$T^\star$-Lite: A Fast Time-Risk Optimal Motion Planning Algorithm for Multi-Speed Autonomous Vehicles

James P. Wilson[†], Zongyuan Shen[†], Shalabh Gupta[†], Thomas A. Wettergren[‡]

[†]*Laboratory of Intelligent Networks and Knowledge-Perception Systems*
Department of Electrical & Computer Engineering, University of Connecticut, Storrs, CT, USA
[‡]*Naval Undersea Warfare Center, Newport, RI 02841, USA*

IEEE/MTS Global Oceans Conference 2020, Singapore-US Gulf Coast

This work was supported by US Office of Naval Research under Award Number N000141613032. Any opinions or findings herein are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

Autonomous Vehicle Path Planning Applications

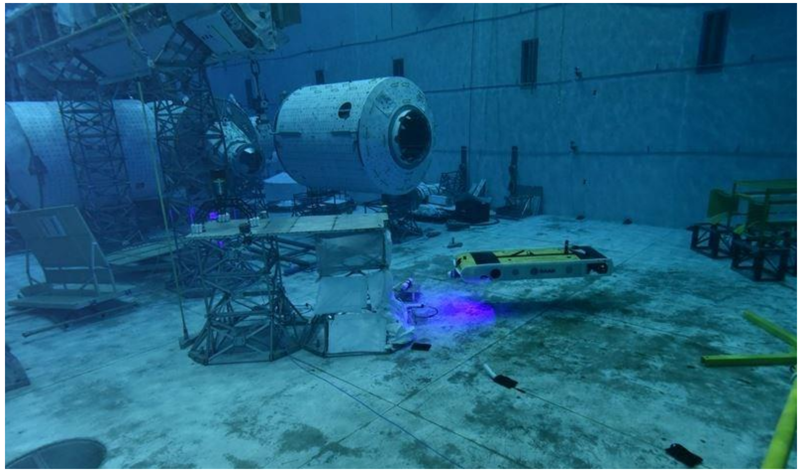
Mine Countermeasures

“Mine Countermeasures”, General Dynamics, n.p.n.d., May 16, 2011

Search & Rescue

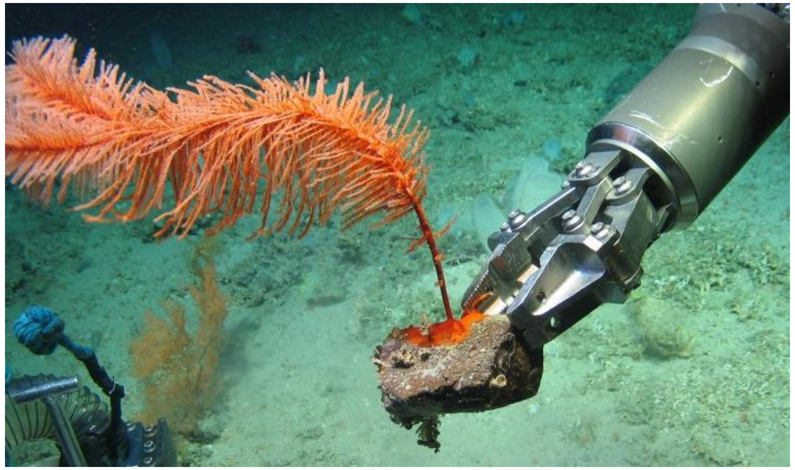
US Coast Guard Office of Search and Rescue (CG-SAR)

Oil & Gas Industry



“Saab Sabertooth AUV/ROV for Oil & Gas Inspection”, Saab, Jan 26,2018

Wildlife Habitat Monitoring



Deep-sea coral specimen collection. Plymouth University et al. 2018

Introduction

Objective: Develop a computationally efficient time-risk optimal motion planner for variable-speed autonomous vehicles in obstacle-rich environments.

Existing Approaches

$T^\star$ for Time-Risk Optimal Motion Planning

- The **$T^\star$ algorithm** [1] is the only motion planner that considers multi-speed vehicles and jointly optimizes time and risk.
- **Limitation**: Grid-based approach is computationally expensive

Sample-based Methods for Rapid Motion Planning

- RRT* and PRM* [2] quickly generate asymptotically-optimal shortest paths as number of samples increase
- **Limitation**: Restricted to single-speed vehicles & no risk considered

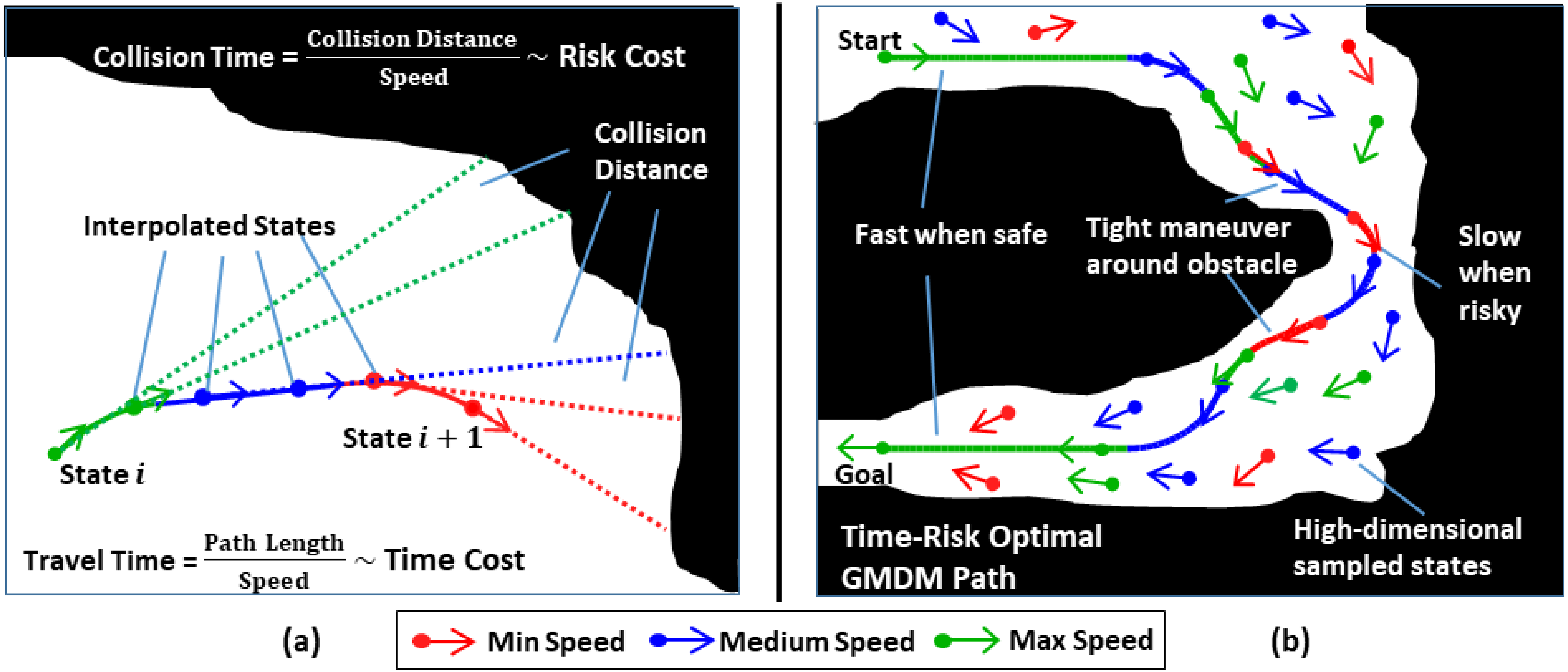
Kinodynamic Motion Models

- Dubins [3] provides shortest paths for single velocity vehicles
  - **Limitation**: Does not consider multi-speed vehicles
- Wolek [4] provides time-optimal paths for multi-speed vehicles
  - **Limitation**: Requires nonlinear solvers
- The recently developed ***Generalized Multi-speed Dubins Motion Model (GMDM)*** [5] overcomes the above limitations:
  - Better maneuvering by controlling the turning radius
  - Speed selected based on obstacle distance to mitigate risk
  - Allows for real-time computation

Features and Contributions of $T^\star$-Lite

- Enables fast time-risk optimal motion planning for variable-speed vehicles by:
  - Porting the novel time-risk cost function from $T^\star$ into a fast and asymptotically-optimal sample-based motion planner
  - Generating samples from a four-dimensional configuration space considering position, heading, and speed.
  - Utilizing the GMDM to produce the optimal time-risk trajectories connecting sampled states
- Algorithm is computationally efficient while providing reasonable solution quality

T*-Lite Overview



(a) Overview of the computation of the time and risk costs in the joint optimization problem.

(b) Example of the high-dimensional sampled vehicle states and the time-risk optimal trajectory produced by the Generalized Multi-speed Dubins Motion Model.

Problem Formulation

Autonomous Vehicle Description & Search Area

Autonomous Vehicle Description

☐ $(x, y, \theta) \in SE(2)$ is the vehicle and position heading

☐ Taking speed $v(t)$ and turning rate $u(t)$ be the inputs, the equations of motion are:

$$\begin{cases} \dot{x}(t) = v(t) \cdot \cos\theta(t) \\ \dot{y}(t) = v(t) \cdot \sin\theta(t) \\ \dot{\theta}(t) = u(t) \end{cases}$$

Bounded Turning Rate
$u(t) \in [-u_{\max}, u_{max}]$ rad/s, $u_{\max} \in \mathbb{R}^+$ is the max turning rate, and “-/+” indicates a *right/left* turn

Variable Speed
$v(t) \in [v_{min}, v_{max}]$ m/s

Curvature
$\kappa(t) = \frac{u(t)}{v(t)}, \quad 0 \leq |\kappa(t)| \leq \frac{u_{max}}{v_{min}}$

Note: curvature is the inverse of turning radius: $r(t) = \frac{1}{\kappa(t)}$

Search Area: $A \in \mathbb{R}^2$

Define vehicle state as $\mathbf{p} = (x, y, \theta, v)$

☐ $(x, y) \in A_{free}$ ☐ $\theta \in [0,2\pi)$ ☐ $v \in [v_{min}, v_{max}]$

Problem Formulation

Time-Risk Cost Function

Admissible Control: Let Γ denote the set of collision-free paths between the start state $\mathbf{p}_{start}$ and goal state $\mathbf{p}_{goal}$. For each path $\gamma \in \Gamma$, the control $\boldsymbol{c}(s) = (\boldsymbol{\kappa}, v)$ at any point s on path γ, belongs to:

$$\boxed{\Omega = \left\{(\kappa, v) \colon v_{\min} \leq v \leq v_{max}, |\kappa| \leq \frac{u_{\max}}{v}\right\}}$$

Cost of a Path: Let $R(s)$ denote the risk cost at point s on path γ. Then the total cost is written as:

$$\boxed{J(\gamma) = \int_{\gamma} \underbrace{\boxed{R(s)}}_{\text{risk cost}} \cdot \underbrace{\boxed{\frac{1}{v(s)}}}_{\text{time cost}} ds}$$

Objective: Find the optimal control $c^{\star} \in \Omega$, which generates the collision-free path $\gamma^{\star}$, such that: $J(\gamma^{\star}) \leq J(\gamma), \forall \gamma \in \Gamma$ in a computationally efficient manner

T*-Lite Algorithm

Approximate Time-Risk Cost Function

Approximate Piecewise Path Cost Function: Assume a constant risk along path $\gamma_{i,i+1}$. Thus:

$$J(\gamma_{i,i+1}) = \boxed{R(\gamma_{i,i+1})} \cdot \boxed{\int_{\gamma_{i,i+1}} \frac{1}{v(s)} ds}$$

risk cost time cost

Risk Cost $R(\gamma_{i,i+1})$: For each evenly interpolated state $\hat{p}_\ell$ along $\gamma_{i,i+1}$:

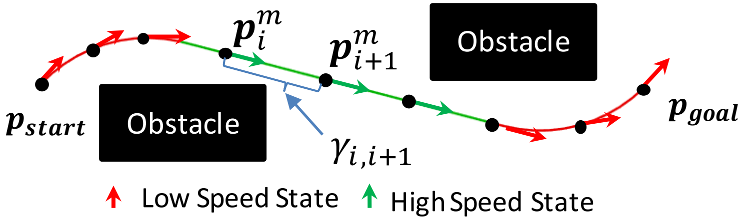
1. Compute ***collision time*** $t_\ell = \frac{d_l}{v_\ell}$
2. Given safety threshold $t^\star$, compute sample risk:

$$risk(\hat{p}_\ell) = \begin{cases} 1 + \log\left(\frac{t^\star}{t_\ell}\right) & \text{if } t_\ell < t^\star \\ 1 & if\ t_\ell \geq t^\star \end{cases}$$

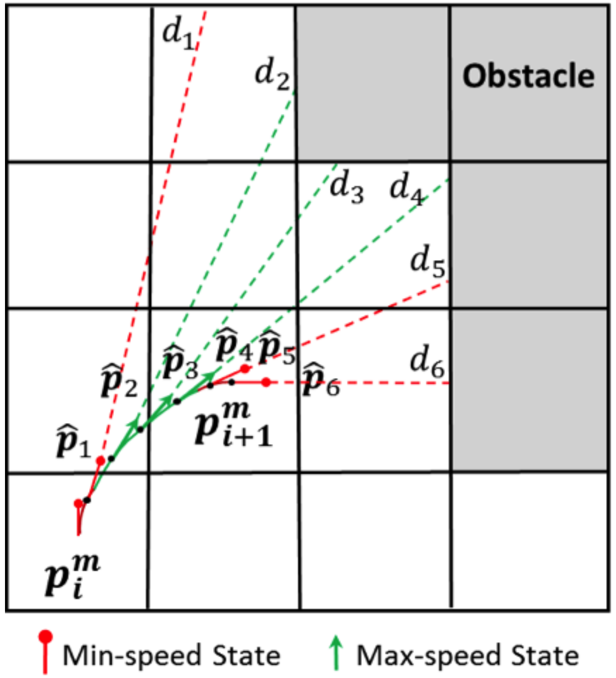
Finally, the piecewise risk is computed as:

$$R(\gamma_{i,i+1}) = \max_{\ell \in \{1,\ldots,M\}} (risk(\hat{p}_\ell))^k$$

☐ $k > 0$ is the user-defined *risk weight*



An example of the interpolated state sequence P^m composed of states $\boldsymbol{p}_i^m = (x_i, y_i, \theta_i, v_i)$



Sample states $\hat{\boldsymbol{p}}_\ell$ along collision-free path $\gamma_{i,i+1}$ and corresponding collision distances d_ℓ

T*-Lite Algorithm

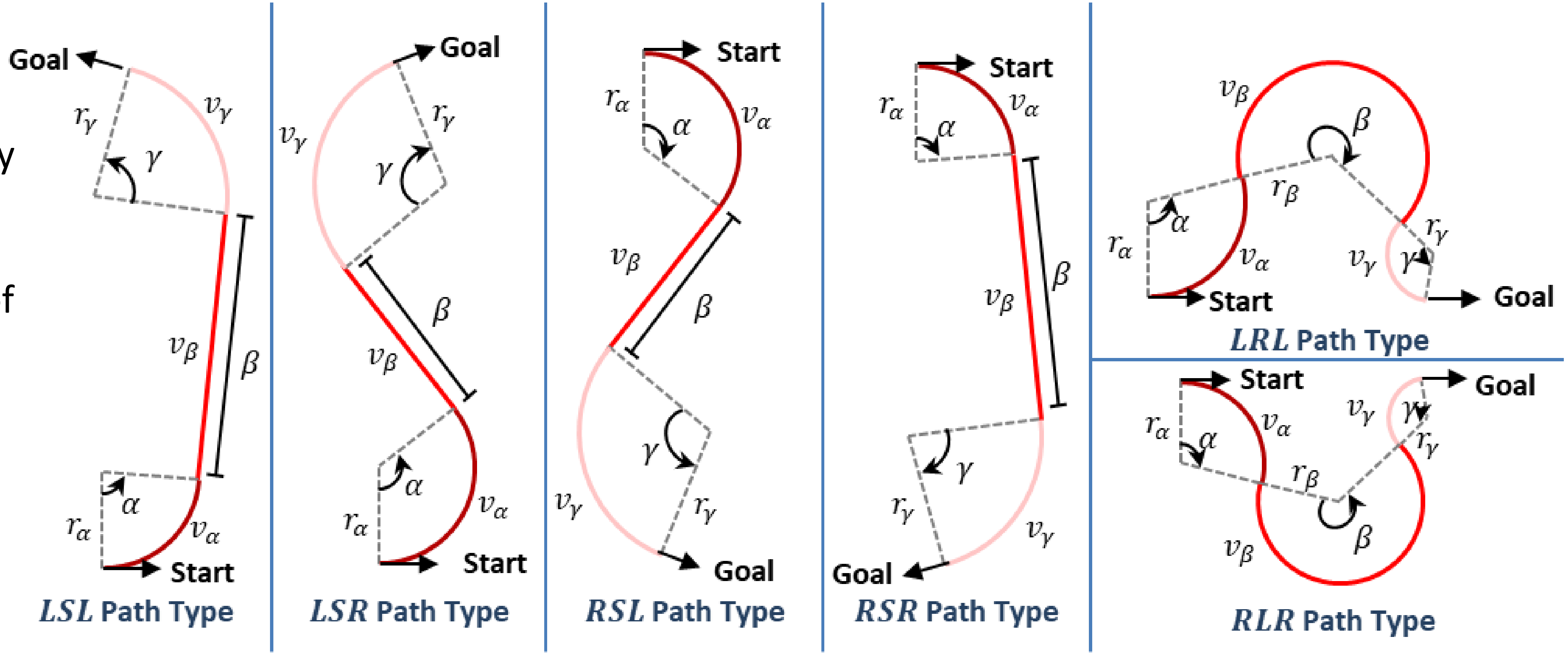
Kinodynamic Motion Model

The **Generalized Multi-speed Dubins Motion Model (GMDM)** is a fundamental improvement of the Dubins model that enables the selection of any speed for any of the three segments of a Dubins path ($L \equiv$ left turn, $S \equiv$ straight, or $R \equiv$ right turn).

- Used to generate a set of candidate trajectories that connect any two states $\mathbf{p}_i$ to $\mathbf{p}_{i+1}$

Main Features:

- Provides path planners the flexibility to select the appropriate speed dynamically based on the perceived risk
- Selection of both turning rate and speed enables selection of appropriate turning radius to smoothly maneuver around obstacles based on their shapes and sizes
- Synthesis is similar to Dubins, thus providing simple closed-form solutions for real-time computation.



LSL Path Type

LSR Path Type

RSL Path Type

RSR Path Type

LRL Path Type

RLR Path Type

Visualization of the Generalized Multi-speed Dubins Motion model for each of the six path types.

T*-Lite Algorithm

RRT* Motion Planner

T*-Lite utilizes the asymptotically-optimal sample-based RRT* framework, which has six core functions:

- ☐ *Nearest neighbor*
- ☐ *Near-by vertices*
- ☐ *Collision check*

no changes from RRT*

- ☐ *Sampling*
- ☐ *Distance*
- ☐ *Local steering*

updated in T*-Lite

Summary of RRT*

- ☐ Sampled states are randomly generated in the obstacle-free space
- ☐ A search tree of minimum-cost collision-free paths that connect states to the start node is created
- ☐ As new states are added, connections between existing states are updated if connections to the new states are faster.

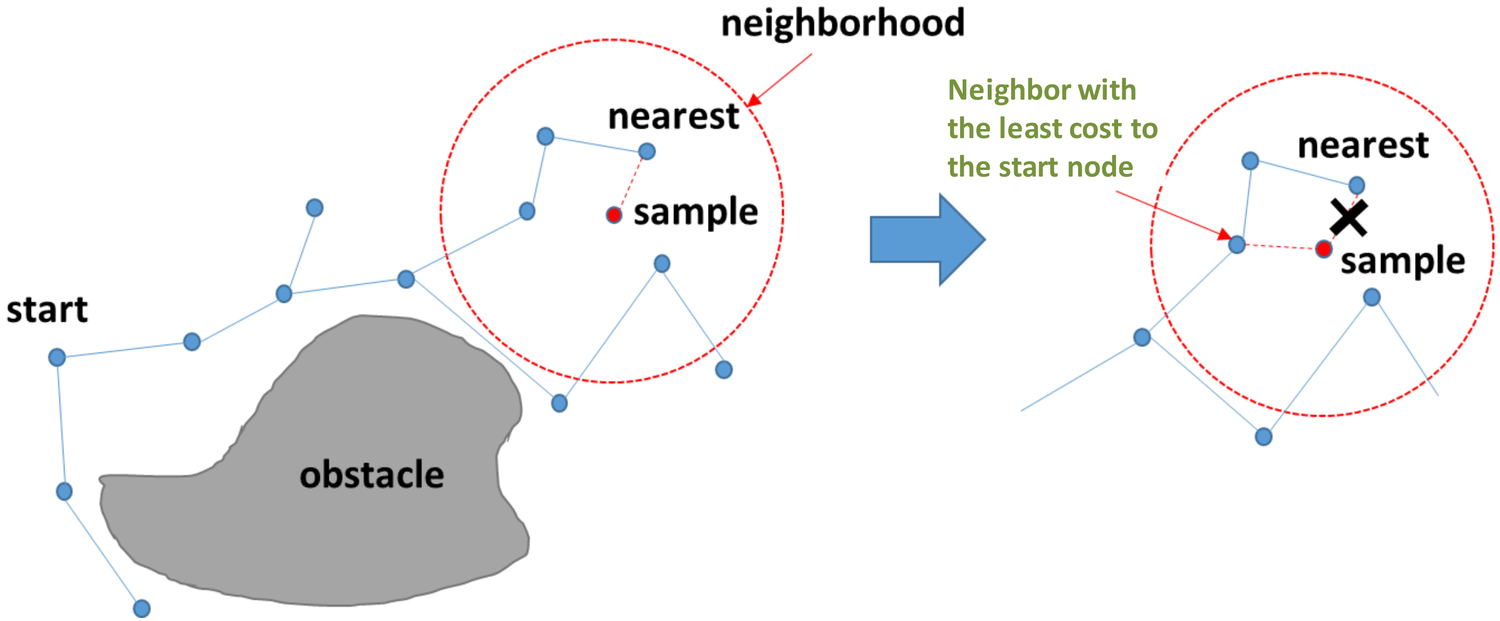


Figure: Illustration of RRT* and the iterative search-tree update for a point vehicle.
Note: connections between nodes in T*-Lite are subject to curvature constraints.

T*-Lite Algorithm

Core Functions

T*-Lite is based on the asymptotically-optimal sample-based RRT* framework, which has six core functions:

☐ *Nearest neighbor*
☐ *Near-by vertices*
☐ *Collision check*

no changes from RRT*

☐ *Sampling*
☐ *Distance*
☐ *Local steering*

updated in T*-Lite

Sampling Function*:* generates randomly sampled collision-free states states $\mathbf{p} = (x, y, \theta, v) \in \mathbf{P}$ in the obstacle-free space A_{free}.

Distance Function: Let $dist: \mathbf{P} \times \mathbf{P} \rightarrow \mathbb{R}^2$ be a function that returns the cost of the time-risk optimal trajectory $\gamma_{i,i+1}^*$ between two states $\mathbf{p}_i, \mathbf{p}_{i+1} \in \mathbf{P}$ such that $dist(\mathbf{p}_i, \mathbf{p}_{i+1}) = J(\gamma_{i,i+1}^*)$.

Local Steering Function*:* Given two states $\mathbf{p}_i, \mathbf{p}_{i+1} \in \mathbf{P}$, the *steer* function produces the optimal collision-free trajectory $\gamma_{i,i+1}^*$ connecting $\mathbf{p}_i$ to $\mathbf{p}_{i+1}$ such that $J(steer(\mathbf{p}_i, \mathbf{p}_{i+1})) = dist(\mathbf{p}_i, \mathbf{p}_{i+1})$. Producing the optimal trajectory requires:

☐ The **approximate optimization function from T*** to evaluate the time-risk costs of the created candidate trajectories
☐ A **kinodynamic motion model** to create a sufficient set of candidate trajectories connecting two states

Simulation Setup

Autonomous Vehicle and T*-Lite Parameters:

- ☐ $(v_{\min}, v_{\max}) = (0.5{,}1.0)\ m/s$
- ☐ $u_{\max} = 0.5\ rad/s$
- ☐ Safety threshold $t^\star = 6\ s$
- ☐ Risk weight $k\ = 2$
- ☐ Num. of interpolated states $M\ = 4$
- ☐ Search tree max size: 3000 sampled states
- ☐ Num. of nearest neighbors: 100
- ☐ Max. connection distance: $3m$
- ☐ Scenario Size: $30m \times 30m$

Motion models used in T*-Lite:

- ☐ Max-speed Dubins motion model
- ☐ Generalized Multi-speed Dubins Motion model

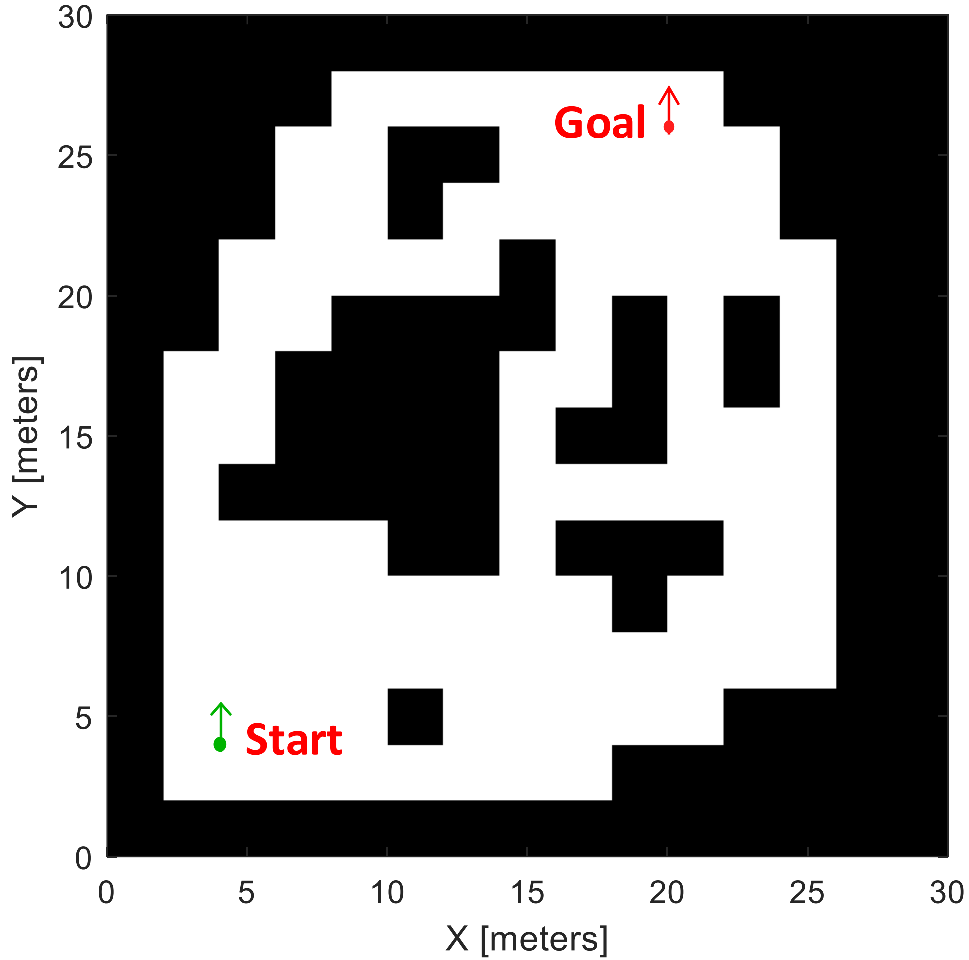
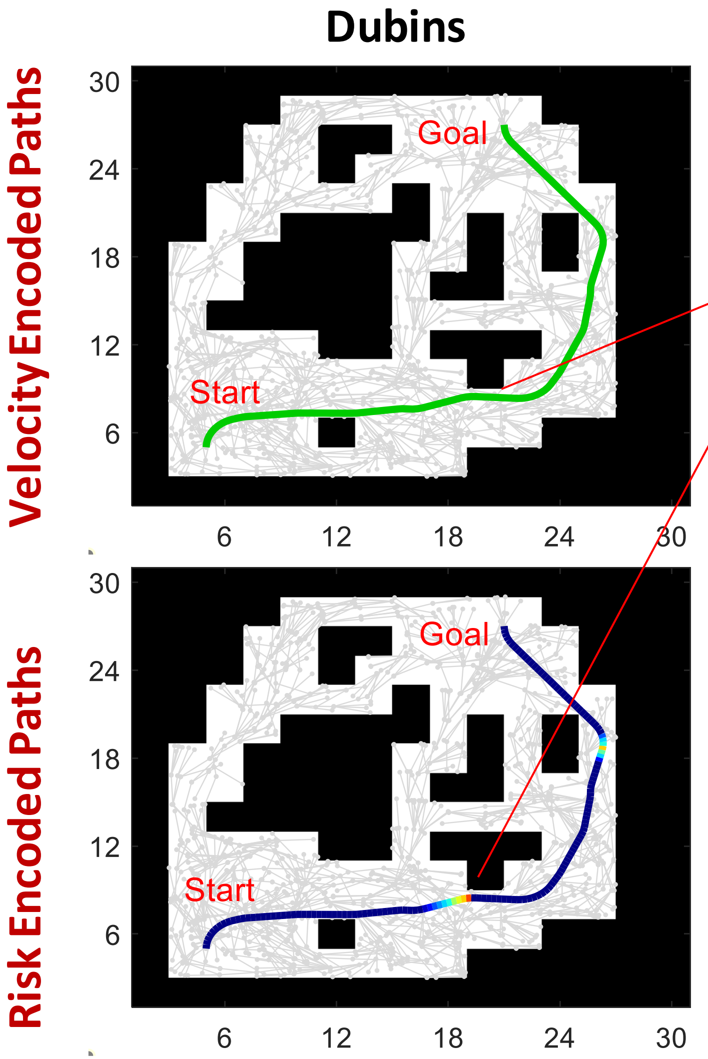


Figure: Scenario used in simulation

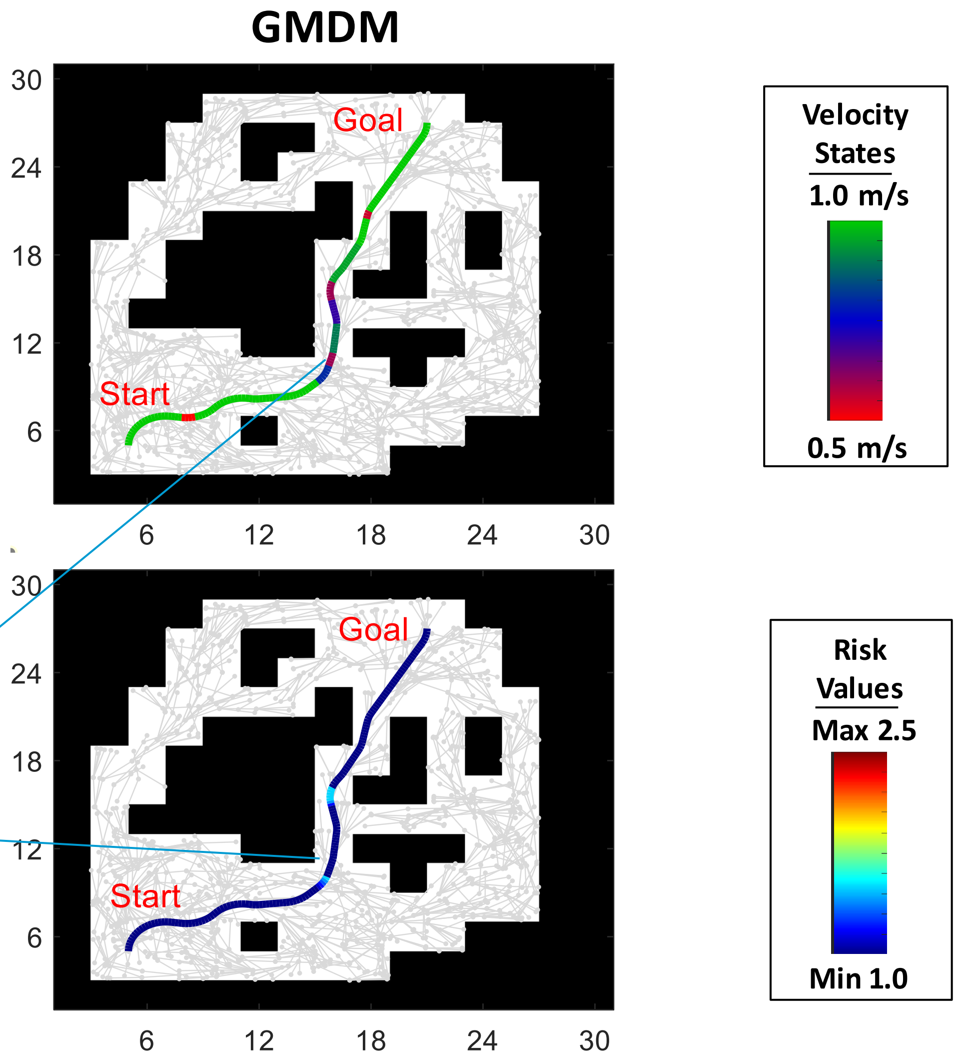
Results



Travel Time: 44.56 s **Max Risk:** 2.13
Time-Risk Total: 60.47 **CPU Time:** 1.34 s

Single speed reduces CPU requirements but produces longer and riskier paths

Variable speeds enhance maneuverability and reduce risk to provide quicker and safer paths



Travel Time: 39.23 s **Max Risk**: 1.69
Time-Risk Total: 45.78 **CPU Time:** 8.65 s

Conclusions & Future Work

Conclusions

- Developed $\mathrm{T}^\star$-Lite for rapid time-risk optimal motion planning for variable-speed autonomous vehicles. Achieved by:
  - Porting the novel time-risk cost function from $\mathrm{T}^\star$ into the RRT* framework
  - Generating high-dimensional samples that considers vehicle position, heading, and speed.
  - Utilizing the Generalized Multi-speed Dubins Motion model to provide near-optimal trajectories in a computationally efficient manner
- Provides fast, safe, and flexible maneuvers in obstacle-rich environments
- Suitable for on-demand real-time motion planning

Future Work

- In-depth analysis of the $\mathrm{T}^\star$-Lite framework in other asymptotically optimal sample-based frameworks.
- Direct comparisons against the grid-based $\mathrm{T}^\star$ in terms of both solution quality and CPU time.
- Develop smart high-dimensional sampling methods for multi-speed vehicles to further enhance solution quality and reduce computation time.
- Extend to multi-agent resilient systems.

References

1. J. Song, S. Gupta, and T. A. Wettergren, “$T^\star$: Time-optimal risk-aware motion planning for curvature-constrained vehicles,” IEEE Robotics and Automation Letters, vol. 4, pp. 33–40, Jan 2019.
2. S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” The International Journal of Robotics Research, vol. 30, no. 7, pp. 846–894, 2011.
3. L. Dubins, “On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents,” American Journal of Mathematics, vol. 79, no. 3, pp. 497–516, 1957.
4. A. Wolek, E. Cliff, and C. Woolsey, “Time-optimal path planning for a kinematic car with variable speed,” Journal of Guidance, Control, and Dynamics, vol. 39, no. 10, pp. 2374–2390, 2016.
5. J. P. Wilson, K. Mittal, and S. Gupta, “Novel motion models for time-optimal risk-aware motion planning for variable-speed AUVs,” in OCEANS 2019 MTS/IEEE SEATTLE, pp. 1–5, 2019.

Thank You!