Introduction

- Despite recent successes such as the NASA Mars Rovers and the DARPA Grand Challenges, many barriers remain for the deployment of autonomous vehicles in both military and civilian applications. In particular, the automation of the task of vehicle motion planning remains challenging as the size, complexity and dimension of the environment to be planned in increases.
- We developed a novel multi-stage path planning algorithm (Tunnel-MILP) that seeks near optimal solutions which satisfy linear vehicle dynamic constraints.
- The Tunnel-MILP algorithm is a three stage path planning method that relies on the identification of a sequence of convex polygons to form an obstacle free tunnel through which to plan a dynamically feasible path.
- We also developed a greedy cut method for improved decomposition of the environment, resulting in fewer regions than existing algorithms.

Problem Formulation

Consider a vehicle navigating through an environment bounded by a convex polygon, \( \mathcal{E}_i \subseteq \mathbb{R}^2 \) \( i \in [0, N] \) be the number of obstacles, and each obstacle \( \mathcal{O}_i \subseteq \mathbb{R}^2 \). Finally, a planning space, \( P \), for the vehicle can be defined as a polygon with holes.

The dynamics of the vehicle are:

\[
\begin{bmatrix}
\dot{\mathbf{x}}(t+1) \\
\dot{\mathbf{v}}(t+1)
\end{bmatrix} =
\begin{bmatrix}
A & B(u(t))
\end{bmatrix}
\begin{bmatrix}
\mathbf{x}(t) \\
\mathbf{v}(t)
\end{bmatrix}
\]

where \( \mathbf{x}(t) \in \mathbb{R}^d \) is the position, \( \mathbf{v}(t) \in \mathbb{R}^d \) the velocity, and \( u(t) \in \mathbb{R}^m \) is the control input. The constraints on the dynamics are:

\[
\begin{align*}
x(t) & \in P, & \forall t \in [1, T] \\
v(t) & \leq u(t), & \forall t \in [1, T] \\
u(t) & \leq u_{\text{max}}, & \forall t \in [1, T] \\
v(t) & \leq v_{\text{max}}, & \forall t \in [1, T] \\
x(T) & = x_{\text{goal}}
\end{align*}
\]

where \( T \) is the maximum number of time-steps allowed for the vehicle to reach the goal.

Mixed Integer Linear Programming (MILP)

- MILP formulations are powerful because they can capture the non-convexity of the problem and encode it using integers.
- MILP is NP-hard in the number of integer variables
- Fortunately, there exist powerful optimization tools that can solve MILP formulations quickly.

The standard MILP formulation is:

\[
\begin{align*}
\text{minimize} & \quad \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{z}
\end{align*}
\]

subject to

\[
\begin{align*}
A_1 \mathbf{x} + A_2 \mathbf{z} & \leq \mathbf{b}
\end{align*}
\]

where \( \mathbf{z} \) must be in the set of integers.

MILP for Control

\[
\begin{align*}
\text{minimize} & \quad \gamma_1 f + (1 - \gamma_1) ||\mathbf{u}||_1 \\
\text{subject to} & \quad \begin{align*}
\mathbf{x}(t+1) = A \mathbf{x}(t) + B(u(t)) \\
\mathbf{u}(t) & \leq \mathbf{u}(t) \leq \mathbf{u}_{\text{max}}, \quad \forall t \in [1, T] \\
\mathbf{v}(t) & \leq \mathbf{v}(t) \leq \mathbf{v}_{\text{max}}, \quad \forall t \in [1, T] \\
\mathbf{x}(T) & = \gamma \mathbf{y}_{\text{goal}}
\end{align*}
\]

where \( \gamma_1 \) is the time that the vehicle reaches the goal and \( \gamma \) is the time that the vehicle reaches the goal.

Tunnel-MILP Algorithm

The Tunnel-MILP algorithm is broken down into three steps:

1. Determine a pre-path through the environment ignoring the dynamics of the vehicle;
2. Decompose the environment into convex polytopes around the pre-path;
3. Solve the dynamically feasible, optimal control problem through the sequence of convex polytopes.

Intuition

Since MILP formulations are NP-hard in the number of binary variables used, it is important to use as few as possible. The Tunnel-MILP method relies on the following insight: requiring the state to remain inside a convex polytope requires no binary variables, whereas requiring the state to remain outside a convex polytope requires one for each edge.

Decomposition Methods

The second step of Tunnel-MILP is to decompose the environment into convex polytopes and determine the sequence of polytopes that entirely contains the pre-path. There are various algorithms which accomplish this decomposition, two examples are:

- Delaunay Triangulation
- Trapezoidal Decomposition

Greedy Cut Algorithm

The Greedy Cut algorithm is defined in three steps:

1. The region of interest is restricted to contain only the area surrounding the pre-path.
2. The set of non-convex vertices is identified and cuts are selected to eliminate them while avoiding those that intersect the pre-path.
3. Finally, the polygons that contain the pre-path are combined to form tunnel in the MILP problem formulation.

Comparison

As shown in the table below, there is only a slight increase in the number of time-steps to reach the goal for the Tunnel-MILP algorithm above the optimal final time, and the input cost difference is also negligible between the two algorithms.

Both of these factors suggest that the significant decrease in the computation time for the Tunnel-MILP algorithm offsets the slight increase in the cost function for the Tunnel-MILP algorithm over the standard MILP method.