified bounds $v_{\min}$ and $v_{\max}$; these bounds may be different for different regions of the workspace. The set of admissible control inputs is

$$U \;:=\; \{(a,\omega) : (v\omega/f_{\mathrm{r}}^{\max})^2 + (a/f_{\mathrm{t}}^{\max})^2 \leqslant 1\}, \qquad (14)$$

where $f_{\mathrm{r}}^{\max}$ and $f_{\mathrm{t}}^{\max}$ are pre-specified. The input constraint defined by (14) is an example of a “friction ellipse” constraint that models the limited tire frictional forces available for acceleration and steering of the vehicle. Finally, we denote by $v_j^{\max}$ and $v_j^{\min}$ pre-specified bounds on the vehicle speed inside the cell corresponding to the vertex $j \in V$.

We may now compute the effective target sets for this vehicle model as follows. We may transform, as in [22], the input constraint (14) to a strict inequality by adding to the L.H.S. of the inequality a small positive quantity $\varepsilon^2$, where $0 < \varepsilon \ll 1$. The tightened constraint implies that acceleration of the vehicle with $|\dot{v}| \geqslant \varepsilon f_{\mathrm{t}}^{\max}$ is always feasible. It follows that the upper and lower bounds for the vehicle speed $v$ at each of the boundaries of adjacent cells in the tile are

$$\begin{aligned} \overline{v}_k &= \min\{v_{j_k}^{\max}, v_{j_{k+1}}^{\max}, \sqrt{\overline{v}_{k+1} + 2\varepsilon f_{\mathrm{t}}^{\max} d}\}, \\ \underline{v}_k &= \max\{v_{j_k}^{\min}, v_{j_{k+1}}^{\min}, \sqrt{\underline{v}_{k+1} - 2\varepsilon f_{\mathrm{t}}^{\max} d}\}, \end{aligned}$$

whenever the cell corresponding to $j_k$ involves traversal across parallel edges, and by $\overline{v}_k = \min\{v_{j_k}^{\max}, \overline{v}_{k+1}\}$, $\underline{v}_k = \min\{v_{j_k}^{\min}, \underline{v}_{k+1}\}$, whenever the cell corresponding to $j_k \in V$ involves traversal across adjacent edges. The upper bound $\kappa_k^{\max}$ on the curvature of paths traversing the cell corresponding to $j_k \in V$ for $k = 1, \ldots, H-1$, is, by (13),

$$\kappa_k^{\max} \;=\; f_{\mathrm{r}}^{\max}\sqrt{1-\varepsilon^2}/(\max\{\overline{v}_k, \overline{v}_{k+1}\})^2. \qquad (15)$$

The bound (15) on the curvature of feasible paths is conservative because the bound on the vehicle speed in the denominator does not involve the initial speed $v_0$, i.e., the maximum reachable speed within each of the cells in the tile may be lower than $\max\{\overline{v}_k, \overline{v}_{k+1}\}$, and may be a less conservative bound on the speed (and consequently, on the curvature). A heuristic approximation to the maximum reachable speed may be obtained by considering maximum acceleration along the longest linear path within the cell (i.e., the diagonal of length $\sqrt{2}d$). Thus, a less conservative, heuristic bound on the curvature is given by

$$\kappa_k^{\max} \;=\; \frac{f_{\mathrm{r}}^{\max}\sqrt{1-\varepsilon^2}}{(\min\{\max\{\overline{v}_k, \overline{v}_{k+1}\}, \sqrt{v_0^2 + 2\sqrt{2} f_{\mathrm{t}}^{\max} d}\})^2}.$$

### C. Illustrative Example: Aircraft Navigational Model

Consider an aircraft navigational model described by

$$\begin{aligned} \dot{x}(t) &= v(t)\cos\gamma(t)\cos\psi(t), \\ \dot{y}(t) &= v(t)\cos\gamma(t)\sin\psi(t), \\ \dot{z}(t) &= v(t)\sin\gamma(t), \\ \dot{\psi}(t) &= -q(t)C_{\mathrm{L}}(t)/mv(t)\cos\gamma(t), \\ \dot{v}(t) &= \left(T(t) - q(v(t))C_{\mathrm{D},0} - KC_{\mathrm{L}}^2(t)\right)/m, \\ \dot{\gamma}(t) &= \left(q(v(t))C_{\mathrm{L}}(t)\cos\phi(t) - mg\cos\gamma(t)\right)/mv(t), \end{aligned}$$

where $x, y$, and $z$ denote the inertial position coordinates, $v$ denotes the speed, $\psi$ denotes the aircraft heading, $\gamma$ denotes the flight path angle, $q(v) := \frac{1}{2}\rho v^2 S$ denotes the dynamic pressure, $m$ denotes the mass of the aircraft, and $C_{\mathrm{D},0}$ and $K$ are pre-specified constants. The control inputs are the thrust $T$, the lift coefficient $C_{\mathrm{L}}$, and the bank angle $\phi$.

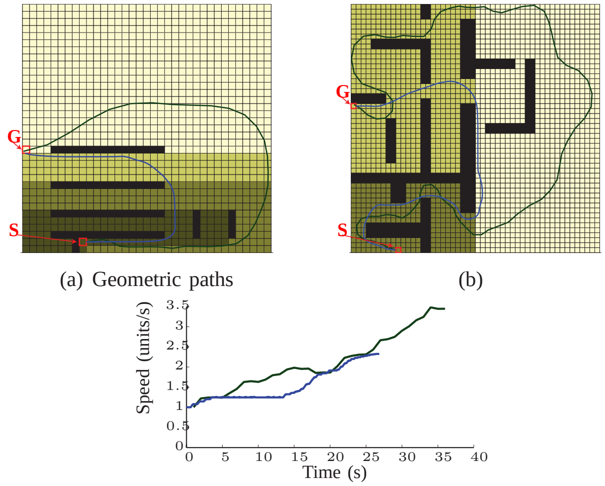
We consider the motion of the aircraft in the horizontal plane, i.e., $\gamma(t) = 0$ and $\dot{\gamma}(t) = 0$, and to this end we set

$$C_{\mathrm{L}}(t) = mg/(q(v(t))\cos\phi(t)).$$

We may assume the aircraft’s cruise speed to be a constant $v_{\mathrm{cr}}$. The thrust input is then given by

$$T(v_{\mathrm{cr}}, \phi(t)) \;=\; q(v_{\mathrm{cr}})C_{\mathrm{D},0} - K((mg/(q(v_{\mathrm{cr}})\cos\phi(t))^2.$$

Alternatively, we may assume a constant thrust input of value $T(v_{\mathrm{cr}}, 0)$, and allow small decreases in the aircraft speed during turning flight. In either case, the upper bound on the curvature, by (13), is given by $\kappa_k^{\max} = g\tan(\min|\phi_{\min}|, |\phi_{\max}|)/v_{\mathrm{cr}}$, for $k = 1, \ldots, H-1$.

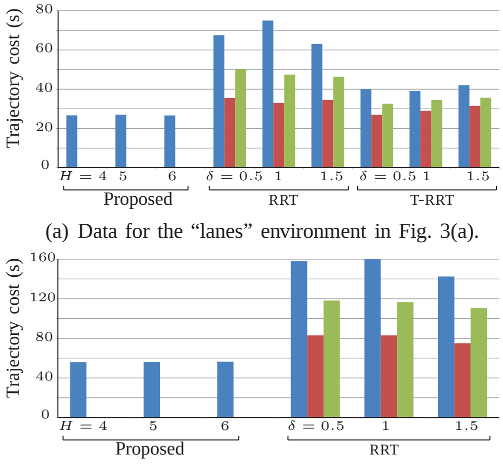


(a) Geometric paths

(b)

(c) Speed profiles

Fig. 3. The colored areas represent different speed limits: $v_{\max} = 1.25$ units/s for the darkest area, $v_{\max} = 2$ units/s, $v_{\max} = 2.5$ units/s, and $v_{\max} = 3.5$ units/s, respectively, for progressively lighter areas.



(a) Data for the "lanes" environment in Fig. 3(a).

(b) Data for the maze-like environment in Fig. 3(b).

Fig. 4. Comparison of trajectory costs: for the RRT and T-RRT data, the blue (left), red (middle), and green (right) bars represent, respectively, the maximum, the minimum, and the average values over 30 trials.

## V. Simulation Results and Discussions

In this section, we present numerical simulation results that show that the overall motion planner enabled by TilePlan results in trajectories of significantly lower cost compared to randomized sampling-based algorithms based on RRTs [23]. We compared the proposed motion planner against the standard RRT-based planner [23], and the T-RRT planner recently reported in [24] for planning minimum-time trajectories for the particle dynamical model[1] in Section IV-B. The T-RRT planner finds low-cost trajectories with respect to a pre-specified state space cost map. As the minimum-time criterion cannot be expressed as a state space cost map, we executed the T-RRT planner with the objective "travel as fast as possible," which is immediately defined by the state space cost map $c(\xi) = v$. For extending known states towards randomly selected new states, we programmed the RRT-based planners to randomly select an input vector from the set of admissible inputs and integrate the vehicle model for a fixed time $\delta$, as recommended in [23]. We conducted 30 trials of the standard RRT and T-RRT algorithms for different values of $\delta$, and we compared the results with the proposed algorithm on the same environment with three different values of $H$.

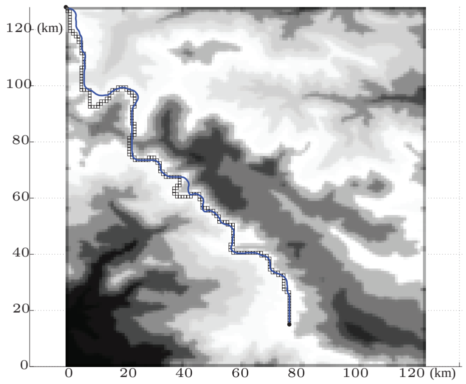


Fig. 5. The blue curve corresponds to the resultant state trajectory, while the channel of cells in black is the result of path planning without vehicle dynamical constraints. The initial position is at the top left corner.

Figures 3(a) and 3(b) show the two environments that we used for the numerical simulations. The black colored regions represent obstacles, and the other colors indicate different bounds on the speed of the vehicle. The blue curve in Figs. 3(a) and 3(b) corresponds to the trajectory found by the proposed algorithm.

The green curve in Fig. 3(a) corresponds to a sample path found by the T-RRT planner. This example illustrates that the "travel as fast as possible" objective is not always a satisfactory alternative to the minimum-time criterion: Figure 3(c) shows that the vehicle achieves higher speeds along the T-RRT trajectory but the travel time is $35.2\%$ higher than the trajectory found by the proposed planner. This result is a consequence of the input constraint (14), which forces the vehicle to traverse paths of lower curvature at higher speeds, thus producing longer geometric paths. Figure 4(a) shows comparative data for the trajectory costs (i.e., time of traversal) resulting from the simulations described above. The proposed motion planner returned trajectories with almost identical costs for each $H$, in particular, the trajectory cost corresponding to $H = 6$ was $26.626$ s. On the other hand, both the standard RRT and T-RRT planners returned, on an average, significantly costlier trajectories.

The green curve in Fig 3(b) corresponds to a sample trajectory found by the standard RRT motion planner[2]. Note that this environment has a narrow "short-cut" between the initial cell and the goal cell. Figure 4(b) shows comparative data for the trajectory costs for this environment. The proposed motion planner returned trajectories with almost identical costs for each $H$; in particular, the trajectory cost corresponding to $H = 5$ was $56.23$ s. The trajectory costs returned by

[1] The "friction ellipse" parameters were fixed: $f_{\mathrm{r}}^{\max} = 1$, $f_{\mathrm{t}}^{\max} = 0.25$.

[2] The T-RRT planner was found to be impractically slow for this example.

the standard RRT planner were significantly higher, mainly because it failed to traverse the aforementioned "short-cut" on several occasions, as illustrated in Fig. 3(b). Clearly, the average costs of trajectories returned by RRT-based planners may be further worsened in environments where the differences between the costs of trajectories corresponding to "short-cuts" and the costs of alternative trajectories is larger.

Finally, Fig. 5 shows the result of simulating the overall motion planner using the aircraft navigational model discussed in Section IV-C with $C_{\mathrm{D},0} = 0.02$, $K = 0.04$, $S = 30$ m$^2$, $mg = 50$ kN, and $v_{\mathrm{cr}} = 85$ m/s. The aircraft speed was assumed to be constant, and the limits on the bank angle control input were set to $\phi_{\min} = -45°$ and $\phi_{\max} = 20°$. The objective was to minimize a cost defined on the workspace (indicated by regions of different intensities in Fig. 5; the darker regions correspond to higher costs).

The preceding simulations were all implemented in the MATLAB simulation environment. Therefore, accurate indications of the computation time of the proposed planner in a real-time implementation are not yet available. However, the reader may refer [20] for comments on implementation-independent performance indicators of the proposed work.

The RRT$^*$ algorithm [25] is a recent development in randomized sampling-based optimal kinodynamic motion planning, and a thorough comparison of the proposed work to RRT$^*$ for motion planning with complex vehicle dynamical models is currently under investigation. The primary challenge in implementing RRT$^*$ for complex dynamical models is the development of an asymptotically optimal point-to-point steering algorithm for the given dynamical model, which is, in general, a more difficult problem than the tile motion planning problem discussed in this work.

## VI. Conclusions and Future Work

We presented a MPC-based local trajectory generation scheme, called TilePlan, to enable a hierarchical motion planner that incorporates vehicle kinematic and dynamic constraints in the geometric planning stage. The proposed TilePlan scheme relies on the idea of effective target sets to transform non-convex state constraints into convex constraints. We illustrated the proposed scheme using two non-trivial examples of vehicle dynamical models. Also, we demonstrated the efficacy of the overall motion planner via numerical simulation results that show significantly lower costs of resultant trajectories as compared to state-of-the-art randomized sampling-based planners. Future work includes applications of the proposed TilePlan scheme to more complex vehicle models and multi-resolution implementations of the overall motion planner.

**Acknowledgement:** This work has been supported in part by ARO MURI award W911NF-11-1-0046.

## References

[1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press, 2005.

[2] R. V. Cowlagi and P. Tsiotras, "Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles," *IEEE Transactions on Robotics*, 2011, to appear.

[3] ——, "On the existence and synthesis of curvature-bounded paths inside nonuniform rectangular channels," in *Proceedings of the 2010 American Control Conference*, Baltimore, MD, USA, 30 June – 2 July 2010, pp. 5382 – 5387.

[4] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, July 1957.

[5] F. Allgöwer and A. Zheng, Eds., *Nonlinear Model Predictive Control*. Berlin, Germany: Birkhäuser, 2000.

[6] J. M. Maciejowski, *Predictive Control with Constraints*. Pearson Education Inc., 2002.

[7] J. Bellingham, A. Richards, and J. P. How, "Receding horizon control of autonomous aerial vehicles," in *Proceedings of the 2002 American Control Conference*, Anchorage, AK, May 8 – 10 2002, pp. 3741 – 3746.

[8] A. Richards and J. P. How, "Aircraft trajectory planning using mixed integer linear programming," in *Proceedings of the 2002 American Control Conference*, Anchorage, AK, May 8 – 10 2002, pp. 1936 – 1941.

[9] Y. Kuwata and J. P. How, "Stable trajectory design for highly constrained environments using receding horizon control," in *Proceedings of the 2004 American Control Conference*, Boston, MA, June 30 – July 2 2004, pp. 902 – 907.

[10] ——, "Three dimensional receding horizon control for UAVs," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, RI, August 16 – 19 2004.

[11] R. Prazenica, A. Kurdila, R. Sharpley, and J. Evers, "Vision-based geometry estimation and receding horizon path planning for UAVs operating in urban environments," in *Proceedings of the 2006 American Control Conference*, Minneapolis, MN, June 14 – 16 2006, pp. 2874 – 2879.

[12] Y. Yoon, J. Shin, H. Jin Kim, and S. Sastry, "Model-predictive active steering and obstacle avoidance for autonomous ground vehicles," *Control Engineering Practice*, vol. 17, pp. 741 – 750, 2009.

[13] T. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 141 – 166, February 2007.

[14] B. Xu, D. J. Stilwell, and A. Kurdila, "A receding horizon controller for motion planning in the presence of moving obstacles," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, May 3 – 8 2010, pp. 974 – 979.

[15] A. Richards and J. P. How, "Robust variable horizon model predictive control for vehicle maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 16, pp. 333 – 351, 2006.

[16] Y. Kuwata and J. P. How, "Robust cooperative decentralized trajectory optimization using receding horizon MILP," in *Proceedings of the 2007 American Control Conference*, New York, NY, July 11 – 13 2007, pp. 522 – 527.

[17] D. P. Bertsekas and I. B. Rhodes, "On the minimax reachability of target sets and target tubes," *Automatica*, vol. 7, pp. 233–247, 1971.

[18] S. Winter, "Modeling costs of turns in route planning," *GeoInformatica*, vol. 6, no. 4, pp. 345 – 361, 2002.

[19] E. Rippel, A. Bar-Gill, and N. Shimkin, "Fast graph-search algorithms for general aviation flight trajectory generation," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 801–811, July-August 2005.

[20] R. V. Cowlagi, "Hierarchical motion planning for autonomous aerial and terrestrial vehicles," Ph.D. dissertation, Georgia Institute of Technology, 2011.

[21] E. Kreyszig, *Differential Geometry*. New York: Dover Publications, Inc., 1991.

[22] E. Velenis and P. Tsiotras, "Minimum-time travel for a vehicle with acceleration limits: Theoretical analysis and receding horizon implementation," *Journal of Optimization Theory and Applications*, vol. 138, no. 2, pp. 275–296, 2008.

[23] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[24] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 647–659, 2010.

[25] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, pp. 846–894, 2011.