Homotopy-Based Divide-and-Conquer Strategy for Optimal Trajectory Planning via Mixed-Integer Programming

Junghee Park, Sisir Karumanchi, and Karl Iagnemma

***Abstract*—This paper proposes an optimal trajectory generation framework in which the global obstacle-avoidance problem is decomposed into simpler subproblems, corresponding to distinct path homotopies. In classical approaches to homotopic trajectory planning, trajectory planning and homotopy identification are performed simultaneously, leading to a substantial computational burden. The main benefit of the proposed approach is the development of a method to enumerate and explicitly represent distinct homotopy classes before trajectory planning or optimization, which allow the problem to be decomposed into simpler independent subproblems. The main contribution of the paper is twofold. The first contribution is the description of a method for utilizing existing cell-decomposition methods to enumerate and represent local trajectory generation problems that can be solved efficiently and independently. In addition, a relationship between the proposed cell-sequence representation and homotopy classes is analyzed. The second contribution is a computationally efficient novel formulation of the trajectory optimization problem within a cell sequence via mixed-integer quadratic programming (MIQP). Computational efficiency and increased solution richness of the proposed approach are demonstrated through simulation studies. The proposed MIQP formulation fits into a linear model-predictive control framework with nonconvex collision-free constraints.**

***Index Terms*—Mixed-integer programming (MIP), model-predictive control (MPC), motion control, nonholonomic motion planning.**

I. Introduction

Motion planning has been extensively investigated over the past several decades and has a vast range of applications. A variety of algorithms to generate a feasible (or optimal) trajectory have been developed and can be sorted into two main types: combinatorial planning methods and sampling-based planning methods [1]. Although combinatorial approaches can be computationally expensive, combinatorial algorithms can provide elegant and practical solutions [2] to problems with certain convenient properties (i.e., low-dimensional models). This paper proposes a divide-and-conquer strategy for combinatorial approaches to optimal kinodynamic motion planning problem.

Optimal trajectory generation is typically a challenging task even for problems employing low-dimensional system models. In particular, for the collision-avoidance problem, the presence of obstacles serves to create holes in the collision-free configuration space, leading to discontinuities in the set of feasible trajectories and disjunctive choices in the search procedure. This increases the difficulty of an optimization-based solution. However, the complexity of the obstacle-avoidance problem can be reduced significantly if we independently consider multiple distinct continuous sets of feasible trajectories, where no discontinuity is presented. In the simple example of a single polygonal obstacle in 2-D environment, it is a straightforward task to find two (local) optimal trajectories among a set of trajectories that avoid the obstacle in either a counterclockwise or clockwise manner, since each respective search space is continuous. The proposed approach in this paper exploits this observation by decomposing a general motion planning problem into multiple independent problems, each with simple obstacle-avoidance constraints, and thereby achieves computational benefits arising from a divide-and-conquer strategy.

In addition to computational benefits, it is valuable in some applications to identify multiple locally optimal solutions corresponding to distinct navigation decisions. For example, in unmanned vehicle navigation problems with human operators in the loop (i.e., in which the operator provides some high-level input related to vehicle navigation decisions), it can be desirable to identify and present multiple choices corresponding to qualitatively distinct vehicle routes. Distinct navigation decisions often bear correspondence to the topological notion of distinct *homotopy classes* in low-dimensional cases. A homotopy class is loosely defined as a set of trajectories that can be continuously deformed into each other without intersecting an obstacle.

The role of homotopy classes in robotic navigation has been studied in [3]. The utility of homotopy class analysis is clear in certain applications such as exploration and mapping, or multi-agent task planning [4]. In addition, in semiautonomous robotic applications, in which a human and control system share control of the robotic system (e.g., see, [5]), it can be important to identify and respect a human operator's intent, expressed through his/her navigation decision, corresponding to the choice to navigate within a particular homotopy class.

Fig. 1 illustrates a divide-and-conquer strategy to address the trajectory generation problem. The proposed method draws correspondence between (typically) multiple local trajectory

Manuscript received November 18, 2014; revised May 13, 2015; accepted July 15, 2015. Date of publication August 7, 2015; date of current version September 30, 2015. This paper was recommended for publication by Associate Editor E. Yoshida and Editor C. Torras upon evaluation of the reviewers' comments. This work was supported by the U.S. Army Research Office under Contract W911NF-11-1-0046 and by the STX Scholarship Foundation.

J. Park and K. Iagnemma are with the Robotic Mobility Group, Laboratory for Manufacturing and Productivity, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: junghee@mit.edu; kdi@mit.edu).

S. Karumanchi is with the Manipulation and Sampling Group, Mobility and Robotic Systems Section, NASA Jet Propulsion Laboratory, Pasadena, CA 91109 USA (e-mail: sisir.b.karumanchi@jpl.nasa.gov).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2015.2459373

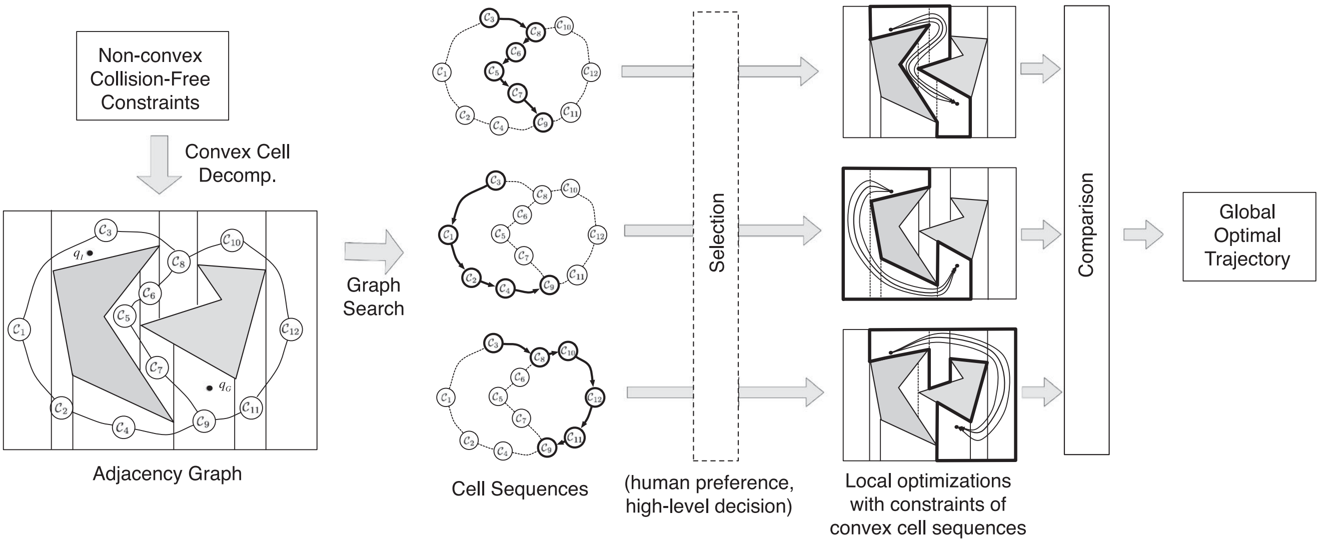
Non-convex Collision-Free Constraints
Convex Cell Decomp.
$\mathcal{C}_1$ $\mathcal{C}_2$ $\mathcal{C}_3$ $\mathcal{C}_4$ $\mathcal{C}_5$ $\mathcal{C}_6$ $\mathcal{C}_7$ $\mathcal{C}_8$ $\mathcal{C}_9$ $\mathcal{C}_{10}$ $\mathcal{C}_{11}$ $\mathcal{C}_{12}$
q_I
q_G
Adjacency Graph
Graph Search
Cell Sequences
Selection
(human preference, high-level decision)
Local optimizations with constraints of convex cell sequences
Comparison
Global Optimal Trajectory

Fig. 1. Divide-and-conquer strategy of optimal trajectory generation.

problems that correspond to distinct homotopy classes. This results in an intuitive representation of homotopy classes as sequences of spatial constraints, which in turn leads to a hierarchical framework: problem decomposition and constraint identification based on homotopy enumeration, followed by local trajectory generation within each constrained region.

In the homotopy enumeration step, the collision-free space is decomposed into convex cells, and each homotopy class is represented as a sequence of convex cells. Local trajectory optimization within each homotopy class is an independent problem and corresponds to a distinct navigation decision. This yields a framework in which multiple local problems, corresponding to an operator's preference (i.e., distinct high-level decisions) are independently solved. For the local trajectory-generation problem, we propose a mixed-integer programming (MIP) formulation, where control inputs are represented as continuous optimization variables and the time steps associated with transitions between adjacent cells are represented as discrete variables. The framework allows access to not only the global optimal solution, but also an optimal solution for each identified homotopy class.

The remainder of this paper is organized as follows. In the next section, previous works related to the proposed approach are described. In Section III, notations and problem definition are presented. Section IV discusses the properties of cell decomposition, and Section V formulates the local trajectory optimization problem subject to homotopy constraints as an MIP problem and demonstrates the computational efficiency of such a method compared with a global solution method. In Section VI, simulation results are presented for a simple case of a point mass model, as well as for the more practical application of autonomous passenger vehicle navigation on a roadway operating under linear model-predictive control (MPC).

II. Related Works

Many different approaches exist for solving the motion planning problem, such as potential fields [6], graph search methods [7], probabilistic roadmaps [8], and many more. As noted previously, combinatorial planning methods and sampling-based planning represent two broad algorithm classes. For combinatorial algorithms, methods for exact cell decomposition represent a key algorithm component. Cell decomposition methods partition the free configuration space into a finite set of regions, which can be exploited for construction of roadmaps. Cell decomposition methods should satisfy the following three basic properties to be useful for roadmap construction [2]: 1) trivial computation of a path from one point to another inside a cell; 2) easy extraction of cell adjacency information; and 3) efficient determination of cells containing start and goal configurations. If a cell decomposition method satisfies these properties, then the motion planning problem is easily reduced to a graph search problem. Typical usage of cell decomposition corresponds to geometric path planning, which reasons about connectivity and ignores dynamics and feasibility. In contrast, the work described in this paper uses cell decomposition results as inputs to solve a set of efficiently defined kinodynamic motion planning problems. For example, a sequence of decomposed cells is used to pose the boundaries and provide guidance of movement for solving kinodynamic motion planning problems. We further investigate additional properties of cell decomposition for the proposed homotopy-based divide-and-conquer approach.

In recent years, some motion planning methods have been proposed that exploit homotopy class knowledge. In [3], these methods are surveyed and categorized into three groups. The first group finds the shortest path from a start to goal region when homotopy information is known. Such methods take a path or constrained area as an input and then find the shortest path within the current homotopy. The second group computes the shortest path from start to goal region and then identifies the homotopy class to which the solution belongs. The identified topology of the homotopy class that contains the global optimal solution can be used to reduce the space of future search calls. By repeating this approach, it is possible to obtain k-shortest paths in distinct homotopy classes. Finally, the third group first enumerates homotopy classes in a given environment and then searches for a path that is contained within each homotopy class. This approach relies on specialized data structures in order to systematically describe the topological properties of the environment. Enumerating homotopy classes before performing path planning allows for independent treatment of the path-planning problem within each distinct homotopy.

The approach described in this paper is similar to the methods of the third group in that it aims to utilize an efficient divide-and-conquer strategy. Exact cell decomposition and graph search are proposed as methods to systematically describe the topological properties of the environment and then to enumerate homotopy classes as sequences of cells. The proposed method builds on the work of [9], where constrained Delaunay triangulation and graph search methods were used to find a desirable safe corridor to achieve safe vehicle navigation. For a particular corridor, spatial bounds were identified to provide collision-avoidance assurance, and these bounds were employed as constraints in an MPC framework for a resulting 1-D system. In such systems, once the desired homotopy class is specified, the corresponding constraint form is convex, and thus, the linear model-predictive controller can exploit a quadratic programming (QP) formulation. This paper generalizes this idea and further explores the relationship between sequences of decomposed cells and homotopy classes. Although the spatial constraints corresponding to homotopy classes are nonconvex in a 2-D environment, this paper shows that the optimization problem can be solved efficiently through the proposed formulation of MIP, by exploiting cell adjacency relationships.

We note that MIP has been widely used to solve trajectory optimization problems. Although MIP is an NP-complete problem [10], tractable anytime solution algorithms have been developed. Richards *et al.* [11] proposed a formulation of the optimal trajectory generation problem as an MIP to handle the nonconvexity of the collision-free constraint. A key idea lies in expressing nonconvex polygons containing a convex polygonal hole as a union of half spaces and then ensuring satisfaction of these half space constraints using binary variables. This study was extended to the problem of regulating senses of conflict resolution between agents by adding constraints on binary variables [12]. A similar idea was proposed for trajectory generation under homotopy class constraints by introducing binary variables that encode homotopy class information [13]. Here, we present an efficient novel formulation of the MIP problem by employing binary variables for the purpose of satisfaction of spatial constraint sequences. In this formulation, the only independent binary variables are associated with transition times between cells; therefore, a significant reduction of the search space of binary variables is achieved.

III. Problem Definition

A. Notations

In this section, we define notations of sets and associated operations that are used in this paper. All sets are defined as closed sets unless otherwise specified. Let P be a set in $\mathbb{R}^2$ space, $P \subset \mathbb{R}^2$, and let ∂P be the boundaries of P. $P^\circ = P \backslash \partial P$ denotes the open set of P, where $\backslash$ is the set subtraction operation. For consistency, we define the closed collision-free space, $\bar{\mathcal{C}}_{\text{free}} = \mathbb{R}^2 \backslash (\cup_i \mathcal{C}_{\text{obs}}{}_i^\circ)$, where $\mathcal{C}_{\text{obs}i}$ is a set of configurations leading to collision with obstacle i.[1] A sequence of cell elements $\mathcal{C}_i$ is expressed as $\{\mathcal{C}_0 \rightarrow \mathcal{C}_1 \rightarrow \mathcal{C}_2 \rightarrow \cdots\}$ in which the order of elements implies the sequence. Let $\vec{\cup}$ be the union of sets where the sequence is preserved, for example, $\{\mathcal{C}_0 \rightarrow \mathcal{C}_1\} \vec{\cup} \{\mathcal{C}_2 \rightarrow \mathcal{C}_3\} = \{\mathcal{C}_0 \rightarrow \mathcal{C}_1 \rightarrow \mathcal{C}_2 \rightarrow \mathcal{C}_3\}$. Let $\text{vertices}(\mathcal{C})$ denote vertices of polygon $\mathcal{C}$. A common edge between two adjacent cells $\mathcal{C}_i$ and $\mathcal{C}_j$ is denoted by $E(\mathcal{C}_i, \mathcal{C}_j)$. In the description of MIP, the range of integers is often introduced and specified with subscripts, for example, we denote integers from 1 to n by $\mathbb{Z}_{1,n}$, i.e., $\mathbb{Z}_{1,n} = \{1, 2, \ldots, n\}$

B. Problem Definition

The objective of the optimal kinodynamic motion planning problem is to generate an optimal input profile minimizing a desired cost function and satisfying both differential and collision-free constraints. The optimization problem is formulated as follows:

$$\min_{\mathbf{u}(\tau)} J = \int_{t_0}^{t_f} g\left(\mathbf{x}(\tau), \mathbf{u}(\tau)\right) d\tau \tag{1}$$

$$\text{subject to } \dot{\mathbf{x}}(\tau) = \mathbf{f}\left(\mathbf{x}(\tau), \mathbf{u}(\tau)\right) \tag{2}$$

$$\mathbf{u}(\tau) \in \Omega \tag{3}$$

$$\mathbf{q}(\tau) \in \mathcal{C}_{\text{free}} \text{ for } \tau \in [t_0, t_f] \tag{4}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \ \mathbf{x}(t_f) = \mathbf{x}_f \tag{5}$$

where $\mathbf{u}(\tau)$ is the control input, $\mathbf{x}(\tau)$ is the robot state, and $\mathbf{x}_0$ and $\mathbf{x}_f$ are given start and goal state. Equation (2) is a state-transition equation in which kinematic and dynamic constraints of a robot are specified through a state vector, augmented vector of configurations $\mathbf{q}(\tau)$, and velocities $\dot{\mathbf{q}}(\tau)$, $\mathbf{x}(\tau) = (\mathbf{q}(\tau), \dot{\mathbf{q}}(\tau))$. We assume $\mathcal{C}_{\text{obs}}$ can be approximated by polygonal shapes and restrict our focus to the 2-D Euclidean space $\mathbb{R}^2$, where analyses of homotopy classes are intuitive and have various practical applications. Here, $\mathcal{C}_{\text{free}} \subset \mathbb{R}^2$ is a polygonal space with holes, which can also be thought as a projection of a multidimensional configuration space onto a 2-D Euclidean workspace. An example application is a ground vehicle operating in 2-D space having three configurations: the location of a reference point(x, y) and heading angle θ, where heading angle θ does not affect the analysis of trajectories and only the projection onto the (x, y) plane is enough to capture homotopy class membership information [14].

We note that the primary difficulty in solving the optimization problem outlined above arises from the collision-free constraints embodied in (4), since the collision-free space $\mathcal{C}_{\text{free}}$ is generally nonconvex. Information about collisions with obstacles is mapped into a configuration space in the form of $\mathbf{q}(\tau) \notin \mathcal{C}_{\text{obs}}$, and collision-free configurations are expressed as a complementary set of configurations. As a result, the collision-free

[1] Note that the real collision-free space $\mathcal{C}_{\text{free}}$ is the open set of $\bar{\mathcal{C}}_{\text{free}}$, i.e., $\mathcal{C}_{\text{free}} = \bar{\mathcal{C}}_{\text{free}}^\circ$, but the notion of the closed set of collision-free space is used in the procedure of cell decomposition.

configuration space can be nonconvex since it contains holes, which makes the set of collision-free trajectory candidates discontinuous. However, we also note that these search space discontinuities give rise to distinct navigation decisions and their associated homotopy classes.

Based on this observation, the method described in this paper aims to represent homotopy classes as continuous spatial constraints and then utilizes a divide-and-conquer strategy where the original nonconvex optimization problem is divided into multiple independent convex problems. Each independent problem, in addition to having direct association with an independent navigation decision, is also relatively easy to solve. In fact, for the problem of finding minimum-length paths in 2-D Euclidean space, Chazelle [15] showed that the shortest path inside a simple polygon can be computed in $O(N \log N)$ time and $O(N)$ space, where N is the number of vertices of the simple polygon. In contrast, the shortest path in general polygonal space with holes requires $O(N^2 \log N)$ time and $O(N^2)$ space.

IV. Divide-and-Conquer Strategy Based on Cell Decomposition

A. Convex Decomposition and Cell Adjacency

The first step of the proposed divide-and-conquer strategy is to decompose $\bar{\mathcal{C}}_{\text{free}}$ into convex polygons based on well-known convex decomposition algorithms. Then, adjacency relationships between decomposed polygons can be employed to represent adjacencies as a graph. The start node on the graph is determined by the current location of the robot. Once a goal node is specified, it is possible to enumerate all possible paths connecting the start and goal pair via standard graph search methods. A path on the graph can be associated with a set of trajectories following the sequence of convex polygons. In addition, it is shown that a loopless path on the graph corresponds to a homotopy class in the following sections.

Definition 1 (Convex Decomposition [16]): A set of convex components $\{\mathcal{C}_i\}$ is a convex decomposition of $\bar{\mathcal{C}}$, $D(\bar{\mathcal{C}})$, if their union is $\bar{\mathcal{C}}$ and all $\mathcal{C}_i$ are interior disjoint, i.e., $D(\bar{\mathcal{C}}) = \{\mathcal{C}_i | \cup_i \mathcal{C}_i = \bar{\mathcal{C}} \text{ and } \forall_{i \neq j} \mathcal{C}_i{}^{\circ} \cap \mathcal{C}_j^{\circ} = \emptyset\}$

Since the system of interest is assumed to be operating in a polygonal space, the components of a convex decomposition of the closed collision-free space $\{\mathcal{C}_i\} = D(\bar{\mathcal{C}}_{\text{free}})$ are convex polygons. We restrict our interest to decompositions that do not create any new vertices except for vertices of the original polygons. This is because new vertices that are not part of obstacles lead to singularities in correspondence between a cell-sequence representation and homotopy classes, a problem that is discussed in detail in the following sections. We also note that this requirement eliminates the case where a feasible trajectory passes through a vertex of a polygon, as shown in Fig. 2(a).

Definition 2 (Convex Decomposition with the Minimal Vertex Set): A set of convex components $\{\mathcal{C}_i\}$ is a convex decomposition of a polygonal space $\bar{\mathcal{C}}$ with the minimal vertex set, $\{\mathcal{C}_i\} = D_{\text{mv}}(\bar{\mathcal{C}})$, if it is a convex decomposition and all vertices of components are vertices of the original polygon, i.e., $\forall_i \text{vertices}(\mathcal{C}_i) \subseteq \text{vertices}(\bar{\mathcal{C}})$.

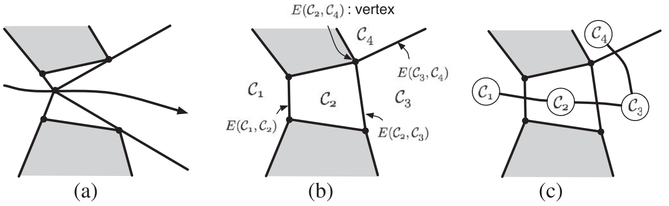
$E(\mathcal{C}_2, \mathcal{C}_4)$: vertex
$\mathcal{C}_4$
$E(\mathcal{C}_3, \mathcal{C}_4)$
$\mathcal{C}_1$
$\mathcal{C}_2$
$\mathcal{C}_3$
$E(\mathcal{C}_1, \mathcal{C}_2)$
$E(\mathcal{C}_2, \mathcal{C}_3)$
$\mathcal{C}_4$
$\mathcal{C}_1$
$\mathcal{C}_2$
$\mathcal{C}_3$
(a)
(b)
(c)

Fig. 2. (a) Singularity of convex decomposition with an additional (i.e., non-minimal) vertex. (b) Convex decomposition with the minimal vertex set. (c) Adjacency relation defined according to the existence of open set of common edge between two different cells.

Many convex decomposition algorithms fall into this category such as well-known trapezoidal decomposition [17] or constrained Delaunay triangulation [18]. Based on decomposition with the minimal vertex set, we construct the adjacency relation graph between the decomposed cells. Since we assumed closed sets, two adjacent cells $\mathcal{C}_i$ and $\mathcal{C}_j$ share a common edge $E(\mathcal{C}_i, \mathcal{C}_j)$. However, we rule out adjacency between two cells sharing a point since it leads to redundancy; there are no collision-free vertices in the decomposed space according to Definition 2. For example, in Fig. 2(c), $\mathcal{C}_2$ and $\mathcal{C}_4$ are not connected in the graph since the common edge is only a point.

Definition 3 (Adjacency Graph): The adjacency graph $G = (V_G, E_G)$ of decomposition $D_{\text{mv}}(\bar{\mathcal{C}})$ is the graph where each node represents a decomposed convex polygon, $V_G = \{\mathcal{C}_i\}$, and the edges are connected if two different cells have a common edge which is not a point, i.e., $E_G = \{e_{ij} | \forall_{i \neq j} E(\mathcal{C}_i, \mathcal{C}_j)^{\circ} \neq \emptyset\}$.

We assume that q_0 and $\mathbf{q}_f$ do not lie on edges $\{\forall_{i \neq j} E(\mathcal{C}_i, \mathcal{C}_j)\}$ so that the start node containing the start configuration and the goal node containing the goal configuration in the graph are uniquely determined and denoted by $\mathcal{C}_0$ and $\mathcal{C}_N$, respectively, i.e., $q_0 \in \mathcal{C}_0$ and $\mathbf{q}_f \in \mathcal{C}_N$.

Once the adjacency graph is constructed, it is straightforward to analyze the graph to identify sequences of cells linking desired configurations. Then, the original problem with collision-free constraints can be decomposed into multiple local problems with associated constraints resulting from sequences of bounded regions. The exact partition of a universal set of feasible trajectories associated with the global problem into an infinite number of sets of feasible trajectories associated with decomposed local problems is described in Section IV-B. For practical purposes, however, we suggest restricting local problems to limited set of sequences of cells. The benefits and limitations of this restriction are discussed in Sections IV-C and IV-D.

B. Partition of Universal Search Space Into Discrete Sets

This section shows that the search spaces of local problems disjointly comprise the original search space, the universal set of all feasible trajectories. In other words, it is shown that all feasible trajectories can be mapped to a sequence of cells on the adjacency graph such that the universal uncountably infinite set of feasible trajectories can be partitioned into a countably infinite set of sequences of cells on the graph. We define a mapping $\mathcal{F}_{\text{SC}}$ in Table I for the sake of clarity, although the correspondence of a feasible trajectory to a sequence of cells

TABLE I
DEFINITION OF MAPPING TO A CELL SEQUENCE $\mathcal{F}_{\mathrm{SC}} : \mathbf{q}(\tau) \rightarrow \mathcal{SC}$ AND TRANSITION TIMES $\mathcal{F}_{ST} : \mathbf{q}(\tau) \rightarrow \mathcal{ST}$

```
$\tau' \leftarrow t_0,\ w \leftarrow 0,\ \mathcal{ST} \leftarrow \emptyset,\ \mathcal{SC} \leftarrow \mathcal{C}_w$
while($w \neq N$)
  $\tau' \leftarrow \inf\{\tau \in [\tau', t_f] \mid \mathbf{q}(\tau) \notin \mathcal{C}_w\}$
  $\mathcal{ST} \leftarrow \mathcal{ST} \cup \{\tau'\}$
  $w \leftarrow w' \mid \lim_{\epsilon \rightarrow 0^+} \mathbf{q}(\tau' + \epsilon) \in \mathcal{C}_{w'}$
  $\mathcal{SC} \leftarrow \mathcal{SC} \vec{\cup} \{\mathcal{C}_w\}$
end while
```

is intuitive. In addition, let $\{\mathcal{SC}_j\}$ be the set of all possible sequence of cells on the adjacency graph.

Lemma 4: Any feasible trajectory can be mapped through $\mathcal{F}_{\mathrm{SC}}$ to a sequence of decomposed cells on the adjacency graph, i.e., $\mathcal{F}_{\mathrm{SC}}(\mathbf{q}(\tau)) = \mathcal{SC} \in \{\mathcal{SC}_j\}$

Proof: Let $\mathbf{q}(\tau) : [t_0, t_f] \rightarrow \mathcal{C}_{\mathrm{free}}$ be a feasible trajectory and a continuous function. Let $\mathcal{C}_w$ be the cell containing a point $\mathbf{q}(\tau_0)$ that does not lie on common edge $\{\forall_{i \neq j} E(\mathcal{C}_i, \mathcal{C}_j)\}$. The unique correspondence of $\mathbf{q}(\tau_0)$ to cell $\mathcal{C}_w$ is preserved for $\tau \in [\tau_0, \bar{\tau})$ until it touches the surrounding common edges $\{\forall_{w \neq j} E(\mathcal{C}_w, \mathcal{C}_j)\}$ at time $\tau = \bar{\tau}$. For the sake of unique correspondence, the mapping is $\mathcal{F}_{\mathrm{SC}}$ defined by assigning $\mathcal{C}_w$ to $\mathbf{q}(\tau)$ for $\tau \in [\bar{\tau}, \tau']$ until it leaves $\mathcal{C}_w$ at the time $\tau = \tau'$, and assigning a new cell once it leaves the cell $\mathcal{C}_w$. In addition, $\mathbf{q}(\tau')$ lies on an open common edge $E(\mathcal{C}_w, \mathcal{C}_j)^\circ$, since every vertices (i.e., intersections of three different cells) lead to collision from Definition 2. Thus, the next cell $\mathcal{C}_{w'}$ corresponding to $\lim_{\epsilon \rightarrow 0^+} \mathbf{q}(\tau' + \epsilon)$ is also unique and one of cells connected on the adjacent graph.

The following two corollaries follow directly from the above.

Corollary 5: The uncountably infinite set of all feasible trajectories, $\mathcal{T}\mathrm{raj}$, can be partitioned into countably infinite sets $\{\mathcal{T}raj_j\}$ where $\mathcal{T}\mathrm{raj}_j = \{\mathbf{q}(\tau) | \mathcal{F}_{\mathrm{SC}}(\mathbf{q}(\tau)) = \mathcal{SC}_j\}$, i.e., $\{\mathcal{T}\mathrm{raj}_j\}$ is a set of preimages of $\{\mathcal{SC}_j\}$, and thus, $\cup_j \mathcal{T}\mathrm{raj}_j = \mathcal{T}\mathrm{raj}$ and $\forall_{i \neq j} \mathcal{T}raj_i \cap \mathcal{T}raj_j = \emptyset$.

Corollary 6: A set of the optimal solutions among each partitioned trajectory sets $\mathcal{T}raj_j$ includes the global optimal solution.

Therefore, the universal set of all feasible trajectories can be partitioned into mutually exclusive and collectively exhaustive sets through the proposed cell sequence representation, as illustrated in Fig. 3(a) and (b). Thus, the global optimal trajectory can be eventually found based on the proposed divide-and-conquer strategy. However, since the partitioned sets are still infinite despite their countableness, we propose an approach that relies on solving restricted sets of local problems for practical purposes. This method is described in the following sections.

C. One-to-one Correspondence of Loopless Cell Sequences With Homotopy Classes

In this section, we restrict our interests to cell sequences containing no loops and investigate the relationship with topological homotopy classes. Fig. 3(c) shows the relationship between loopless sequences on the graph, their preimages in the trajectory set, and homotopy classes. In short, the loopless sequences have one-to-one correspondence with homotopy classes. In other words, a feasible trajectory is homotopic with any feasible trajectories corresponding to the same loopless sequence of cells on the graph, and not homotopic with all feasible trajectories in different sequences of cells in the graph. Fig. 4 shows examples of the proporties and singularities with non-minimum vertex set.

Theorem 7: If two feasible trajectories correspond to the same sequence of cells, they are homotopic.

Proof: Let $\mathbf{q}_A(\tau)$ and $\mathbf{q}_B(\tau)$ be feasible trajectories corresponding to the same sequence of cells $\mathcal{SC}$, i.e., $\mathcal{F}_{\mathrm{SC}}(\mathbf{q}_A(\tau)) = \mathcal{F}_{\mathrm{SC}}(\mathbf{q}_B(\tau)) = \mathcal{SC}$. For any cell $\mathcal{C}_w \in \mathcal{SC}$, there always exist corresponding segments of both trajectories, i.e., $\mathbf{q}_A(\tau)$ for $\tau_{Ai} < \tau \leq \tau_{Aj}$ and $\mathbf{q}_B(\tau)$ for $\tau_{Bi} < \tau \leq \tau_{Bj}$, where τ_{Xi} and τ_{Xj} are $(w-1)$th and wth elements of the set of transition times $\mathcal{ST}_{\mathcal{X}} = \mathcal{F}_{ST}(\mathbf{q}_X(\tau))$. These two trajectory segments can always be continuously deformed into each other due to the convexity of the decomposed cells. Therefore, it is always possible to find a continuous function $Q(\gamma, \tau) : [0,1] \times [0,1] \rightarrow \mathcal{C}_{\mathrm{free}}$ such that $Q(0, \tau) = \mathbf{q}_A(\tau)$ and $Q(1, \tau) = \mathbf{q}_B(\tau)$. □

This property is not restricted to loopless sequences of cells, but rather can be applied to all sequences of cells including sequences with loops. The converse, however, is not true for all sequences with loops. In other words, two trajectories corresponding to different sequences of cells with loops could be homotopic. Therefore, there exists a mapping from $\{\mathcal{T}\mathrm{raj}_j\}$ to $\{\mathcal{H}_i\}$, but it is not bijective, as shown in Fig. 3(b). If we restrict our focus to loopless sequences, however, the converse is true, so the mapping from $\{\mathcal{SC}_j^{\mathrm{nc}}\}$ to $\{H_i^{\mathrm{nc}}\}$ is bijective as shown in Fig. 3(c).

Theorem 8: If two feasible trajectories correspond to two different cell sequences, which are loopless in the adjacency graph constructed through a convex decomposition with the minimal vertex set, they are not homotopic.

Proof. Let feasible trajectories $\mathbf{q}_A(\tau)$ and $\mathbf{q}_B(\tau)$ correspond to $\mathcal{SC}_A = \mathcal{F}_{\mathrm{SC}}(\mathbf{q}_A(\tau)) = \{\mathcal{C}_0 \rightarrow \mathcal{C}_{A1} \rightarrow \mathcal{C}_{A2} \rightarrow \cdots \rightarrow \mathcal{C}_N\}$ and $\mathcal{SC}_B = \mathcal{F}_{\mathrm{SC}}(\mathbf{q}_B(\tau)) = \{\mathcal{C}_0 \rightarrow \mathcal{C}_{B1} \rightarrow \mathcal{C}_{B2} \rightarrow \cdots \rightarrow \mathcal{C}_N\}$, respectively, where $\forall_i \mathcal{C}_{Ai}$ are different, and $\forall_j \mathcal{C}_{Bj}$ are different, because both are loopless. In addition, there exists at least one cell contained only in either one of the sequences, i.e., $[\{\forall_i \mathcal{C}_{Ai}\} \cup \{\forall_j \mathcal{C}_{Bj}\}] - [\{\forall_i \mathcal{C}_{Ai}\} \cap \{\forall_j \mathcal{C}_{Bj}\}] \neq \emptyset$, because both are different sequences. Then, the closed loop $\mathbf{q}_{CL}(\tau)$ formed by $\mathbf{q}_A(\tau)$ in forward direction and $\mathbf{q}_B(\tau)$ in backward direction, $\mathbf{q}_{CL}(\tau) = \mathbf{q}_A(2\tau)$ if $\tau \leq t_f/2$, and $\mathbf{q}_A(2(t_f - \tau))$ if $\tau \geq t_f/2$, correspond to the sequence $\mathcal{SC}_{CL} = \mathcal{F}_{\mathrm{SC}}(\mathbf{q}_{CL}(\tau)) = \{\mathcal{C}_0 \rightarrow \mathcal{C}_{A1} \rightarrow \mathcal{C}_{A2} \rightarrow \cdots \rightarrow \mathcal{C}_N \rightarrow \cdots \mathcal{C}_{B2} \rightarrow \mathcal{C}_{B1} \rightarrow \mathcal{C}_0\}$ on the graph. Then, $\mathcal{SC}_{CL}$ has at least one loop consisting of at least three different cells; therefore, $\mathbf{q}_{CL}(\tau)$ encircles at least one obstacle because there are no collision-free common vertices of any three different cells according to Definition 2. Therefore, $\mathbf{q}_A(\tau)$ and $\mathbf{q}_B(\tau)$ are not homologous, and thus they are not homotopic.

D. Heuristic Strategies for Selection of Local Problems

The proposed divide-and-conquer strategy relies on solving decomposed local problems; however, there generally exists an infinite number of local problems (note that $\{\mathcal{SC}_j\}$ is an infinite set). Thus, for practical purposes, we aim to selectively solve a finite number of problems or prioritize the problems

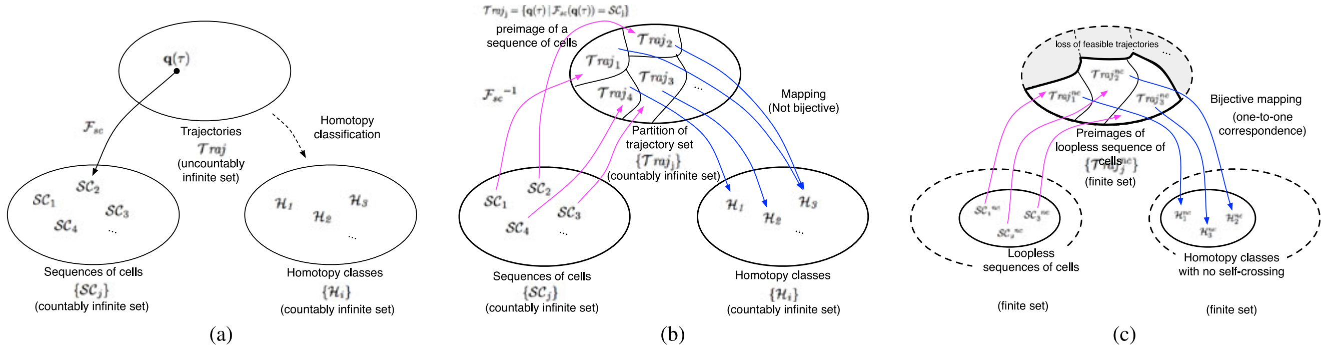
q(τ)
$\mathcal{F}_{sc}$
Trajectories
$\mathcal{T}raj$
(uncountably infinite set)
Homotopy classification
$\mathcal{SC}_1$ $\mathcal{SC}_2$ $\mathcal{SC}_3$ $\mathcal{SC}_4$ ...
Sequences of cells
$\{\mathcal{SC}_j\}$
(countably infinite set)
$\mathcal{H}_1$ $\mathcal{H}_2$ $\mathcal{H}_3$ ...
Homotopy classes
$\{\mathcal{H}_i\}$
(countably infinite set)
(a)
$\mathcal{T}raj_j = \{q(\tau) \mid \mathcal{F}_{sc}(q(\tau)) = \mathcal{SC}_j\}$
preimage of a sequence of cells
$\mathcal{F}_{sc}^{-1}$
$\mathcal{T}raj_1$ $\mathcal{T}raj_2$ $\mathcal{T}raj_3$ $\mathcal{T}raj_4$ ...
Mapping (Not bijective)
Partition of trajectory set
$\{\mathcal{T}raj_j\}$
(countably infinite set)
Sequences of cells
$\{\mathcal{SC}_j\}$
(countably infinite set)
Homotopy classes
$\{\mathcal{H}_i\}$
(countably infinite set)
(b)
loss of feasible trajectories
$\mathcal{T}raj_1^{nc}$ $\mathcal{T}raj_2^{nc}$ $\mathcal{T}raj_3^{nc}$
Bijective mapping
(one-to-one correspondence)
Preimages of loopless sequence of cells
$\{\mathcal{T}raj_j^{nc}\}$
(finite set)
$\mathcal{SC}_1^{nc}$ $\mathcal{SC}_2^{nc}$ $\mathcal{SC}_3^{nc}$
Loopless sequences of cells
(finite set)
$\mathcal{H}_1^{nc}$ $\mathcal{H}_2^{nc}$ $\mathcal{H}_3^{nc}$
Homotopy classes with no self-crossing
(finite set)
(c)

Fig. 3. Relations between trajectories, cell sequences, and homotopy class. (a) All feasible trajectories can be mapped to sequences of cells on adjacency graph. (b) Sequence of cells represents a partitioned set of feasible trajectories. All feasible trajectories within a partitioned set are homotopic. (c) Loopless sequences have a one-to-one correspondence with homotopy classes.

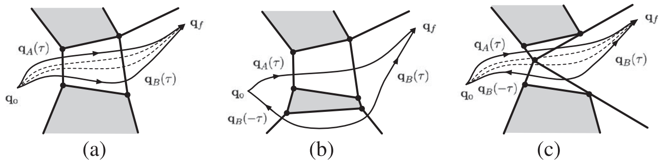

Fig. 4. (a) All feasible trajectories corresponding to the same sequence of cells are homotopic. (b) Two feasible trajectories corresponding to different loopless sequences of cells are not homotopic: one-to-one correspondence is formed between loopless sequences and homotopy classes. (c) Singularity of decomposition with nonminimum vertex set: two trajectories in the same homotopy class correspond with different loopless sequences.

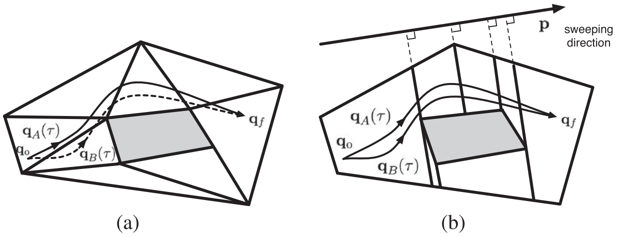

Fig. 5. Elimination of feasible trajectories and benefits of line-sweep decomposition. (a) $\mathbf{q}_A(\tau)$ and $\mathbf{q}_B(\tau)$ are homotopic, but $\mathbf{q}_B(\tau)$ is excluded from the search space of loopless sequences since it corresponds to a sequence with a loop. (b) Line-sweep decomposition is efficient in a sense that feasible trajectories that are monotonically increasing in sweeping direction are preserved in the search space.

to be solved. A generic method to achieving this is to assign heuristic costs to paths on the adjacency graph, then cost-rank potential paths and compare to a pre-defined cost threshold.

This approach is practically useful in the following senses. 1) It allows us to explicitly consider geometric properties (e.g., path width, length, and curvature) of fields of travel, as well as properties of specific trajectories. For instance, a heuristic measure of desirableness of fields of travel is also provided in previous work by Anderson *et al.* [9]. 2) It is possible to assign a heuristic likelihood of containing the global optimal trajectory to each cell sequence, in order to quickly choose a sequence containing the global optimal trajectory. However, such heuristics are often problem-specific depending on the given objective function and dynamics. We here propose a generic approach to utilize the existence of loops in the cell sequences to prioritize local problems to solve.

1) Strategy of Focusing on Loopless Cell Sequences: We propose to focus on loopless sequences on the graph, as a means to eliminate inefficiency arising from repeated visits of the same cells of candidate trajectories. In addition, loopless cell sequences have a desirable property of one-to-one correspondence with homotopy classes, as demonstrated in the preceding section. We can partition the entire set of all possible cell sequences $\{\mathcal{SC}_j\}$ into a set of sequences having repeated cells, $\{\mathcal{SC}_j{}^{\text{cycle}}\}$, and a set of sequences without repeated cells, $\{\mathcal{SC}_j{}^{\text{nc}}\}$; note that $\{\mathcal{SC}_j{}^{\text{nc}}\}$ is a finite set.

The limitation of this strategy is that some feasible trajectories are unavoidably eliminated when we restrict our scope to loopless sequences, as shown in Fig. 3(c). The trajectories that are not considered during optimization are those having cycles in corresponding sequences of cells, $\{\mathcal{T}\text{raj}_j^{\text{cycle}}\}$. Note that these trajectories have loops in cell sequences, but do not necessarily have loops in trajectories themselves, as shown in Fig. 5(a). Despite the small likelihood of occurrence, if an optimal trajectory in a homotopy class happens to correspond to a cell sequence with loops, it is not found by restricting ourselves to loopless cell sequences; an example class of problems where this may occur is nonholonomic robots. Clearly, it would be desirable to ensure that the optimal trajectory is not included in $\{\mathcal{T}\text{raj}_j^{\text{cycle}}\}$, although it is still possible to find the optimal trajectory by considering cell sequence with loops as well.

It is difficult to guarantee that the optimal solution corresponds to a loopless cell sequence before trajectory planning or optimization, for problems with general dynamics and objective functions. However, given knowledge about a potential optimal trajectory in a specific problem domain, we are able to utilize this knowledge in the decomposition procedure. Here, we suggest a specific type of cell decomposition based on its intuitive tendency to preserve optimal trajectories in loopless cell sequences. The motivation is an empirical observation that for cases where desirable trajectories exhibit a tendency to move monotonically in a certain direction (i.e., toward a goal region), it is useful to perform trapezoidal decomposition (rather than other decompositions such as triangulation) with parallel edges normal to this direction of movement.

Trapezoidal decomposition is one of the most popular types of exact cell decomposition [19]. In trapezoidal decomposition, an imaginary line is swept through the space in a principal direction

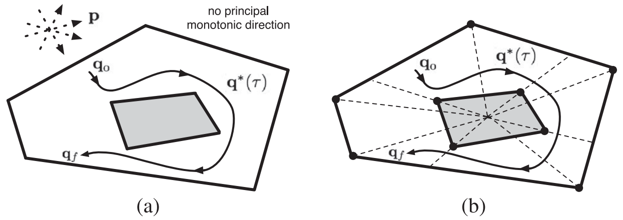

Fig. 6. (a) Example where the global optimal trajectory does not have any principal axes on which the projection is monotonic. (b) Alternate decomposition to preserve the global optimal trajectory in a loopless cell sequence.

and decomposes the space into cells whenever it meets critical points, as shown in Fig. 5(b). If we can find a sweeping axis onto which the projection of the global optimal trajectory is monotonic, we can ensure that the global optimal trajectory has no loops in the corresponding cell sequence, as restated below.

Corollary 9: The global optimal trajectory corresponds to a loopless cell sequence, i.e., $\mathcal{F}_{\mathrm{SC}}(\mathbf{q}^*(\tau)) \in \{\mathcal{SC}_j^{\mathrm{\,nc}}\}$, by trapezoidal decomposition with a sweeping axis $\mathbf{p}$ such that $\frac{d}{d\tau}(\mathbf{q}^*(\tau) \cdot \mathbf{p}) \geq 0$ (or ≤ 0) for $\forall \tau \in [t_0, t_f]$.

In practical applications, it is often intuitive to determine the sweeping direction. For instance, it is often possible to fit a straight line to the start and goal locations as shown in Fig. 5(b), since that may be a desirable direction to move. In some instances, such as vehicle navigation on roads, it is better to use the longitudinal direction of the road as the sweeping direction, since it is very unlikely that moving in reverse in the longitudinal direction is the optimal solution.

There may, however, be the cases where the global optimal trajectory does not have any principle axis $\mathbf{p}$ on which the projection is monotonic due to its dynamics or objective functions, as shown in Fig. 6(a). Fig. 6(b) illustrates an example of another relevant decomposition to keep the optimal trajectory in a loopless cell. However, choosing a relevant decomposition method before trajectory planning or optimization generally requires problem-specific heuristics, or assumptions about the nature of feasible or optimal trajectories.

2) Strategy Allowing Loops a Single Time for Each: In the cases where it is difficult to determine a judicious decomposition method *a priori*, ensuring that the global optimal trajectory corresponds to a loopless cell sequence, we can also take into account cell sequences with loops. The next priority is naturally assigned to local problems corresponding to cell sequences with unique loops only, i.e., those with multiple loops, but allowing only one of each loop. Note that the number of cell sequences with unique loops is also finite.

It is more likely that cell sequences with unique loops contain more desirable trajectories than cell sequences with multiple repeated loops. Trajectories corresponding to cell sequences with multiple identical loops either contain self-crossing in themselves, or cross the same edges multiple times back and forth, as shown in Fig. 7. As illustrated in the right example in Fig. 7(b), cell sequences with more than a single cycle for the loop consisting of more than two cells contain only trajectories that cross themselves, unless $\mathcal{C}_v$ and $\mathcal{C}_w$ are not the start cell or the goal cell. Thus, such trajectories are very unlikely to represent the global optimal solution, except for special cases of objective functions.

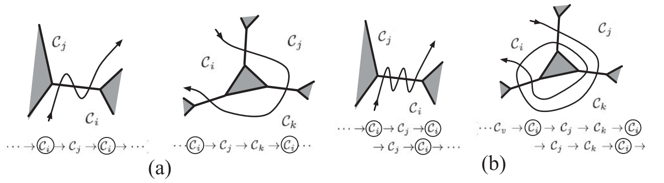
$\mathcal{C}_j$
$\mathcal{C}_i$
$\cdots \rightarrow \mathcal{C}_i \rightarrow \mathcal{C}_j \rightarrow \mathcal{C}_i \rightarrow \cdots$
$\mathcal{C}_i$
$\mathcal{C}_j$
$\mathcal{C}_k$
$\cdots \mathcal{C}_i \rightarrow \mathcal{C}_j \rightarrow \mathcal{C}_k \rightarrow \mathcal{C}_i \cdots$
(a)
$\mathcal{C}_j$
$\mathcal{C}_i$
$\cdots \rightarrow \mathcal{C}_i \rightarrow \mathcal{C}_j \rightarrow \mathcal{C}_i$
$\rightarrow \mathcal{C}_j \rightarrow \mathcal{C}_i \rightarrow \cdots$
$\mathcal{C}_i$
$\mathcal{C}_j$
$\mathcal{C}_k$
$\cdots \mathcal{C}_v \rightarrow \mathcal{C}_i \rightarrow \mathcal{C}_j \rightarrow \mathcal{C}_k \rightarrow \mathcal{C}_i$
$\rightarrow \mathcal{C}_j \rightarrow \mathcal{C}_k \rightarrow \mathcal{C}_i \rightarrow$
(b)

Fig. 7. (a) Trajectories with unique loops in cell sequence could be desirable due to certain dynamics or objective functions. (b) Trajectory with more than one cycle for each loop is less desirable than ones with unique loops. (a) Unique loops in cell sequence. (b) Double loops in cell sequence.

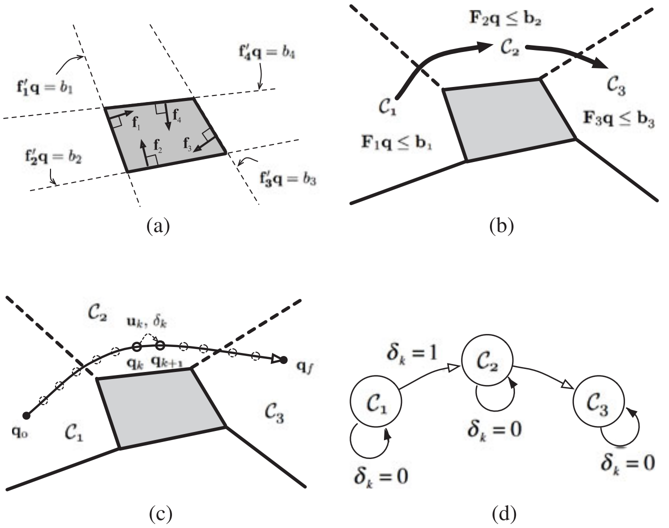

Fig. 8. MIP with homotopy constraints represented as a sequence of convex polygons. (a) Collision-free space surrounding of a convex polygonal obstacle can be represented as a union of half spaces[11]. (b) Homotopy constraints can be represented as a sequence of disjoint convex polygons. (c) MIP formulation of trajectory optimization within homotopy constraint with continuous variables for control inputs $\mathbf{u}_k$ and binary variables for transition indicators δ_k at each time step k. (d) Transition indicator δ_k: binary variables representing whether to remain in the current cell or move to the next cell in the sequence at each time step k.

V. Trajectory Optimization

A. Mixed-Integer Programming for Collision Avoidance

This section describes previous approaches that serve as background of the proposed approach described in Section V-B. Due to their nonconvex nature, problems involving collision-free constraints often lead to difficulties in the optimal trajectory generation problem. Richards [11] proposed a method to handle nonconvex collision-free constraints by formulating the optimization problem as an MIP problem containing binary variables. A key idea is to represent collision-free space with a convex polygonal obstacle as the union of half spaces, as seen in Fig. 8(a). This is possible because the convex polygonal obstacle can be represented as the intersection of half spaces, and the complement of a half space is still a half space. The union of half spaces can also be represented as the intersection of linear inequalities by introducing binary variables and a sufficiently large constant M (hence the term *Big-M method*), as follows:

$$\mathbf{f}'_{\mathrm{ijk}}\mathbf{q}_k \le b_{\mathrm{ijk}} + M(1 - z_{\mathrm{ijk}}), \quad \text{for } i \in \mathbb{Z}_{1,m_j}, j \in \mathbb{Z}_{1,n}, k \in \mathbb{Z}_{1,p} \tag{6}$$

$$\sum_{i=1}^{m_j} z_{\mathrm{ijk}} \ge 1, \quad \text{for } j \in \mathbb{Z}_{1,n}, k \in \mathbb{Z}_{1,p} \tag{7}$$

$$z_{\mathrm{ijk}} \in \{0, 1\}, \quad \text{for } i \in \mathbb{Z}_{1,m_j}, j \in \mathbb{Z}_{1,n}, k \in \mathbb{Z}_{1,p} \tag{8}$$

where m_j is the number of half spaces for obstacle j (i.e., the number of edges of convex polygonal obstacle j), n is the number of obstacles, and p is the number of sampling points over the horizon. For each time step k, the half space outside of the jth edge for obstacle i is represented as $\mathbf{f}'_{\mathrm{ijk}}\mathbf{q}_k \le b_{\mathrm{ijk}}$ and is released or imposed depending on the value of binary variable z_{ijk} in (6). Note that this half space can vary with time step k and thus can handle dynamic obstacles. When $z_{\mathrm{ijk}} = 0$, the half space $\mathbf{f}'_{\mathrm{ijk}}\mathbf{q}_k \le b_{\mathrm{ijk}}$ is released due to the last term, which is sufficiently large, $M \gg \mathbf{f}'_{\mathrm{ijk}}\mathbf{q}_k - b_{\mathrm{ijk}}$, and when $z_{\mathrm{ijk}} = 1$, it is imposed since the last term vanishes. The inequality constraint (7) ensures that at least one half space constraint is imposed for each obstacle j and time step k.

This representation is useful since optimization problems with nonconvex constraints can be formulated as MIP problems. However, since MIP is an NP-complete problem, efficient formulation of the problem is important for achieving reasonable computational performance. Such methods may include exploiting prior knowledge or properties of the problem structure. For example, Kim *et al.* [13] achieved improved computation time by removing redundancies in the set of possible combinations of binary variables. For example, any point $\mathbf{q}$ cannot be in $\mathbf{f}'_1\mathbf{q} \le b_1$ and $\mathbf{f}'_3\mathbf{q} \le b_3$ at the same time in Fig. 8(a), despite the fact that (7) allows this possibility.

One clear limitation of this problem formulation is that it applies to problems with convex polygonal obstacles and not to those with nonconvex polygonal obstacles. Formation of convex polygonal approximations of nonconvex obstacles, while a potential solution to this issue, could lead to highly conservative obstacle descriptions.

B. Mixed-Integer Programming With Cell Sequence Constraints

This section describes an efficient formulation of an MIP that exploits knowledge of given homotopy constraints, i.e., sequences of interior disjoint convex cells, that have been decomposed from collision-free constraints. Two main properties of homotopy constraints are used in this formulation. First, the robot can be located only in one decomposed disjoint cell at each time step so that only one convex constraint can and should be satisfied. This replaces inequality constraints in the problem formulation with equality constraints and significantly reduces the search space of the combination of binary variables. Second, we utilize the fact that the sequence of convex cells is provided *a priori*. The binary variables z's indicating satisfaction of each of the cells are not truly independent from each other, and thus, the replaced equality constraints still exhibit redundancies. By identifying more restrictive constraints in the combination of variable z's, we can improve efficiency of the MIP solution

$$\mathbf{F}_{ik}\mathbf{q}_k \le \mathbf{b_{ik}} + \mathbf{M}(1 - z_{ik}), \quad \text{for } i \in \mathbb{Z}_{1,N_D}, k \in \mathbb{Z}_{1,p} \tag{9}$$

$$\sum_{i=1}^{N_D} z_{ik} = 1, \quad \text{for } k \in \mathbb{Z}_{1,p} \tag{10}$$

$$\sum_{i=1}^{N_D} i \cdot z_{ik} - \sum_{j=0}^{k-1} \delta_j = 1, \quad \text{for } k \in \mathbb{Z}_{1,p} \tag{11}$$

$$\sum_{k=0}^{p-1} \delta_k = N_D - 1 \tag{12}$$

$$z_{ik} \in \{0, 1\}, \quad \text{for } i \in \mathbb{Z}_{1,N_D}, k \in \mathbb{Z}_{1,p} \tag{13}$$

$$\delta_k \in \{0, 1\}, \quad \text{for } k \in \mathbb{Z}_{0,p-1} \tag{14}$$

where N_D is the number of convex polygons in the cell sequence. The convex polygons are represented as a set of linear inequalities, $\mathbf{F}_{ik}\mathbf{q}_k \le \mathbf{b_{ik}}$, and their imposition is encoded through binary variables z_{ik} using vector $\mathbf{M}$ whose dimension is the same as $\mathbf{b_{ik}}$ and elements are large enough, $\mathbf{M} \gg \mathbf{F}_{ik}\mathbf{q}_k - \mathbf{b_{ik}}$ in (9). Since decomposed convex polygons are interior disjoint with each other, only one of them can be imposed at each time step k, yielding the equality constraints of (10). z_{ik} are not independent from each other since the convex polygon imposed at time step k does not change arbitrarily when the sequence between them is given.

The core independent decision variables are transition times between convex polygons. This decision freedom is expressed in constraint form through binary variables δ_k, indicating a decision of whether or not to move to the next convex polygon in the sequence or stay in the current convex polygon at the next time step $k+1$, as shown in Fig. 8(d). δ_k will uniquely determine z_{ik} through (11). For example, suppose the robot remains in the $\bar{i}$th convex polygon at time step k; then, $\bar{i} = 1 + \sum_{j=0}^{k-1} \delta_j$, i.e., one plus the number of transitions until time step k. Therefore, two inequality constraints, i.e., (10) and (11), yield a unique solution of $z_{\bar{i}k} = 1$ and $\forall_{i \ne \bar{i}} z_{ik} = 0$. Equation (12) represents a constraint on the robot's arrival at the final convex polygon where the goal configuration exists.

Combining the above with a cost function and a discretized model of the system dynamics, the trajectory optimization problem becomes an MIP problem with standard continuous optimization variable, control input $\mathbf{u}_k$ at each time step k, and the binary optimization variables involved in the homotopy constraints: independent binary variables δ_k indicating transition decisions, and dependent binary variables z_{ik} indicating imposition of corresponding convex polygon constraints. The following problem formulation applies to a typical case with linearized system dynamics and quadratic cost function, where the optimization reduces to a mixed-integer quadratic programming (MIQP) problem

$$\min_{\mathbf{u}_k} J = \sum_{k=1}^{p} \mathbf{x}_k{}' \mathbf{Q}_k \mathbf{x}_k + \sum_{k=0}^{p-1} \mathbf{u}_k{}' \mathbf{R}_k \mathbf{u}_k \qquad (15)$$

$$\text{subject to } \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k \qquad (16)$$

$$\mathbf{u}_{\min} \le \mathbf{u}_k \le \mathbf{u}_{\max} \qquad (17)$$

$$\text{homotopy constraints : } (9)\text{–}(14). \qquad (18)$$

C. Analysis of Problem Complexity

This section analyzes the computational benefits of the proposed divide-and-conquer approach. Since homotopy constraints for local problems do not have holes and can be represented as sequences of convex polygons, the resulting optimization problem can be solved in an efficient manner compared with the global problem with general collision-free constraints. Although problem solution time clearly depends on the performance of the particular MIP solver, the problem formulation has a great influence on computational performance. Specifically, an increased number of combinations of discrete variables leads to a large search space and high computational burden. We here analyze the reduction of the search space of binary variables in MIQP by comparing the number of possible combinations of binary variables satisfying constraints in the original global problem and a decomposed problem.

The possible number of combinations of binary variables are reduced by simply replacing inequality constraints (7) with equality constraints (10) using interior disjoint convex cells. In the case of inequality constraints of (7), the number of feasible combination is $\{\prod_{j=1}^{n}(2^{m_j}-1)\}^p$. If we assume that all obstacles have the same number of edges, $m_j = m$, for the purpose of simplicity of analysis, it is $(2^m-1)^{np}$. On the other hand, (10) has $(N_D)^p$ as the number of possible combinations of binary variables, where N_D is the number of convex cells. N_D cannot exceed the total number of edges $\sum_{j=1}^{n} m_j$, since a convex cell has at least one obstacle edge that is not shared with other cells. Similarly, for the case of $m_j = m$, the possible number of combinations is upper-bounded as $(N_D)^p \le (nm)^p$. Therefore, the number of possible combinations of binary variables becomes polynomial in the number of obstacles n, while it remains exponential in the number of horizons p.

In the case of the decomposed problem exploiting homotopy constraints, the variables z's are not independent from each other and are uniquely determined by δ_k through (11). The number of possible combinations of binary variables δ_k is "p choose $N_D - 1$", i.e., $\frac{p!}{(N_D-1)!(p-N_D+1)!}$ under the assumption of enough sampling points compared with the number of convex cells, i.e., $p > N_D$. This means the number of possible combinations does not monotonically increase as N_D increases. More specifically, the binomial coefficient is upper-bounded as $\frac{p!}{(N_D-1)!(p-N_D+1)!} \le \frac{p^{(N_D-1)}}{(N_D-1)!} \le e^p$. Therefore, the number of obstacles and edges of obstacles do not significantly affect the computation time as much as in the predecomposed problem. Note that this reduction of search space comes from the fact that a cell sequence is defined ahead of time.

Note that constraint (9) that is imposed and released by a single binary variable is a convex polygon, i.e., an intersection of half spaces, and thus, it is more restrictive than the single half space represented in (7). In terms of continuous optimization variables, imposing more restrictive constraints could yield slow optimization performance; however, since convexity is preserved, this influence on the total computation time is here not as significant as the effect of the reduced search space of binary variables. Therefore, the dominant cause of high computational burden is the extensive search space of binary variables. In Section VI-A, actual computation times for example problems are compared for undecomposed global problems and decomposed local problems.

We note that although N_D does not affect the upper bound of the number of possible binary variable combinations, the number of convex polygonal constraints in (9) increases as N_D increases. Therefore, it is better to reduce N_D as a means to improve computational efficiency. For example, the triangular decomposition does not have benefits compared with trapezoidal decomposition in terms of a reduced number of constraints involving continuous variables. However, there is a tradeoff since determining a decomposition that uses the smallest number of convex cells is NP-hard [20].

Finally, we note that the number of local problems is equal to the number of loopless paths on the adjacency graph, which is upper-bounded by the number of loopless homotopy classes, 2^n, where n is the number of obstacles. Although the growth of the number of local problems is exponential, since each local problem with a loopless sequence of cells uniquely corresponds to a homotopy class, a high-level evaluation step can allow filtering of undesired local problems. For example, it may be possible to heuristically evaluate the desirability (e.g., likelihood of containing a feasible solution) of a homotopy class based on the geometric properties of connected cells as in [9], as discussed in Section VI-B. When heuristic costs are assigned to the adjacency graph, it is possible to rank paths from the shortest to Kth shortest using Yen's loopless path ranking algorithm [21]. The time complexity of the algorithm is known to be $O(KN(E + N \log N))$, where E and N are the number of edges and nodes on the graph, respectively. It is also potentially possible to reflect a human supervisor's intention in the selection of homotopy classes; therefore, a single local problem that is consistent with the human's intention can be solved.

VI. Example Applications

In this section, simulation results for the proposed divide-and-conquer approach are shown for various applications. The computational benefits of the proposed approach are first demonstrated for a motion planning problem involving a simple point mass model. Next, a problem involving vehicle navigation on a roadway is presented to show benefits of correspondence of navigation decisions and cell sequences. Finally, the application to linear MPC with nonconvex collision-free constraints is demonstrated. All implementations (i.e., convex decomposition and graph search) except for optimization were coded in MATLAB. IBM ILOG CPLEX 12.5.1 was used for solving the

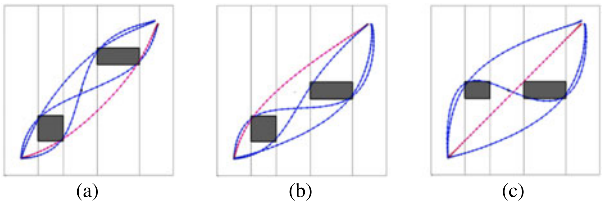

Fig. 9. Optimal trajectories for each homotopy class in scenarios with distinct obstacle positions and with $n = 2$, $p = 50$ (the global optimal trajectories in red and other local optimal trajectories in blue). (a) Scenario A. (b) Scenario B. (c) Scenario C.

MIQP. The simulations were carried out on a 2.4-GHz personal computer with 2 GB of RAM.

A. Point Mass Example

Here, the performance of the proposed approach is compared with a baseline approach for a simple model of a 2-D point mass. The dynamics can be represented with the linear model $\ddot{x} = u_{\text{ax}}$, $\ddot{y} = u_{\text{ay}}$, where u_{ax} and u_{ay} are accelerations in orthogonal directions. The time scale and the size of environment are normalized as $\mathbf{q}(\tau) : \tau \in [0, 1] \to (x, y) \in [0, 1]^2$, and start and goal configurations are given as $x(0) = 0.1,\ y(0) = 0.1$, and $x(1) = 0.9,\ y(1) = 0.9$. The initial and final velocities of the robot are chosen as $\dot{x}(0) = 0,\ \dot{y}(0) = 0$, and $\dot{x}(1) = 0,\ \dot{y}(1) = 0$. The trajectory is designed to minimize control effort, and the problem is thus formulated as a *minimum-fuel* problem with a quadratic cost function $J = \int_0^1 (u_{\text{ax}}^2 + u_{\text{ay}}^2) d\tau$. Note that the square root of the cost is the root-mean-squared acceleration, $a_{\text{rms}} = \sqrt{J}$.

Fig. 9 and Table II present simulation results for three scenarios involving two obstacles at distinct positions. The MIQP formulation of the undecomposed global problem and decomposed local problem are implemented in the same manner except for the handling of collision-free constraints, (6), (7), and homotopy constraints, (9)–(12), respectively. While the undecomposed global problem formulation generated the globally optimal trajectory in each environment, the proposed divide-and-conquer approach not only found the globally optimal trajectory, but also optimal trajectories contained within each homotopy class. As a result, explicit comparison of the minimum cost of each locally optimal trajectory is possible. For ease of recognition, the four homotopy classes in the scenarios are denoted by obstacle-avoidance directions. For example, homotopy cw-ccw avoids the lower-left obstacle in the clockwise (cw) direction and the upper-right obstacle in the counterclockwise (ccw) direction. The global optimality among the locally optimal solutions corresponding to each of homotopy constraints is explained in the Appendix.

The computation time of the undecomposed approach varied depending on the scenario. The computation time for scenario A for the undecomposed approach was observed to be an order of magnitude larger than the others due to the existence of a locally optimal solution with similar cost to the globally optimal solution. This can be verified by examining the costs of the two homotopies ccw-ccw and cw-cw, which are similar since the locations of the two obstacles are nearly symmetric about the line connecting the start and the goal. (The homotopy ccw-ccw contains the globally optimal solution.) In scenarios B and C, the globally optimal trajectories were found much faster than scenario A in the undecomposed approach. In summary, this example illustrates the fact that the required computation time (particularly for the globally optimal solution) is highly sensitive to the specific configurations of a particular scenario.

In contrast, it can be seen that proposed divide-and-conquer approach exhibits a fairly uniform computation time for this particular problem. In addition, since each local problem with homotopy constraints are independent, their computations are parallelizable. In scenario A, the summed computation time of each solution is much less than the computation time for the global problem formulation. However, in scenario C, the global problem formulation requires less computation time than some of local problems, since the global optimal solution in this case is a straight trajectory with minimum acceleration and deceleration toward the goal. The RMS acceleration of the second best homotopies, cw-cw and ccw-ccw, are approximately 1.43 times greater than that of the best homotopy, ccw-cw. However, the computation time for the global problem formulation remains greater than the computation time for the local problem formulation with corresponding homotopy constraints ccw-cw.

Fig. 10 shows box plots of the required computation time for both problem formulations from 1200 simulations with randomized location and sizes of obstacles, with whiskers indicating a 1.5 IQR. The number of obstacles, n, was chosen randomly between 1 and 7. The locations and sizes of rectangular obstacles were randomized, with minimum length of both edges of 0.1. The number of sampling points, p, was fixed at 16. While the computation time to solve the undecomposed global problem tends to increase as the number of obstacles increases, the computation time for the decomposed local problem remains bounded as the number of obstacles increases to 7. Note that the computation times are shown in log scales. This is consistent with the analysis in Section V-C and results from the exponentially increasing number of possible combinations of discrete variables with n in the global problem formulation, compared with the upper bound independent of n in the decomposed local problem formulation. In addition, the decomposed local problem formulation exhibits a fairly uniformly distributed computation time, while the global problem formulation has a widely distributed computation time, as seen in the instance of Fig. 9 and Table II. Fig. 11 shows results for varying number of sampling points, p, when the number of obstacles is fixed at $n = 3$. The computation times for the decomposed problems grow slowly than those for the undecomposed problems although they both increases exponentially with the number of sampling points as discussed in Section V-C.

We highlight the fact that there is no guarantee of collision-free motion between sampling points, although each sampling point is guaranteed to be collision free. Both formulations exhibit this limitation, which is inherent for optimization approaches with discretized time steps. From a practical perspective, this issue can be mitigated by the increased number of sampling points or via obstacle dilation by a desired tolerance. A more rigorous approach, developed in [22], is to specify

TABLE II
COMPARISON OF THE RESULTS OF THE SCENARIOS IN FIG. 9

			divide-and-conquer approach				
				homotopy			
	scenarios	undecomposed approach [11]	cell decomp.	cw-cw	cw-ccw	ccw-cw	ccw-ccw
A	a^*_{rms} $(\times 10^{-3})$	13.2	·	13.4	16.8	18.9	13.2
	comp. time (s)	598.47	0.82	21.39	28.69	36.28	16.03
B	a^*_{rms} $(\times 10^{-3})$	12.8	·	12.8	24.7	13.9	16.1
	comp. time (s)	44.43	1.12	18.84	36.58	15.90	28.12
C	a^*_{rms} $(\times 10^{-3})$	11.7	·	16.1	36.8	11.7	16.1
	comp. time (s)	20.48	0.86	16.40	43.17	15.26	24.37

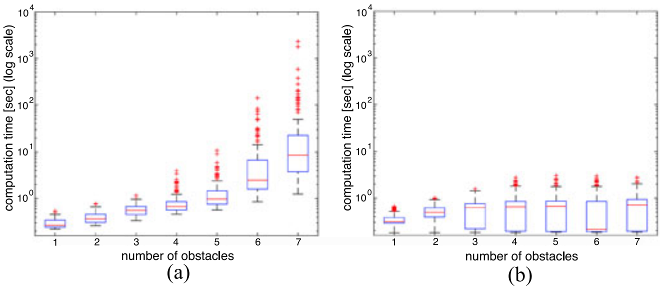

Fig. 10. Comparison with the previous approach [11] of computation time distribution of 1200 simulations with randomized locations and sizes of obstacles and varying number of obstacles n and the fixed number of sampling points, $p = 16$. (a) Undecomposed approach. (b) Decomposed approach.

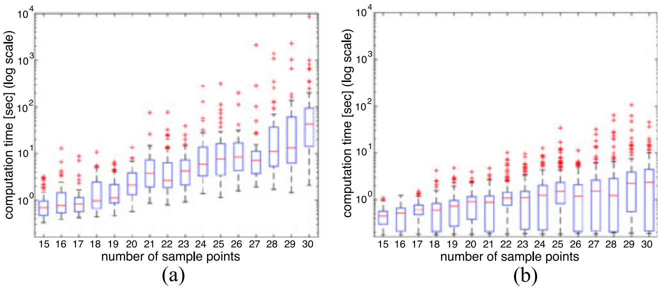

Fig. 11. Comparison with the previous approach [11] of computation time distribution of 1000 simulations with randomized locations and sizes of obstacles and varying number of sampling points p and the fixed number of obstacles, $n = 3$. (a) Undecomposed approach. (b) Decomposed approach.

sequences of cells with a denser decomposition in a way to ensure that any line connecting two points in adjacent cells does not cross obstacles.

B. Vehicle Navigation and Decision Making Support on Roadways

As a second example, we here analyze a scenario involving vehicle navigation on roadways, where trajectories with minimum effort ensuring satisfaction of driving requirements are generated for various navigation decisions. This could be utilized as a decision making support system by quantifying the desirability of each navigation decisions in terms of the minimum required acceleration levels. In this simulation, a standard bicycle model with a no-slip assumption was employed, with the following form:

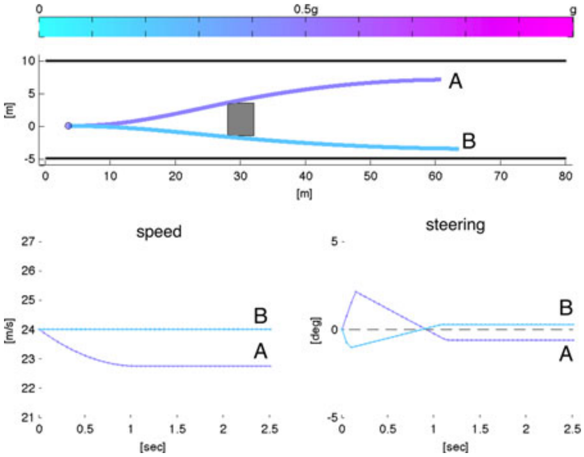

Fig. 12. Optimal trajectories for two different decisions in vehicle navigation on roads: $a^*_{\mathrm{rms}} = \sqrt{J^*/T}$ are 4.5(m/s^2) and 2.1(m/s^2) for A and B, respectively.

$$\dot{x} = v\cos\theta,\ \ \dot{y} = v\sin\theta,\ \ \dot{\theta} = \frac{v}{L}\tan u_s,\ \ \dot{v} = u_a \tag{19}$$

where L is the wheel base, u_a is the longitudinal acceleration, and u_s is the steering angle of the front wheel. In the simulation, $L = 2$ (m) was assumed. For simplicity, the vehicle model was linearized about a nominal speed v_0 under the assumption of a small heading angle θ and steering angle u_s:

$$\dot{x} = v,\ \ \dot{y} = v_0\theta,\ \ \dot{\theta} = \frac{v_0}{L}u_s,\ \ \dot{v} = u_a. \tag{20}$$

The objective was to minimize the root-mean-squared acceleration; therefore, the quadratic cost function in terms of the control inputs u_a and u_s was constructed as follows:

$$J = \int_0^T \left(\dot{v}^2 + (v\dot{\theta})^2\right)dt \approx \int_0^T \left(u_a^2 + \left(\frac{{v_0}^2}{L}\right)^2 u_s^2\right)dt \tag{21}$$

where T is the prediction horizon. Then, $a_{\mathrm{rms}} = \sqrt{J/T}$ is the root-mean-squared acceleration of the trajectory so that no tuning parameters are introduced in the comparison of optimal trajectories between various navigation decisions, and the cost retains a physical meaning.

Fig. 12 shows simulation result of the simple case where two navigation decisions exist. In on-road navigation, the vehicle is

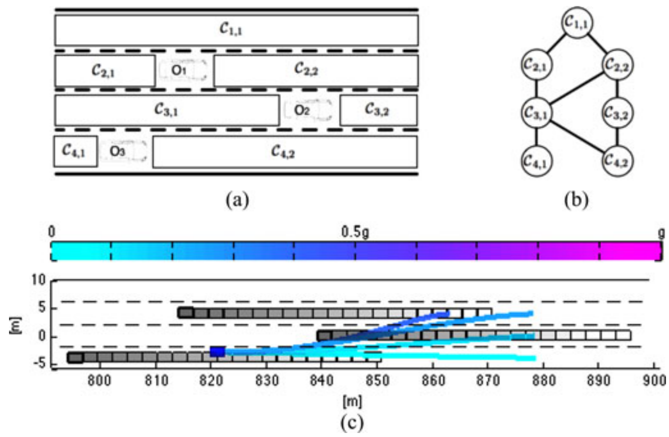

Fig. 13. Application to highway vehicle navigation. (a) Lane decomposition. (b) Adjacency graph. (c) Optimal trajectories for different navigation decisions.

TABLE III
COSTS OF THE OPTIMAL TRAJECTORIES AND COMPUTATION TIME OF THE SIMULATION OF FIG. 13

feasible sequences of cells	a^*_{rms} (m/sec^2)	comp. time (sec)
$\mathcal{C}_{4,2}$	0.04	0.146
$\mathcal{C}_{4,2} \rightarrow \mathcal{C}_{3,1}$	0.94	0.248
$\mathcal{C}_{4,2} \rightarrow \mathcal{C}_{3,1} \rightarrow \mathcal{C}_{2,1}$	3.78	0.216
$\mathcal{C}_{4,2} \rightarrow \mathcal{C}_{3,1} \rightarrow \mathcal{C}_{2,2}$	2.40	0.269

expected to remain within the road boundaries, and a goal point is often not precisely specified. However, it is generally desirable for the vehicle to align with the principal direction of the road; therefore, the final heading angle can be regulated to be aligned with the road, i.e., $\theta(T) = 0$ in this simulation. The feasible input range was set as $-4 \leq u_a \leq 4$ (m/s^2), $-10 \leq u_s \leq 10$ (degree), and $-15 \leq \dot{u}_s \leq 15$ (°/s). The prediction time horizon T was 2.5 (s) and the number of sampling points p was 50 so that the time interval between sampling points is 0.05 (s).

A and B in Fig. 12 are the optimal trajectories associated with two different navigation decisions. Since a goal point is not specified, it is impossible to determine whether or not the two trajectories are homotopic. However, the two trajectories correspond with different sequences of cells so that the navigation decisions are distinguished in this case. In trajectory A, moving in a clockwise manner about the obstacle, the vehicle was required to both steer and reduce speed to avoid the obstacle, whereas in trajectory B, moving in a counterclockwise manner about the obstacle does not require the vehicle to modify its speed. The required minimum RMS accelerations for collision avoidance are explicitly compared, and it can be observed that avoiding the obstacle in a counterclockwise manner results in lower cost trajectories.

Fig. 13 and Table III show application of the proposed approach to a highway navigation problem. The key difference of this application from the previous one is the existence of lane structure. The divided lanes on the road can be naturally utilized in the cell decomposition process. The basic idea is to decompose each lane according to the locations of the obstacles in Fig. 13(a). Each cell is denoted by $\mathcal{C}_{i,j}$, where i is the index of the lane, and j is the index of decomposed cells in the ith lane. The adjacency graph between decomposed cells is constructed as in Fig. 13(b), where the cells $\mathcal{C}_{i,j}$ and $\mathcal{C}_{i+1,j'}$, one in the ith lane and one in the adjacent $(i+1)$th lane, are connected when they share a common edge. Note that any cells in the same lane are not connected on the graph since they are separated by obstacles if decomposed properly.

When obstacles are moving within a lane, the size and location of the decomposed cells will change as a function of the motion properties of the obstacles. However, the indices of decomposed cells will remain the same, i.e., the nodes of the graph would be unchanged. The edges connecting the nodes on the graph can be created or removed at each time step. However, a single time-invariant adjacency graph can be constructed by taking a conservative approach, as follows: If two nodes are connected at least once over the prediction horizon, an edge is created in the single graph.

Although a unique goal point is not typically specified in the highway navigation problem, goal regions, i.e., ranges of lateral position, can generally be specified. In terms of the adjacency graph, each decomposed lane component $\mathcal{C}_{i,j}$ can be chosen as a goal node. Since the start node is uniquely defined based on the current location of the host vehicle, it is possible to enumerate possible paths on the graph for different goal nodes. These paths correspond to different navigation decisions, even though their relationship to homotopy classes is not well defined. For example, Fig. 13(c) shows identified feasible trajectories for different navigation decisions; in this example, only four navigation decisions contained feasible trajectories.

The likelihood of navigation decisions to contain feasible trajectories, or the desirability of various navigation decisions, can be evaluated heuristically based on the geometric properties corresponding sequence of cells. For example, if a unit cost is assigned to each edge of the graph, the cost of an identified path represents the number of lane change, and thus, it is possible to restrict the maximum number of lane changes since a high number of lane changes is generally not desirable from a practical perspective.

In the simulation of Fig. 13, it is assumed that other vehicles are moving at a constant speed and maintaining their current lane position. The speeds of obstacles were 18, 14, and 18 m/s for O_1, O_2, and O_3, respectively. The ranges of feasible input were $-4 \leq u_a \leq 4$ (m/s^2), $-30 \leq u_s \leq 30$ (degree), and $-60 \leq \dot{u}_s \leq 60$ (°/s) and the prediction time horizon was 3 s and the number of samples p was 20 so that the sampling time Δt was 0.15 s. To simulate an online application, receding-horizon planning was performed where the prediction horizon was shifted at the next time step to generate a new optimal trajectory based on new measurements to enable the system to quickly respond to a dynamically changing environment.[2]

C. Model-Predictive Control With Nonconvex Collision-Free Constraints

This section shows an example of MIQP formulation used in linear MPC framework. MPC is a receding horizon control

[2] A video of the simulation is available at https://vimeo.com/96673683.

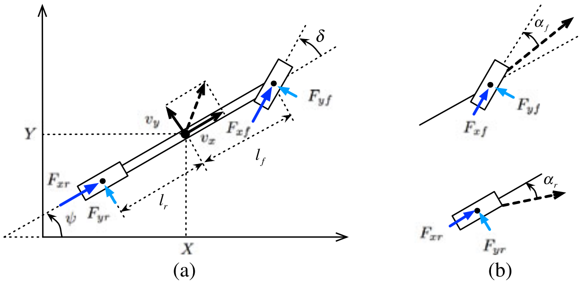

Fig. 14. Vehicle model for control. (a) Slip bicycle model. (b) Front and rear side slip angles.

approach where the optimal input sequence is solved online with a designed cost function and required constraints. Linear MPC is frequently used due to its computational benefits, since optimization problems with quadratic cost function and convex polygonal constraints reduce to QP problems, a convex optimization.

Due to the nonconvexity of collision-free constraints, most linear MPC-based vehicle navigation frameworks are decomposed into two stages. In a high-level stage, a desirable reference trajectory is generated with motion planning algorithms that often employ a simple vehicle model. In the low-level stage, deviation from this reference trajectory is penalized via a cost function without imposing nonconvex constraints. In this decoupled approach, it is common to use a simple vehicle model in the motion planning stage to reduce computational burdens for handling nonconvex collision-free constraints. However, simply penalizing deviation from the reference trajectory computed based on the simple model does not guarantee collision avoidance for the controlled vehicle.

In contrast, we here show an efficient way of incorporating nonconvex constraints into the optimization problem, specifically an MIQP formulation for running linear MPC with nonconvex collision-free constraints.

Fig. 14 shows the states of bicycle model with slip, and definition of side slip angles. The dynamics of the vehicle model is computed as follows, where the two inputs are the longitudinal tire force for braking F_x and steering δ. It is assumed that the braking force is distributed at a fixed ratio b, i.e., $F_{xf} = bF_x$, $F_{xr} = (1-b)F_x$:

$$\dot{X} = v_x \cos\psi - v_y \sin\psi \quad (22)$$

$$\dot{Y} = v_x \sin\psi + v_y \cos\psi \quad (23)$$

$$\dot{\psi} = \omega \quad (24)$$

$$\dot{v}_x = \frac{1}{m}(F_{xf}\cos\delta - F_{yf}\sin\delta + F_{xr}) + v_y\omega \quad (25)$$

$$\dot{v}_y = \frac{1}{m}(F_{xf}\sin\delta + F_{yf}\cos\delta + F_{yr}) - v_x\omega \quad (26)$$

$$\dot{\omega} = \frac{1}{I_{zz}}\{l_f(F_{xf}\sin\delta + F_{yf}\cos\delta) - l_r F_{yr})\}. \quad (27)$$

Normal tire forces are assumed to be constant, and lateral tire forces are assumed to be linear with respect to the side slip angles. The side slip angles are approximated as follows:

$$F_{yf}(\alpha_f) = \mu F_{zf} C_f \alpha_f \quad (28)$$

$$F_{yr}(\alpha_r) = \mu F_{zr} C_r \alpha_r \quad (29)$$

$$\alpha_f(\delta, v_x, v_y, \omega) \approx \delta - \frac{v_y + l_f\omega}{v_x} \quad (30)$$

$$\alpha_r(v_x, v_y, \omega) \approx -\frac{v_y - l_r\omega}{v_x} \quad (31)$$

$$F_{zf} = mg\frac{l_r}{l_f + l_r} \quad (32)$$

$$F_{zr} = mg\frac{l_f}{l_f + l_r}. \quad (33)$$

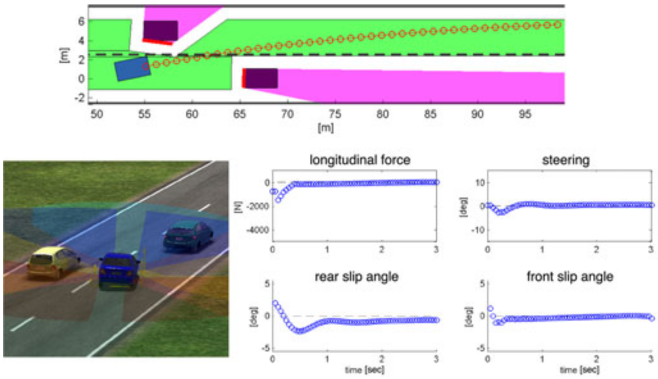

Fig. 15. Single time step results of optimization with nonconvex safe region constraint detected by a visibility sensor model with the snap shot of CarSim visualization.

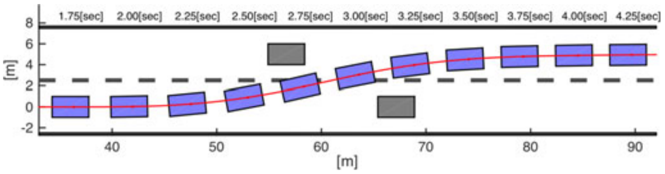

Fig. 16. Resulting trajectory of the vehicle controlled in MPC framework with nonconvex collision-free constraints.

The nonlinear vehicle dynamics is discretized and linearized about current states and previous inputs. The resulting dynamics is represented as a linear system in the following form [23]:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{g}. \quad (34)$$

Figs. 15 and 16 illustrate a snapshot and a resulting trajectory of a vehicle controlled in CarSim based on the proposed MPC framework for an obstacle-avoidance maneuver. The vehicle is assumed to have low-resolution visibility sensors with limited range of 30 m and tries to change the lane without collisions as soon as it detects the obstacle ahead. The initial speed of the vehicle was 80 km/h. A standard passenger vehicle (i.e., sedan) model was employed in CarSim.[3] Collision-free

[3] $m = 1650$ kg, $I_{zz} = 3234$ kg·m^2, $l_f = 1.4$ m, $l_r = 2.1$ m, $C_f = 10$, $C_r = 10$, $\mu = 1$, $b = 0.7$.

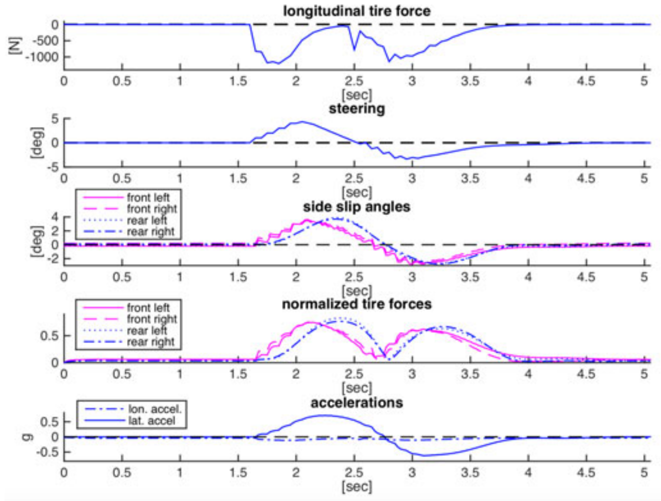

Fig. 17. MPC inputs (i.e., longitudinal tire force and steering), resulting side slip angles, resulting friction utilization (i.e., normalized magnitude of each horizontal tire force with respect to each normal tire force), and resulting accelerations in units of gravity acceleration g.

constraints were constructed as nonconvex polygons dilated by the size of the vehicle from the safe region detected by sensor models. Nonconvex polygons were decomposed into multiple convex polygonal cells by trapezoidal decomposition, and the cell sequence constraint from the starting cell to the goal cell was imposed in the form of (9)–(14).

The feasible input constraints $-\mu mg \leq F_x \leq 0$ (N), $-20 \leq \delta \leq 20$ (degree), and $-20 \leq \dot{\delta} \leq 20$ (°/s) were expressed and imposed in the polyhedral forms. Terminal constraints for alignment with the lane center were also imposed, i.e., yaw angle ψ and position Y at the final step of the horizon. The cost function was designed to minimize the magnitude of the front tire force and the deviation from the lane center. The sampling time for prediction was 0.05 s, and the number of prediction horizon and control horizon were 60 and 30, respectively. The control steps over the entire prediction steps were equally distributed, i.e., a control move per every two steps was optimized. The MPC control loop was run at 20 Hz.

Fig. 17 shows the executed input from the model-predictive controller and the measured resulting states of the controlled vehicle in CarSim.[4] For obstacle avoidance and to ensure that the vehicle remained within the road, both steering and brake inputs were imposed minimally with respect to the resulting front tire friction forces based on the slip bicycle model. Asymmetry between the left and right side of the vehicle in slip angles and tire forces was not significant to cause instability or collisions in this minimal control example. The vehicle was successfully controlled with the friction utilization up to 0.82 and lateral acceleration up to 0.7 g.

VII. Conclusion

Here, we have developed a divide-and-conquer approach to the kinodynamic motion planning problem in a 2-D polygonal space. The global problem is decomposed (or divided) into simpler local problems by convex decomposition with the minimal vertex set. An efficient formulation of MIP has been presented as a method to exploit desirable properties of local problems. This method has been compared with previous MIP formulation, and the reduction in complexity that can be achieved under the proposed method has been both analyzed and demonstrated through a number of simulations.

Each decomposed local problem can be independently solved (or conquered) and explicitly parallelized in their computation. However, it is still desirable to filter out lists of the local problems or prioritize the order of the local problems to be solved, since the cell sequences can be, in principle, enumerated infinitely by allowing loops. In this selection and prioritization process, it is possible to reflect high-level evaluation of desirableness of cell sequences corresponding to distinct navigation decisions. The paper showed one-to-one correspondence between loopless cell sequence and homotopy class. It also investigated the inefficiency of trajectories corresponding to cell sequences with more than a single cycle for the loops. In terms of decomposition methods, trapezoidal decomposition was efficient to retain the desirable trajectories, including the global optimal trajectory, in a loopless cell sequence.

The explicit enumeration of distinct navigation decisions has been successfully shown in the context of decision making support for vehicle navigation. In addition, the one-to-one correspondence between loopless cell sequence and homotopy class allowed utilization of the proposed approach in homotopic optimal trajectory planning. Finally, the paper has shown that the proposed MIQP formulation fits into a linear MPC framework with nonconvex collision-free constraints, which can serve as a feedback control law.

Appendix
Sketch of the Proof of the Global Optimality of the Example of Point Mass Minimum-Fuel Problem

The optimal solutions corresponding to all loopless cell sequences is monotonic in the sweeping axis (x-axis) of trapezoidal decomposition in Fig. 9. Besides, any possible cell sequences with loops contain trajectories that are all not monotonic in x-axis. Since the objective in the problem is to minimize the fuel, all trajectories that are not monotonic in x-axis have higher cost than the trajectories that are monotonic in x-axis. Therefore, the best solution among the solutions corresponding to loopless sequences is the global optimal solution.

References

[1] S. M. LaValle, “Motion planning,” *IEEE Robot. Autom. Mag.*, vol. 18, no. 1, pp. 79–89, Jun. 2011.

[2] S. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[3] E. Hernàndez Bes *et al.*, “Path planning with homotopic constraints for autonomous underwater vehicles,” Ph.D. dissertation, Universitat de Girona, 2012.

[4] S. Bhattacharya, V. Kumar, and M. Likhachev, “Search-based path planning with homotopy class constraints,” presented at the 3rd Annu. Symp. Combinatorial Search, Atlanta, GA, USA, 2010.

[5] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, “An optimal-control-based framework for trajectory planning, threat assessment, and

[4] A video of the simulation is available at https://vimeo.com/126836637.

semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *Int. J. Veh. Auton. Syst.*, vol. 8, no. 2, pp. 190–216, 2010.

[6] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Trans. Syst., Man and Cybern.*, vol. 22, no. 2, pp. 224–241, Mar./Apr. 1992.

[7] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning a*," *Artif. Intell.*, vol. 155, no. 1, pp. 93–146, 2004.

[8] L. Kavraki and J.-C. Latombe, "Randomized preprocessing of configuration for fast path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 2138–2145.

[9] S. J. Anderson, S. B. Karumanchi, K. Iagnemma, and J. M. Walker, "The intelligent copilot: A constraint-based approach to shared-adaptive control of ground vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 2, pp. 45–54, Summer 2013.

[10] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, vol. 18. New York, NY, USA: Wiley, 1988.

[11] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *J. Guidance, Control, Dyn.*, vol. 25, no. 4, pp. 755–764, 2002.

[12] A. Richards, "Constraining the sense of conflict resolution: Supervision of route optimization," presented at the SESAR Innovation Days, Toulouse, France, 2011.

[13] S. Kim, K. Sreenath, S. Bhattacharya, and V. Kumar, "Optimal trajectory generation under homology class constraints," in *Proc. IEEE 51st Annu. Conf. Decision Control*, Dec. 2012, pp. 3157–3164.

[14] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Auton. Robots*, vol. 33, no. 3, pp. 273–290, 2012.

[15] B. Chazelle, "A theorem on polygon cutting with applications," in *Proc. 23rd Annu. Symp. Foundations Comput. Sci.*, 1982, pp. 339–349.

[16] J. M. Lien and N. M. Amato, "Approximate convex decomposition of polyhedra," in *Proc. ACM Symp. Solid Phys. Modeling*, 2005, pp. 121–131.

[17] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. New York, NY, USA: Springer, 1985.

[18] L. P. Chew, "Constrained delaunay triangulations," *Algorithmica*, vol. 4, nos. 1/4, pp. 97–108, 1989.

[19] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA, USA: MIT Press, 2005.

[20] A. Lingas, "The power of non-rectilinear holes," in *Proc. 9th Colloq. Automata, Lang. Program.*, 1982, pp. 369–383.

[21] J. Y. Yen, "Finding the *k* shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, 1971.

[22] M. H. Maia and R. K. Galvao, "On the use of mixed-integer linear programming for predictive control with avoidance constraints," *Int. J. Robust Nonlinear Control*, vol. 19, no. 7, pp. 822–828, 2009.

[23] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. Tsengz, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *Proc. 46th IEEE Conf. Decision Control*, 2007, pp. 2980–2985.

Junghee Park received the B.S. and M.S. degrees in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 2007 and 2009, respectively. He is currently working toward the Doctoral degree in mechanical engineering with Massachusetts Institute of Technology, Cambridge, MA, USA.

He had served as an Officer/Lecturer with the Department of Electrical Engineering and Information Science, Korea Military Academy, Seoul, from 2009 to 2012. His research interests include robot motion planning, optimal control, autonomous/semiautonomous vehicle navigation, and multiagent system.

Sisir Karumanchi received the Bachelor's degree in mechatronic engineering from University of Sydney, Sydney, Australia, in 2005 and the Ph.D. degree from the Australian Centre for Field Robotics, University of Sydney, in 2010.

He is currently a Technologist with the Manipulation and Sampling Group under the Mobility and Robotic Systems Section, NASA Jet Propulsion Laboratory, Pasadena, CA, USA. Between 2011 and 2014, he was a Postdoctoral Researcher with Massachusetts Institute of Technology under Dr. K. Iagnemma in the Robotic Mobility Group. His research interests include guaranteed motion safety, mobile manipulation, semiautonomous control, vehicle–terrain interaction, and Bayesian nonparametric inference.

Karl Iagnemma received the B.S. degree from University of Michigan, Ann Arbor, MI, USA, and the M.S. and Ph.D. degrees from Massachusetts Institute of Technology (MIT), Cambridge, MA, USA.

He is currently a Principal Research Scientist with MIT and the Director of the Robotic Mobility Group, where he was a National Science Foundation Graduate Fellow. He is the author of the monograph *Mobile Robots in Rough Terrain: Estimation, Planning and Control with Application to Planetary Rovers* (New York, NY, USA: Springer, 2004). He has recently led research programs for organizations including the U.S. Army Tank-Automotive and Armaments Command, the Army Research Office, DARPA, the NASA Mars Program Office, Ford Motor Company, Nissan, and Samsung, among others. He has authored or coauthored nearly 100 conference and journal papers on a wide range of robotic topics. His primary research interests include the design, sensing, motion planning, and control of mobile robots in challenging environments.

Dr. Iagnemma has served as an Associate Editor of IEEE TRANSACTIONS ON ROBOTICS and *Journal of Field Robotics* and has been the Co-editor of books on the DARPA Grand Challenge and Urban Challenge Autonomous Vehicle competitions.