Convex Optimization Approach for Car-Like Robot Trajectory Planning

Xiyang Wu, Yutao Chen
wuxiyang@umd.edu, cheny@umd.edu

December 17, 2021

1 Introduction

Planning a collision-free and dynamically feasible trajectory is a fundamental problem in robotics. The common strategy of getting a collision-free, near-optimal, and dynamically feasible trajectory includes two steps [1]. The first step is to explore the unknown environment and figure out a feasible collision-free trajectory, usually realized by excluding some kinematic properties of the vehicle in planning. Popular methods for this task are sample-based, like Rapidly-exploring random tree (RRT) [2] and Probabilistic Roadmap (PRM) [3], or search-based, like A* algorithm [4] and Dynamic Window Approach [5]. Sampling-based trajectory planning algorithms, though always more efficient in finding a collision-free trajectory under an ambiguous and uncertain scenario, compared with search-based methods, especially environments with complicated obstacles, trajectories generated through these methods are typically highly sub-optimal and dynamically infeasible since they ignored the real-world kinematics during their trajectory planning.

To make the proposed trajectory near-optimal and dynamically feasible, one idea is to introduce the shortcut policy into trajectory optimization. Algorithms that utilize this idea include RRT* [6], which re-organizes the connectivity among nodes after RRT exploration, and RRT*-Smart [7] that substitute the creeping collision-free trajectory between two nodes with straight lines. Though more practical in general, trajectories optimized with these shortcut methods may not necessarily satisfy the real-world kinematics model due to the sharp turns that may be introduced by the shortcut trajectories. Another idea is to formulate the trajectory optimization problem as a convex optimization problem. Taking the whole trajectory as the optimization object, optimization-based methods make it possible to find a more globally optimal and dynamically feasible trajectory via convex optimization solvers than their shortcut counterparts.

Proposed by [8], a convex-optimization-based trajectory planning method named Convex Elastic Smoothing (CES) is designed to smooth and optimize trajectories acquired with sampling-based trajectory planning methods. This method separates the general trajectory optimization problem into two sub-problems for optimization and incorporates two convex optimization algorithms to solve these problems, Elastic Band Stretching method [9] for trajectory shape optimization and Minimum-Time Speed Optimization Subroutines (MTSOS) [10] algorithm for velocity optimization, using an iterative optimization structure. This method reveals its superior performance on the smoothness of the trajectory shape and optimality in traversal time over trajectory after optimization, compared with other existing methods.

In this project, we plan to implement the Convex Elastic Smoothing method on optimizing sub-optimal trajectories generated by several baseline sampling-based trajectory planning algorithms, including Rapidly-exploring random tree (RRT), Probabilistic Roadmap (PRM), and Fast Marching Trees (FMT) [11]. Trajectories generated through these algorithms will be taken as reference trajectories. The performance of the trajectory optimization algorithm will be evaluated under several diverse 2D trajectory planning scenarios,

including sparse circular maze, compacted circular maze, and lane changing. Result evaluation in each experiment is based on the following criteria: (a) Optimality: The overall length of the generated trajectory and the length reduction percentage compared with the reference trajectory (b) Efficiency: The traversal time from the start to the goal over the generated trajectory (c) Time Complexity: The time required for the trajectory optimization algorithm on each reference trajectory.

As a potential modification to improve the optimality of the trajectory, an artificial potential field (APF) [12] will be introduced in the trajectory planning for further optimization, The general idea is to assign repulsive potential on obstacles and attractive potential on the goal, compared with other areas. Differentiating potentials throughout the scenario helps the vehicle figure out the passable places and keep trajectories far enough to obstacles. The effect of the trade-off coefficient between the general will also be covered by setting different trade-off coefficients for trajectory optimization and comparing trajectories generated under different scenarios.

This report is organized as follows: In Section 2, we state the problem we are trying to solve by proposing the mathematical model and vehicle kinematics model. In Section 3, we illustrate the CES algorithm covered in this project and our modification over CES algorithm with the corresponding formulated convex optimization problem structure. In Section 4, we propose several experiments under different scenarios and discuss the results achieved. The trade-off between CES optimization and artificial potential field will also be included. In Section 5, we draw some conclusions based on the experiment results and discuss some potential directions for further research.

2 Problem Statement

The statement for the convex optimization problem we are trying to investigate is: Given a vehicle that follows the friction circle car model [10] located in a rectangular workspace with multiple obstacles, a collision-free reference trajectory is computed by one of the baseline trajectory planning algorithms. The goal is to smooth the reference trajectory via some convex optimization-based methods, using the information about the vehicle's model, obstacles, and the reference trajectory. Meanwhile, the speed profile throughout the trajectory proposed will also be optimized such that the vehicle can traverse the optimized trajectory with minimum time.

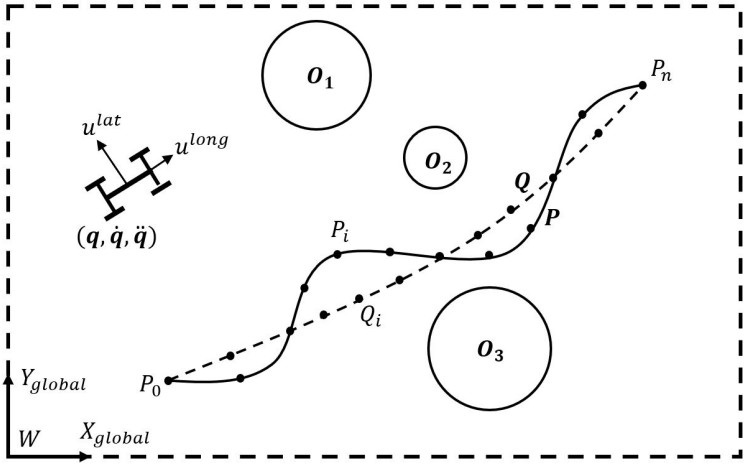


Figure 1: The Overall Setting for Convex Trajectory Optimization Problem

Following the dynamic model proposed in [8] and [10], the overall setting for our convex trajectory optimization problem is given in Figure 1. Let $q \in \mathbf{R}^2$ be the position of the vehicle on a two dimensional

workspace $\mathcal{W}$, where its coordinate is (X_{global}, Y_{global}). Take the first and second derivative of q, $\dot{q}$ and $\ddot{q}$, as the vehicle's velocity and acceleration, respectively. Consider a non-drifting and non-reversible vehicle with mass m, so that the heading of the vehicle is the same as the direction of the instantaneous velocity. $\phi(\dot{q})$ is the orientation angle of the vehicle, while $\phi : \mathbf{R}^2 \to \mathbf{R}$ maps the vehicle's position to the direction that the vehicle's velocity is heading. Two dimensional control input is denoted as $u = [u^{long}, u^{lat}]$, where u^{long} is the longitudinal force and u^{lat} is the lateral force. The obstacle set is $\mathcal{O} = \{O_1, O_2, ..., O_m\}$, with $O_i \subset W$. The reference trajectory is given by the node set $\mathcal{P} = \{P_1, P_2, ..., P_n\}$ and the trajectory composed by vehicle positions Q_i with respect to P_i during optimization is given by $\mathcal{Q} = \{Q_1, Q_2, ..., Q_n\}$, while $P_i, Q_i \subset \mathcal{W} \setminus \mathcal{O}$. The first and last noded of reference trajectory $\mathcal{P}$, P_1 and P_n are the start and goal of the trajectory planning task. In this case, the dynamic model of the vehicle is given by

$$m\ddot{q} = \begin{bmatrix} \cos\phi(\dot{q}) & -\sin\phi(\dot{q}) \\ \sin\phi(\dot{q}) & \cos\phi(\dot{q}) \end{bmatrix} u.$$

The constraint set $\mathcal{C}$ given by the friction circle constraint for $(\dot{q}^2, \ddot{q}, u)$ is

$$\mathcal{C} = \{(\dot{q}^2, \ddot{q}, u) | ||u||_2 \leq \mu mg, u^{long} \leq W_f \mu mg\}.$$

where μ is the friction coefficient, g is the gravitational acceleration, and W_f is the percentage of vehicle weight on the front tyres, which marks the upper bounded of longitudinal force applied on the vehicle. We note the upper bound of longitudinal force as $U^{long} = W_f \mu mg$.

We also assume there is a minimum turning radius R_{min} for the vehicle, which induces a constraint on the lateral force. More precisely,

$$u^{lat} \leq m\frac{||\dot{q}||^2}{R_{min}}.$$

3 Methodology

3.1 Baseline Trajectory Planning Algorithms

The first stage of this project is to generate the reference trajectory between the start and the goal in the given trajectory planning scenario. Reference trajectories made by three baseline sampling-based trajectory planning algorithms, Rapidly-exploring random tree (RRT), Probabilistic Roadmap (PRM) and Fast Marching Trees (FMT). The successive optimization problems and experiments are based on them. In this stage, the vehicle is taken as a mass point without extra dynamical constraints mentioned in Section 2.

3.1.1 Rapidly-exploring random tree (RRT)

Rapidly-exploring random tree (RRT) is an efficient sampling-based trajectory planning algorithm for non-convex, high-dimensional spaces by randomly establishing a space-filling tree. Consider a trajectory planning problem of finding a feasible trajectory in a two-dimensional space, same as our problem setting. The RRT algorithm sets up a tree rooted at the starting point. In each iteration, a sample is drawn randomly within the search space, and the tree expands towards the sample position by following some predefined growing mechanisms, like the vehicle dynamic model or a fixed step size measured by some norm (e.g., L_2 norm). When the trajectory branch between the parent and child node is feasible (i.e., passes entirely through free space and obeys any constraints), the child node and its trajectory will be added to the tree. Otherwise, this trajectory branch will be discarded. The expansion ends when the tree reaches the predefined neighborhood region of the goal. Trajectories made by RRT could be dynamically feasible, though the iteration number for finding a feasible trajectory may be high limited iterations, and trajectory marked by RRT usually consists of more nodes. In this project, the RRT trajectory generation procedure follows the dynamic model defined in Section 2 in the trajectory branch extension and collision detection functions.

3.1.2 Probabilistic Roadmap (PRM)

Probabilistic Roadmap (PRM) is another sampling-based trajectory planning method. This algorithm first samples from the free space (i.e., the search space excluding the obstacles) randomly to generate a collision-free graph that includes the start point and goal. With the collision-free graph, PRM uses the graph search algorithm to find out the optimal path from the start point and goal. The algorithm we are using in this project is A^*, or best-first search, which ranks the feasible, explored paths from the start by the summation of the cost function f, which is the length of the finished paths, and the estimation function g, which is the distance between the current position and the goal. To generate the reference trajectory, the finished need to interpolate in some level to generate a trajectory with intensively distributed nodes. Like other algorithms presented in this section, the optimal solution is not guaranteed to find by PRM. To find a feasible reference trajectory made by PRM, we need to adjust the hyperparameters for PRM including the searching scope for the node connectivity, the sample number, and the iteration number.

3.1.3 Fast Marching Trees (FMT)

Fast Marching Trees (FMT) searches for a feasible trajectory over a predetermined number of sampled points. It will maintain and expand a tree structure that stores the explored nodes following some mechanisms. Sampled points are classified into three classes. The first class is the set of open nodes. It is denoted by V_{open} and contains the leave nodes of the tree. The second class is the set of closed nodes. It is denoted by V_{closed} and contains the non-leave nodes. The third class is the set of unvisited nodes. It is denoted by $V_{unvisited}$ and contains all the sampled points that are not in the tree. For each node, there exists a cost associated with it. The cost function can be freely chosen like the weighted sum of the distance to the tree root, which is the start point, and the distance to the goal. Then, the algorithm will choose the node z with the lowest-cost from the set V_{open} and find its neighbors within $V_{unvisited}$. Given a neighboring node x, the algorithm finds the neighbors of x within V_{open} and searches for a locally-optimal one-step connection that has no intersection with any of the obstacles, then add this connection to the tree. After all neighbors of z in $V_{unvisited}$ have been explored, the algorithm adds successfully connected nodes to V_{open}, places z in V_{closed}, and proceeds to the next iteration. The algorithm ends when the tree reaches the area considered to be the goal. This algorithm is proven to be asymptotically optimal and reveals its ability in converging to an optimal solution faster than its state-of-the-art counterparts.

3.2 Elastic Band Method

In this section, we follow the discussion about the elastic band method in [8]. Originally from [9], the general idea of the elastic band method is straightforward. The trajectory marked by a series of nodes $\mathcal{Q}$ could be taken as a tensioned elastic band, fixed at the start and goal point, and with forces applied on each node. The objective function is the summation of forces applied on this band, while its amount is proportional to the distance between nodes, following Hooke's Law. The goal is to minimize the overall force applied on the elastic band and generate a feasible, collision-free trajectory. To achieve this goal, the first step is to generate a set of collision-free bubbles with respect to the obstacles around the node over the reference trajectory. The bubble set forms the domain for the convex optimization problem that minimizes the objective function, the summation of forces applied on the band, which can be solved efficiently using well-established convex optimization solvers.

3.2.1 Bubble Generation

The bubble generated with respect to P_i is denoted by $\mathcal{B}_i$, while A_i, r_i are the center coordination and radius of the bubble, respectively. Since our problem is formulated in a two-dimensional space, bubbles are indeed

circles, i.e the set represented by $\mathcal{B}_i$ is $\mathcal{B}_i = \{x \in \mathbf{R}^2 | ||x - A_i||_2 \leq r_i\}$. Each bubble must satisfy following conditions during their generation.

1. The radius is upper bounded by $r_u > 0$. Under this condition, we limit the smoothing procedure within a relatively small portion of the work space.

2. The radius is no less than $r_l > 0$, if possible. The lower bounded is determined by the maximum distance between adjacent P_i. Under this condition, every bubble overlaps with its neighboring bubbles and the placement of new P_i for trajectory stretching does not have any "gaps".

3. A_i is as close as possible to P_i.

The pseudocode can be found in Algorithm 1. After loading the reference trajectory $\mathcal{P}$, obstacle set $\mathcal{O}$, bubble bounds r_l and r_u, bubbles located within the scope marked by the half radius of the previous bubble will be merged with their previous bubbles. The function $GenerateBubble(P_i)$ calculates the largest bubble centered at P_i that is collision-free and the radius within the range $[r_l, r_u]$. $TranslateBubble(A_i, r_i)$ identifies the edge of the obstacle closet to A_i and computes the outward normal direction to the edge so that the center of the bubble is moved along such direction until a ball of radius r_l can be placed. If still impossible, $TranslateBubble(A_i, r_i)$ returns the ball of the largest radius among the balls whose centers lie in the aforementioned normal direction. Finally, the algorithm returns a set of bubbles that forms a collision-free "tube" so that we can optimize (adjust) P_i within the "tube" during the elastic band stretching process.

Algorithm 1 Bubble Generation

Require:
Reference trajectory $\mathcal{P}$, obstacle set $\mathcal{O}$, bubble bounds r_l and r_u.

for $i = 2 : (n-1)$ **do**
  if $||P_i - A_{i-1}|| < 0.5 \cdot r_{i-1}$ **then**
    $\mathcal{B}_i \leftarrow \mathcal{B}_{i-1}$
    $A_i \leftarrow A_{i-1}$
    $r_i \leftarrow r_{i-1}$
    Continue
  end if
  $\mathcal{B}_i \leftarrow GenerateBubble(P_i)$
  $r_i \leftarrow Radius\ of\ \mathcal{B}_i$
  $A_i \leftarrow P_i$
  if $r_i < r_l$ **then**
    $\mathcal{B}_i \leftarrow TranslateBubble(A_i, r_i)$
    $r_i \leftarrow Radius\ of\ \mathcal{B}_i$
    $A_i \leftarrow Center\ of\ \mathcal{B}_i$
  end if
end for

3.2.2 Elastic Band Stretching

The general idea of Elastic Band Stretching is to take the reference trajectory $\mathcal{P}$ as an elastic band with forces applied on n nodes. By adjusting "forces" applied on these nodes, their positions could change within the feasible region marked their corresponding bubbles. The resulting nodes are regarded as new nodes for the smoothed trajectory $\mathcal{Q}$. To smooth a trajectory, the goal is then to place new nodes $Q_1, ..., Q_n$ within the

bubbles so as to minimize the sum of the norms of the balance forces subject to constraints with respect to the vehicle's dynamic model. The balance force $\mathbf{N}_k$ at node Q_k is

$$\mathbf{N}_k = \mathbf{F}_{k-1} - \mathbf{F}_k, k = 1, ..., n$$

where $\mathbf{F}_k = Q_{K+1} - Q_k$ is the artificial tensile force between Q_k and Q_{k+1}.

To ensure the convexity of the optimization problem proposed on the elastic band, a series of approximations are necessary to be applied on the constraints for the placement of Q_i. Assume the longitudinal force u^{long} and velocity v_k are given at each node Q_k and denote R_k as the instantaneous turning radius for the car at the kth node, u^{lat} can be upper bounded by

$$u^{lat} = \frac{||v_k||^2}{R_k} \leq \sqrt{(\mu\, g)^2 - (\frac{u_k^{long}}{m})^2} \triangleq \alpha_k.$$

Hence, we have $\frac{1}{R_k} \leq \frac{\alpha_k}{||V_k||^2}$. To apply this inequality to N_k, we make the following approximations.

- $||F_k|| \cong ||F_{k-1}||$,
- θ_k is small,
- $\theta_k \cong \frac{||Q_{k+1} - Q_||}{R_k}$.

These approximations are valid when the nodes that belong to $\mathcal{Q}$ are uniformly distributed over a trajectory and dense "enough". In this case, we have

$$\begin{aligned} ||N_k|| = ||F_{k-1} - F_k|| &\cong 2 \cdot ||F_k|| \cdot \sin(\frac{\theta_k}{2}) \\ &\cong ||Q_{k+1} - Q_k||\theta_k \cong ||Q_{k+1} - Q_k||^2 / R_k \\ &\leq ||Q_{k+1} - Q_k||^2 \alpha_k / ||v_k||^2. \end{aligned} \tag{1}$$

Lastly, we formulate the following equation as a quadratic function by approximating $Q_{k+1} - Q_k$ as the average length d for the intervals marked by Q_k along the reference trajectory. More precisely,

$$||Q_{k+1} - Q_k|| \approx d \triangleq \frac{\sum_{k=1}^{n-1} ||P_{k+1} - P_k||}{n-1}.$$

Combining with (1), we can conclude the following.

$$||N_k|| = |2Q_k - Q_{k-1} - Q_{k+1}|| \leq \min\{\alpha_k \left(\frac{d}{||v_k||}\right)^2, \frac{d^2}{R_{min}}\}.$$

where these constraints need to hold at the same time, $\alpha_k \left(\frac{d}{||v_k||}\right)^2$ and $\frac{d^2}{R_{min}}$ are constant values given v_k, a_k, d for each elastic band stretching problem. Then, we can formulate the optimization problem as

$$\underset{Q_3, ..., Q_{n-2}}{\text{minimize}} \quad \sum_{k=2}^{n-1} ||2Q_k - Q_{k-1} - Q_{k+1}||^2 \tag{2a}$$

$$\text{subject to} \quad Q_1 = P_1, \quad Q_2 = P_1 + d \cdot \frac{v_1}{||v_1||}, \tag{2b}$$

$$Q_n = P_n, \quad Q_{n-1} = P_1 - d \cdot \frac{v_{n-1}}{||v_{n-1}||}, \tag{2c}$$

$$Q_k \in \mathcal{B}_k = \{x \in \mathbf{R}^2 | ||x - A_k||_2 \leq r_k\}, \quad k = 3, ..., n-2, \tag{2d}$$

$$||2Q_k - Q_{k-1} - Q_{k+1}|| \leq \min\left\{\alpha_k \left(\frac{d}{||v_k||}\right)^2, \frac{d^2}{R_{min}}\right\}, \quad k = 2, ..., n-1 \tag{2e}$$

Note that $N_k = 2Q_k - Q_{k-1} - Q_{k+1}$ by definition. Constraints presented by (2b) and (2c) is to maintain the position and heading angles of the starting point and the goal point unchanged. Problem (2) is a QCQP because of the existence of convex quadratic objective function (2a) and convex quadratic inequality constraints mentioned by Constraint (2d) and (2e), so that a quadratic programming solver is required when comes to the algorithm implementation stage.

3.2.3 Bubble Updating

According to the initial version of the CES algorithm mentioned in [8], the bubble set $\mathcal{B}$, which serves as the domain of convex optimization problem, is fixed throughout the whole optimization procedure. However, during our early-stage investigation, the fixed $\mathcal{B}$ limits the optimization performance, since the nodes on the optimized trajectory may deviate over each iteration, which leads to a more optimal solution. However, this behavior may be constraint by the outdated solution domain $\mathcal{B}$ given by the initial reference trajectory $\mathcal{P}$. Since $\mathcal{B}$ comes from the idea of marking the feasible neighborhood region around $\mathcal{Q}$, or more precisely, the convex feasible domain for the position optimization of each node Q_i, which is constrained by obstacles, a new bubble updating mechanism comes to our mind that updates the bubble set $\mathcal{B}$ by changing the bubble radius with respect to the obstacles after each iteration. For those nodes that locate too close to the obstacles, the bubble updating mechanism updates the position of $_i$ and the corresponding bubble center position A_i to maintain a tolerable distance $\bar{r}_l$ from its closest obstacle. According the convex optimization theory, if we adjust the feasible domain when maintaining the domain convexity, we will be able to search for the optimal solution over a larger domain and it's likely to find more optimal solution than using the fixed domain.

3.3 Velocity Optimization

In this section, we leverage the Minimum-Time Speed Optimization Subroutines (MTSOS) method proposed by [10] to address the problem of finding the control inputs that allow a vehicle to traverse a specified trajectory found by Elastic Band Stretching in minimum time. To formulate this problem, we denote $q \in \mathbf{R}^p$ as the generalized position or configuration vector of the object and its time derivative is denoted by $\dot{q} \in \mathbf{R}^p$ where p is the degree of freedom in the configuration. Let T be the time at which the vehicle reaches the end of the path. In this case, the optimization problem over the traversal time T can be formulated as

$$\begin{aligned}
\text{minimize} \quad & T && \text{(3a)}\\
\text{subject to} \quad & R(q(t))u(t) = M(q(t))\ddot{q}(t) + C(q(t),\dot{q}(t))\dot{q}(t) + d(q(t)), \quad t \in [0,T], && \text{(3b)}\\
& s(\theta(t)) = q(t), \quad t \in [0,T], && \text{(3c)}\\
& (\dot{q}(t)^2, \ddot{q}(t), u(t)) \in \mathcal{C}(q(t)), \quad t \in [0,T] && \text{(3d)}
\end{aligned}$$

Equality constraint (3b) corresponds to the dynamics model of the object that presents in a second-order ODE. $R : \mathbf{R}^p \to \mathbf{R}^{p\times r}$ is the control matrix, $M : \mathbf{R}^p \to \mathbf{S}^p_{++}$ is the mass matrix, $\mathbf{S}^p_{++}$ is the set of symmetric positive definite matrices in $\mathbf{R}^{p\times p}$, $C : \mathbf{R}^{2p} \to \mathbf{R}^{p\times p}$ is the centrifugal matrix and is linear in $\dot{q}$, and $d : \mathbf{R}^p \to \mathcal{R}^p$ is the force dependent on the configuration. In the friction circle car model we are using in this project, we have

$$M = mI, C = 0, d = 0, R = \begin{bmatrix} \cos\phi(\dot{q}) & -\sin\phi(\dot{q}) \\ \sin\phi(\dot{q}) & \cos\phi(\dot{q}) \end{bmatrix}.$$

Equality constraint (3c) constrains the trajectory to be traversed, which is defined as $s : [0,1] \to \mathbf{R}^p$ in the generalized position space. We say a vehicle moves along trajectory s when (3c) holds. In (3c), $\theta : [0,T] \to [0,1]$ satisfies $\theta(0) = 0$, $\theta(T) = 1$, and $\dot{\theta} \geq 0$. Hence, the function θ provides the velocity of

the vehicle along the trajectory. Lastly, equality constraint (3d) is about control input. $\mathcal{C}(q) \subseteq \mathbf{R}^{p\times p\times r}$ is a set valued mapping to convex sets. Given by the friction circle constraint, the constraint set $\mathcal{C}$ for $(\dot{q}^2, \ddot{q}, u)$ is about velocity limits in certain region in the workspace, which is

$$\mathcal{C} = \{(\dot{q}^2, \ddot{q}, u) | ||u||_2 \leq \mu mg, u^{long} \leq W_f \mu mg\}.$$

We notice that the above problem is not necessarily convex. In the following, we apply the change of variables to reformulate Problem (3) as a convex problem. First of all, we notice the the following equations holds by definition.

$$\dot{q}(t) = s'(\theta(t))\dot{\theta}(t),$$
$$\ddot{q}(t) = s'(\theta(t))\ddot{\theta}(t) + s''(\theta(t))\dot{\theta}(t)^2.$$

Then, we apply the equations to (3b) which yields

$$\tilde{R}(\theta)u = \tilde{m}(\theta) + \tilde{c}(\theta)\dot{\theta}^2 + \tilde{d}(\theta).$$

where

$$\tilde{R} = R(s(\theta)),$$
$$\tilde{m}(\theta) = M(s(\theta))s'(\theta),$$
$$\tilde{c}(\theta) = M(s(\theta))s''(\theta) + C(s(\theta))s'^2(\theta),$$
$$\tilde{d}(\theta) = d(s(\theta)).$$

Then, we define two quantities.

$$a(\theta) = \ddot{\theta}, \quad b(\theta) = \dot{\theta}^2.$$

We notice that the two quantities are related in the following way.

$$b'(\theta) = 2a(\theta).$$

Now, we can reformulate the objective reported in (3a) as

$$T = \int_0^T 1dt = \int_{\theta(0)}^{\theta(T)} \dot{\theta}^{-1} d\theta = \int_0^T \dot{\theta}^{-1} d\theta = \int_0^1 b(\theta)^{-1/2} d\theta.$$

Combining them together, the optimization problem can be reformulated as

$$\begin{aligned}
\text{minimize} \quad & \int_0^1 b(\theta)^{-1/2} d\theta && \text{(4a)}\\
\text{subject to} \quad & \tilde{R}(\theta)u = \tilde{m}(\theta) + \tilde{c}(\theta)\dot{\theta}^2 + \tilde{d}(\theta), \quad \theta \in [0,1], && \text{(4b)}\\
& b'(\theta) = 2a(\theta), \quad \theta \in [0,1], && \text{(4c)}\\
& (a(\theta), b(\theta), u(\theta)) \in \tilde{\mathcal{C}}_\theta, \quad \theta \in [0,1] && \text{(4d)}
\end{aligned}$$

where

$$\tilde{\mathcal{C}}_\theta = \left\{(a(\theta), b(\theta), u(\theta)) \mid (s'(\theta)^2 b(\theta), s'(\theta)a(\theta) + s''(\theta)b(\theta), u(\theta)) \in C(s(\theta))\right\}$$

We notice that (4a) is the integral of a convex function of b. Since the integral of a convex function is convex, we can conclude that the objective function is convex. The dynamics constraint (4b) is affine in a, b, and u and is therefore convex. Equality constraint (4c) is convex as the derivative is a linear operator. Finally, $\tilde{\mathcal{C}}_\theta$ is an affine transformation of a convex set and is therefore convex. Then, we can conclude that the problem reported in (4) is convex which can be solved using the methods of discretization to address the differential

constraints and continuous objective function. After the discretization, the optimization problem can be re-written in the form presented in Problem (5). More details about velocity optimization with MTSOS could be found in [10]. Note the initial velocity optimization problem is now formulated as a finite-dimensional convex optimization problem. The practical utilization of this optimization method, however, requires the usage of fourth-order Runge–Kutta (RK4), which is a widely-used iterative method used to approximate ODE solutions, to give a numerical solution of the ODE that contributes to the new velocity distribution throughout the optimized trajectory $\mathcal{Q}$.

$$\text{minimize} \quad 2\sum_{i=1}^{m-1}\left(\frac{\theta_{i+1}-\theta_i}{b_i^{1/2}+b_{i+1}^{1/2}}\right) \tag{5a}$$

$$\text{subject to} \quad \tilde{R}(\bar{\theta}_i)u_i = \tilde{m}(\bar{\theta}_i)+\tilde{c}(\bar{\theta}_i)\frac{b_{i-1}+b_i}{2}+\tilde{d}(\bar{\theta}_i), \quad \bar{\theta}_i=\frac{\theta_{i-1}+\theta_i}{2}, i=1,...,n, \tag{5b}$$

$$b_0 = \left(\frac{||v_{init}||_2 d\theta}{||s(\theta_1)-s(\theta_0)||_2}\right)^2, \tag{5c}$$

$$b_i - b_{i-1} = 2a_i d\theta, \quad i=1,...,n, \tag{5d}$$

$$(a_i,b_i,u_i)\in\tilde{\mathcal{C}}_{\theta_i}, \quad i=1,...,n \tag{5e}$$

3.4 Artificial Potential Field

Artificial Potential Field (APF) is one of the most fundamental and traditional algorithms used to solve the problem of trajectory planning and obstacle detection. This algorithm assigns an artificial potential value to every point throughout the workspace to encourage the robot to locate within a expected region. In this project, the intention for artificial potential field design is to help to minimize the vehicle's traversal time along the trajectory when maintaining an appropriate distance from obstacles. With this, attractive and repulsive potential fields are assigned to the goal and obstacles respectively. Considering the disciplined convex programming rule set (DCP ruleset) in CVX [13] and MOSEK [14], while both of them are convex programming solvers we are using in this project, the potential field function must have an overall convex profile, otherwise, they could not be solved easily, which indicates that artificial potential field design may not follow our original idea that leads to its ideal performance in this problem. After intensive attempts, we propose the attractive potential field at Q_i with respect to the goal P_n as

$$\psi_{attr}(Q_i,P_n) = ||Q_i-P_n||^2$$

The repulsive potential field at Q_i with respect to the center of obstacle O_j as

$$\psi_{rep}(Q_i,O_j) = -||Q_i-O_j||_1^1$$

The total potential energy at Q_i is

$$\psi(Q_i) = \alpha\psi_{attr}(Q_i,P_n)+\beta\sum_{j=1}^{m}\psi_{rep}(Q_i,O_j) = \alpha||Q_i-P_n||^2-\beta\sum_{j=1}^{m}|||Q_i-O_j||_1^1$$

where α and β are weight coefficients over the goal potential field and obstacle potential field.

According to the potential field functions we proposed, the total potential energy function for Q_i $\psi(Q_i)$ is non-convex, due to the concave term $\psi_{rep}(Q_i,O_j)$, so that the trajectory optimization over the artificial potential field may not be able to solve by $\min_{\mathcal{Q}}\sum_{i=1}^{n}\psi(Q_i)$ directly in an independent optimization step, which may cause the failure of convex programming solvers. Besides, the attempts for using some other decreasing potential field functions for obstacles, like $-log(||Q_i,O_j||)$ or $exp(-||Q_i,O_j||)$ also fails due

to the computation complexity and the constraints of DCP ruleset. In this case, we need to combine this optimization task with some other optimization problems, which introduce the problem about the choice of trade-off coefficient between two objective functions. In our convex trajectory optimization problem, we are trying to incorporate the summation of potential value over each trajectory nodes Q_i with the objective function in Problem (2). Since the potential energy on each Q_i directly relates to the l_1 and l_2 norm of the difference between two points, which is large after potential energy summation, the trade-off coefficient needs to be a small value, like those in terms of 10^{-4} that we used in the experiments later.

The modified optimization problem is presented in Problem (6), while γ is the trade-off coefficient between two objective functions for elastic band stretching, α is the relative weight over the goal potential field, after setting β as 1.

$$\underset{Q_2,...,Q_{n-1}}{\text{minimize}} \quad \sum_{k=2}^{n-1} ||2Q_k - Q_{k-1} - Q_{k+1}||^2 + \gamma \sum_{i=1}^{n} \psi(Q_k) \tag{6a}$$

$$\text{subject to} \quad Q_1 = P_1, \quad Q_2 = P_1 + d \cdot \frac{v_1}{||v_1||}, \tag{6b}$$

$$Q_n = P_n, \quad Q_{n-1} = P_1 - d \cdot \frac{v_{n-1}}{||v_{n-1}||}, \tag{6c}$$

$$Q_k \in \mathcal{B}_k = \{x \in \mathbf{R}^2 | ||x - A_k||_2 \leq r_k\}, \quad k = 3, ..., n-2, \tag{6d}$$

$$||2Q_k - Q_{k-1} - Q_{k+1}|| \leq \min \left\{ \alpha_k \left(\frac{d}{||v_k||} \right)^2, \frac{d^2}{R_{min}} \right\}, \quad k = 2, ..., n-1, \tag{6e}$$

$$\psi(Q_k) = \alpha ||Q_k - P_n||^2 - \sum_{j=1}^{m} |||Q_k - O_j||_1^1, \quad k = 1, ..., n \tag{6f}$$

3.5 Overall Algorithm

The overall framework of the algorithm we proposed for the experiment follows an iterative procedure, presented in Algorithm 2. The algorithm reads the reference trajectory $\mathcal{P}$, the obstacle set $\mathcal{O}$, initial bubble set $\mathcal{B}^{[0]}$ that corresponds to the reference trajectory $\mathcal{P}$ from the file, and loads necessary hyper-parameters for each function, including vehicle hyper-parameters $\mathcal{H} = (m, \mu, W_f, R_{min})$, trade-off coefficients for APF α and γ, optimization iteration number $iter$. Here, $\mathcal{Q}^{[i]}$, $\mathcal{V}^{[i]}$ and $\mathcal{B}^{[i]}$ denote the value of optimized trajectory $\mathcal{Q}$, the velocity distribution $\mathcal{V}$ and bubble set $\mathcal{B}$ at ith iteration, while for $i = 0$ these represent the initial value of corresponding variables.

Algorithm 2 Overall Algorithm Framework

Require:

Reference trajectory $\mathcal{P}$, obstacle set $\mathcal{O}$, initial bubble set $\mathcal{B}^{[0]}$, vehicle hyper-parameters $\mathcal{H} = (m, \mu, W_f, R_{min})$, trade-off coefficients α and γ, optimization iteration number $iter$.

$\mathcal{V}^{[0]}, \mathcal{U}^{long[0]} \leftarrow VelocityOptimizer(\mathcal{P}, \mathcal{H})$
$\mathcal{Q}^{[0]} \leftarrow \mathcal{P}$
for $i = 1 : iter$ **do**
    $\mathcal{Q}^{[i]} \leftarrow ElasticBand(\mathcal{Q}^{[i-1]}, \mathcal{V}^{[i-1]}, \mathcal{U}^{long[i-1]}, \mathcal{B}^{[i-1]}, \mathcal{O}, \mathcal{H}, \alpha, \gamma)$
    $\mathcal{Q}^{[i]}, \mathcal{B}^{[i]} \leftarrow BubbleUpdating(\mathcal{Q}^{[i]}, \mathcal{O})$
    $\mathcal{V}^{[i]}, \mathcal{U}^{[i]} \leftarrow VelocityOptimizer(\mathcal{Q}^{[i]}, \mathcal{H})$
end for

During iteration, the algorithm first generates the initial velocity distribution over the reference trajectory based on the vehicle dynamic model, since only the node position on the reference trajectory is given. After that, a new optimized trajectory $\mathcal{Q}^{[i]}$ is generated through $ElasticBand$, based on Optimization Problem (2) if no artificial potential field is given, otherwise based on Optimization Problem (6). After the generation of a new optimized trajectory, the bubble set is updated by $BubbleUpdating$ with new trajectory $\mathcal{Q}^{[i]}$ and obstacle set $\mathcal{O}$. The velocity distribution $\mathcal{V}^{[i]}$ is updated with $VelocityOptimizer$ given new trajectory $\mathcal{Q}^{[i]}$ and vehicle hyper-parameters $\mathcal{H}$. The iteration is performed by $iter$ times to find out the optimized solution for the trajectory. At the end of each iteration, the value of the average length of the optimized trajectory $\mathcal{Q}$ is updated with

$$d^{[i]} = \frac{\sum_{k=1}^{n-1} ||Q_{k+1}^{[i-1]} - Q_k^{[i-1]}||}{n-1}$$

4 Experiments

In this section, we investigate the performance of the CES method over two major aspects: (a) Optimialty, reflected by the overall length of the optimized trajectories and length reduction percentage compared with the reference trajectory before optimization. This criterion provides a straightforward idea on the optimization performance and the overall trajectory shape; (b) Efficiency, supported by the traversal time from the start to the goal and the reduction percentage compared with the traversal time on reference trajectory. This criterion intends to incorporate the optimized velocity at each node throughout the trajectory with the overall trajectory length, as a compensation for the trajectory length criterion; (c) Time complexity, presented by computation time and its average over the number of trajectory points. Its purpose is to briefly illustrate the time complexity of the CES method over each reference trajectory proposed.

The algorithm evaluation includes five sections of experiments. In Section 4.1, we evaluate the algorithm performance in a simple maze that contains several sparse circular obstacles, named Sparse Circular Maze. This section intends to verify the feasibility of modified CES on each reference trajectory given a simple obstacle avoidance scenario. Section 4.2 includes the experiments made on a more compacted circular maze scenario called Compacted Circular Maze, while obstacles occupy approximately 40% overall map area. The goal is surrounded by a series of obstacles with a narrow, creeping channel that connects to the start. This scenario intends to test the algorithm's performance in a difficult trajectory planning task, which may significantly reveal its superiority over vanilla trajectory planning methods without any optimization. In Section 4.3, a practical obstacle avoidance scenario, Lane Changing, is presented to verify the algorithm's performance in a real-world trajectory planning task. In Section 4.4, we discuss the effect caused by the bubble updating mechanism. By adjusting the bubble shape with respect to the obstacles after each iteration, a more optimal and feasible trajectory may be found. Section 4.5 illustrates the effect made by introducing artificial potential field in trajectory optimization and its trade-off versus the original objective function for elastic band stretching. Due to the limitation made by the non-convex artificial potential field constraints and computation complexity that exists in trajectory planning and optimization in a much larger map like the Lane Changing scenario, discussion about the effectiveness of the artificial potential field is only based on the reference trajectories generated by PRM in Sparse and Compacted Circular Maze scenario. After presenting and analyzing all experiment results, we will discuss the underlying mechanisms that contribute to these phenomena we observed and the effectiveness of the extra steps we introduced to the initial CES algorithm to improve the optimization performance in Section 4.6.

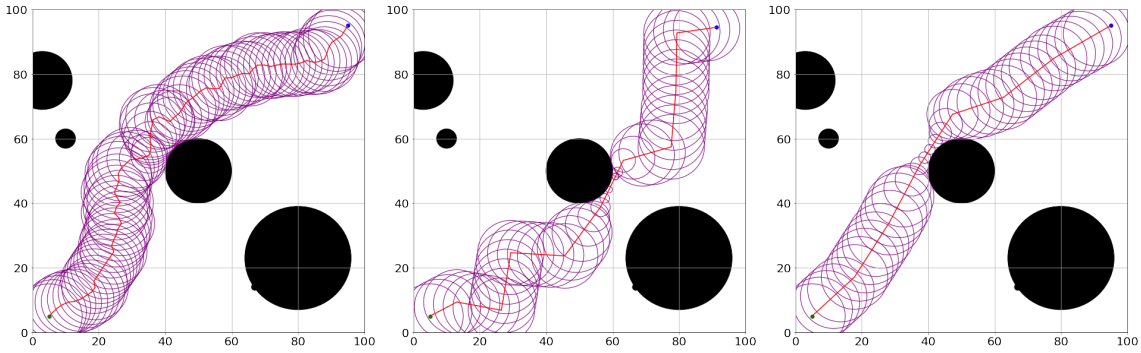
Hyper-parameters for each experiment are set as follows. For the bubble generation and updating part, the upper and lower bound for bubble radius are $r_u = 10m$ and $r_l = 1m$ during the bubble generation stage just after the reference trajectory generation, while bubble radius lower bound in the bubble updating step is $\bar{r}_l = 0.1m$ to ensure that the reference nodes on the trajectory always locate outside of each obstacle. The

up-sampling interval for generating new trajectory reference node is $3m$. The bubble intersection threshold and translation step are all $0.1m$. In the convex optimization section, the hyper-parameters for the vehicle are $m = 1,000kg$, $\mu = 0.3$, $U^{long} = 0.5\mu mg$ and $R_{min} = 0.5m$. The maximum iteration number for MTSOS velocity optimization is 100, while the iteration number for overall convex optimization procedure for shape and velocity optimization is $iter = 30$.

All experiments are performed on a computer with an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz processor and 16GB RAM. Baseline trajectory planning algorithms were implemented manually with Python. The CES algorithm was implemented in MATLAB with CVX [13] convex optimization toolbox for elastic band stretching, MTSOS [10] for velocity optimization, YALMIP convex optimization toolbox [15] and MOSEK convex optimization solver [14] for elastic band stretching with a trade-off between the summation of the artificial potential field throughout the trajectory.

4.1 Sparse Circular Maze

In this scenario, the workspace is a $100m \times 100m$ square with five randomly placed obstacles. The coordinate of start and goal are $(5, 5)$ and $(95, 95)$, respectively. For RRT, the reference trajectory is generated by $1,000$ samples with a step size of $2m$, while the interpolation rate for collision detection is 100. For PRM, the maximum iteration for A* searching is $100,000$, the sample number for generating the reference trajectory is 50, the interpolation rate for collision detection is 100, and the search scope for the neighboring node is 20. For FMT, the iteration number is $10,000$, the sample number for generating the reference trajectory is $1,000$ and the search scope for the neighboring node is 20. The reference trajectories created by each method and the bubble generated for elastic band stretching are shown in Figure 2. In this figure, black solid circles are the obstacles, red lines are the reference trajectory, purple circles are bubbles generated, while the start and goal are marked by green and blue points.



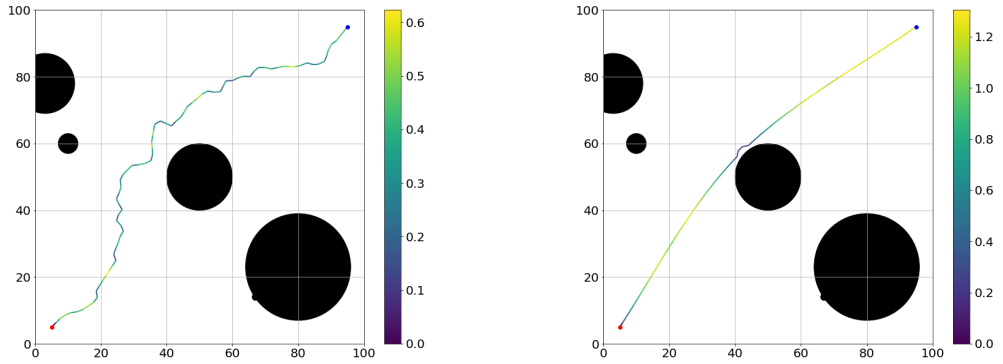
(a) RRT (b) PRM (c) FMT

Figure 2: Reference Trajectory and Bubble Generation under Sparse Circular Maze

After implementing the convex optimization algorithm on each reference trajectory, we could acquire the new trajectory with adjusted velocity distribution. The velocity distribution over the reference trajectory and the optimized trajectory made by RRT is presented in Figure 3, while the full result for all reference trajectories is included in Figure 8 in Appendix. With this, we could compute the overall length, transit time, and computation time for each reference trajectory. Length variation for reference and optimized trajectories under this scenario is presented in Table 1, while the result on traversal time and computation time are presented in Table 2 and Table 3, respectively.

According to the result, the CES optimization method reveals its ability in shortening the trajectory length generated by RRT and PRM and is splendid in shrinking the traversal time for both trajectories. The

reason that contributes to this phenomenon is that sharp turns and deceleration due to the obstacles are reduced by convex optimization, which verifies its feasibility. However, the length for the FMT trajectory inflates after optimization, while the reduction in its traversal time is not as remarkable compared with the other two methods. A potential explanation for this arises from the evading behavior around the central obstacle, which decelerates the vehicle by forcing it to take a sub-optimal trajectory. The average computation time for RRT is the lowest among all methods, while FMT and PRM take more time on average. This may be caused by the overall distance between the trajectory and the obstacles, since implementing planning and optimization with respect to the obstacle is generally more computation expensive.



(a) RRT Reference (b) RRT Optimized

Figure 3: Velocity distribution over Reference and Optimized Trajectories under Sparse Circular Maze

	Reference Length	Optimized Length	Reduction (%)
RRT	149.147	130.026	12.820
PRM	154.118	129.676	15.860
FMT	126.676	129.544	-2.264

Table 1: Length Variation for Reference and Optimized Trajectories under Sparse Circular Maze

	Reference Traversal Time	Optimized Traversal Time	Reduction (%)
RRT	443.789	163.682	63.228
PRM	319.701	161.799	49.390
FMT	176.840	155.128	12.278

Table 2: Traversal Time for Reference and Optimized Trajectories under Sparse Circular Maze

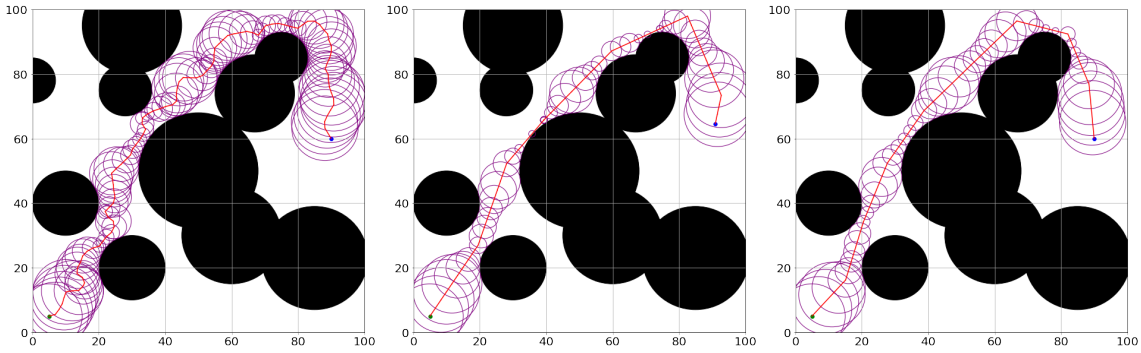
	Node Number	Computation Time	Average Time
RRT	80	43.789	0.547
PRM	49	34.046	0.695
FMT	40	31.664	0.792

Table 3: Computation Time for Reference and Optimized Trajectories under Sparse Circular Maze

4.2 Compacted Circular Maze

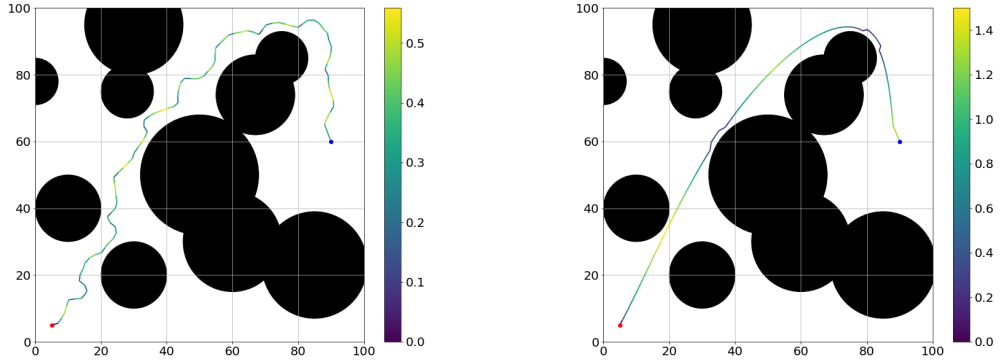
In Compacted Circular Maze, another workspace with a $100m \times 100m$ square and ten compactly-placed obstacles is set. The coordinate of start and goal are $(5, 5)$ and $(90, 60)$, respectively. The hyperparameters for RRT and FMT are the same as those in Section 4.1, while for PRM, the maximum iteration for A* searching is $1,000,000$ and the search scope for the neighboring node is 30. The reference trajectories created by each method and the bubble generated for elastic band stretching are shown in Figure 4.

The velocity distribution over the reference and optimized trajectory made by RRT under this scenario is presented in Figure 5, while the full result is included in Figure 9 in Appendix. With this, we could compute the overall length, transit time, and computation time for each reference trajectory. Length variation for reference and optimized trajectories under this scenario is presented in Table 4, while the result on traversal time and computation time are presented in Table 5 and Table 6, respectively.



(a) RRT (b) PRM (c) FMT

Figure 4: Reference Trajectory and Bubble Generation under Compacted Circular Maze



(a) RRT Reference (b) RRT Optimized

Figure 5: Velocity distribution over Reference and Optimized Trajectories under Compacted Circular Maze

According to the result, the trajectory optimization method contributes to the notable reduction traversal time for RRT compared with the other two methods, though the variation in trajectory length for all methods is trivial. Due to the constraint made by obstacles, the domain for convex optimization is limited, so that the feasible trajectory in this scenario, if exists, could be taken as near-optimal, especially for those line-dominant trajectories made by PRM and FMT. For RRT, since the reference trajectory is high sub-optimal,

though dynamically feasible, the improvement made by optimization is still remarkable. As for the time complexity, the average computation time for RRT is still the lowest among all methods, while the average time for FMT and PRM is similar.

	Reference Length	Optimized Length	Reduction (%)
RRT	185.778	155.808	16.132
PRM	160.972	153.450	4.673
FMT	157.132	154.659	1.574

Table 4: Length Variation for Reference and Optimized Trajectories under Compacted Circular Maze

	Reference Traversal Time	Optimized Traversal Time	Reduction (%)
RRT	537.435	248.666	53.731
PRM	292.317	223.323	23.603
FMT	232.787	218.815	6.002

Table 5: Traversal Time for Reference and Optimized Trajectories under Compacted Circular Maze

	Node Number	Computation Time	Average Time
RRT	98	56.069	0.572
PRM	50	34.279	0.685
FMT	50	34.497	0.690

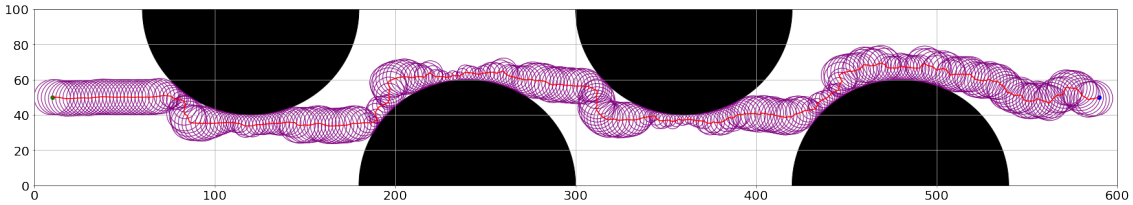
Table 6: Computation Time for Reference and Optimized Trajectories under Compacted Circular Maze

4.3 Lane Changing

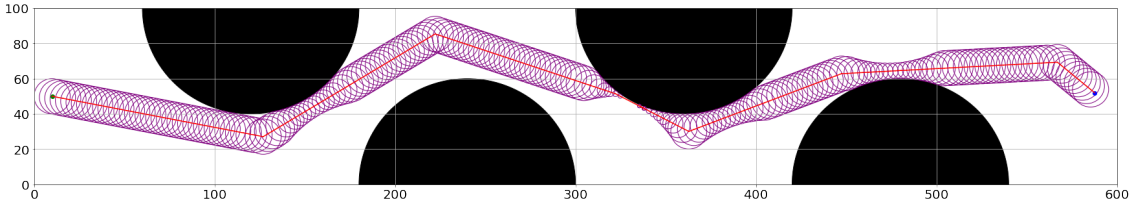
Lane Changing has a $600m \times 100m$ workspace with four uniformly-disturbed obstacles on both sides of the map. The coordinate of start and goal are $(10, 50)$ and $(390, 50)$. The hyperparameters for RRT and FMT are the same as those in Section 4.1 and 4.2, while PRM is simplified due to the exponential growth in computation complexity under an enlarged scenario, the sample number is 30, the maximum iteration for A* searching is $500,000$ and the search scope for the neighboring node is 120. The reference trajectories created by each method and the bubble generated for elastic band stretching are shown in Figure 6.

The velocity distribution over the reference and optimized trajectory made by RRT under this scenario is presented in Figure 7, while the full result is included in Figure 10 in Appendix. With this, we could compute the overall length, transit time, and computation time for each reference trajectory. Length variation for reference and optimized trajectories under this scenario is presented in Table 7, while the result on traversal time and computation time are presented in Table 8 and Table 9.

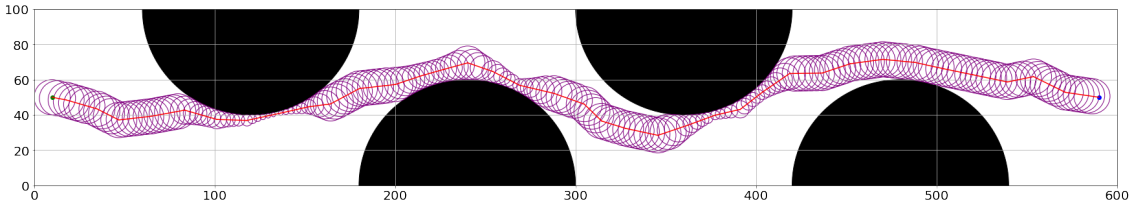
Under a more uniform, realistic scenario, the ability of trajectory optimization in trajectory length and traversal time reduction is less notable than the previous two scenarios, though both trajectory length and traversal time are still reduced by roughly $5\ 10\%$. Though still sub-optimal, reference trajectories under this scenario resemble the possible optimal trajectory, which is a line-like curvature with mild turning. In this case, the space for optimization is limited. The average computation time increases heavily for all methods, especially for RRT, since the computation complexity for convex optimization solver is exponential to the number of variables, which increases by several times compared with the other two scenarios.



(a) RRT



(b) PRM



(c) FMT

Figure 6: Reference Trajectory and Bubble Generation under Lane Changing

	Reference Length	Optimized Length	Reduction (%)
RRT	671.721	671.721	12.438
PRM	625.756	587.949	6.042
FMT	608.429	586.087	3.672

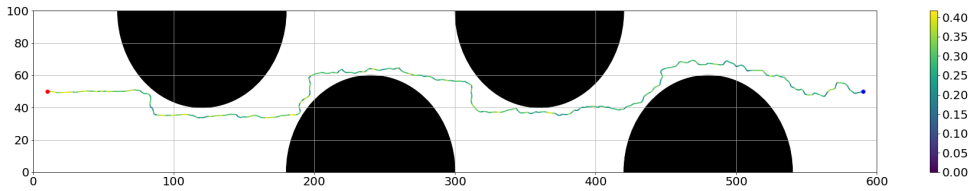
Table 7: Length Variation for Reference and Optimized Trajectories under Lane Changing

	Reference Traversal Time	Optimized Traversal Time	Reduction (%)
RRT	2371.093	1892.362	20.190
PRM	1952.763	1870.957	4.189
FMT	1982.794	1833.086	7.550

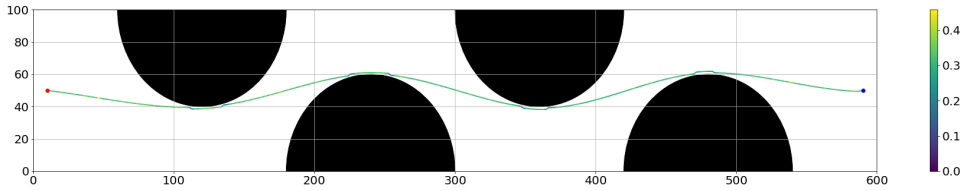
Table 8: Traversal Time for Reference and Optimized Trajectories under Lane Changing

	Node Number	Computation Time	Average Time
RRT	347	758.42	2.186
PRM	205	350.983	1.712
FMT	187	333.201	1.782

Table 9: Computation Time for Reference and Optimized Trajectories under Lane Changing



(a) RRT Reference



(b) RRT Optimized

Figure 7: Velocity distribution over Reference and Optimized Trajectories under Lane Changing

4.4 Bubble Updating

The initial CES algorithm uses a fixed domain for optimization problems marked by bubble generation, which may limit its ability in trajectory optimization. As mentioned in Section 3.2.1, we proposed the bubble updating mechanism during the convex optimization, which intends to adjust the convex optimization domain after each iteration that contributes to finding more optimal trajectories. Its feasibility has been verified in experiments from Section 4.1 to Section 4.3. Based on previous results, since the convex optimization method improves the performance of RRT reference trajectory mostly, the discussion on the effect of bubble changing mechanism will be based on the result on RRT reference trajectory under each scenario. Optimized Trajectories with and without bubble updating mechanism Evaluation metrics in this section only include trajectory length and traversal time, which are notable in RRT optimization.

	Reference	Non-updating	Reduction (%)	Updating	Reduction (%)
Sparse Circular Maze	149.147	130.026	12.820	129.387	13.249
Compacted Circular Maze	185.778	155.808	16.132	671.721	12.438
Lane Changing	671.721	588.171	12.438	596.627	11.179

Table 10: Length Variation for RRT Reference and Optimized Trajectories under Each Scenario

	Reference	Non-updating	Reduction (%)	Updating	Reduction (%)
Sparse Circular Maze	445.125	163.681	63.228	154.387	65.316
Compacted Circular Maze	537.435	248.666	53.731	159.113	70.394
Lane Changing	2371.093	1892.362	20.190	1854.184	21.800

Table 11: Traversal Time for RRT Reference and Optimized Trajectories under Each Scenario

The velocity distribution throughout optimized trajectories made by RRT with and without bubble updating mechanism under each scenario is presented in Figure 11, while the length variation and traversal time over both trajectories are presented in Table 10 and Table 11. During the optimization, the value of the objective function for elastic band equipped with bubble updating mechanism converges greatly, compared with the objective function for the non-bubble-updating case. As for the optimization result reflected by

trajectory length and traversal time, though the variation between the two is trivial, the traversal Time after implementing the bubble updating mechanism is reduced by over 2%, while the trajectory for Compacted Circular Maze is improved by about 17%, which is notable. Besides, the overall shape for the trajectory with bubble updating is smoother than the non-bubble-updating case, which indicates that trajectories generated through this are more dynamically feasible. With this, implementing the bubble updating mechanism helps to improve the overall trajectory optimization quality.

4.5 Artificial Potential Field

Incorporating artificial potential field with trajectory optimization helps to find more optimal and feasible trajectories, while the trade-off coefficient between the original objective function and artificial potential field summation needs to investigate. In this section, the artificial potential field over the map follows the rule set in Section 3.4. Trade-off coefficients used in the experiment are 5×10^{-5}, 1×10^{-4}, and 2×10^{-4}, with the vanilla CES optimization algorithm with the bubble-updating mechanism. The weight coefficient over the goal potential field and obstacle potential field is $\alpha = 0.5$. Considering the time-complexity of implementing convex optimization for each trajectory, experiments on the artificial potential field are only based on PRM reference trajectories under Sparse and Compacted Circular Maze. Besides, due to the existence of non-convex and the combination multi-class convex constraints caused by the artificial potential field, the optimization toolkit and solver uses YALMIP [15] and MOSEK [14] that help to find a feasible solution for the modified optimization problem. Evaluation metrics in this section only include traversal time.

Trade-off (γ)	Traversal in Sparse	Reduction (%)	Traversal in Compacted	Reduction (%)
Reference	319.701	0	292.317	0
$\gamma = 0$	161.799	49.390	223.323	23.603
$\gamma = 5 \times 10^{-5}$	167.854	47.496	237.826	18.641
$\gamma = 1 \times 10^{-4}$	168.612	47.259	229.405	21.522
$\gamma = 2 \times 10^{-4}$	146.478	54.183	221.521	24.219

Table 12: Traversal Time for PRM with Different Trade-off Coefficients with Artificial Potential Field under Sparse and Compacted Circular Maze

The velocity distribution over PRM optimized trajectories with artificial potential field trade-off under each scenario is presented in Figure 12 and Figure 13. The traversal time under both scenarios is presented in Table 12. The result shows that incorporating artificial potential field slightly deteriorates the optimization result under both scenarios when $\gamma = 5 \times 10^{-5}$ and $\times 10^{-4}$, while the optimization performance improves when $\gamma = 2 \times 10^{-4}$. According to the velocity distribution, the overall shape and smoothness for the trajectories with $\gamma = 2 \times 10^{-4}$ are better than the other two cases. The collision avoidance performance for the case $\gamma = 2 \times 10^{-4}$ is much better than the reference vanilla case since the vehicle does not have to decelerate when approaching the central obstacle and maintains an appropriate distance to obstacles. On the other hand, the unsatisfying performance in traversal time may be caused by the deceleration behavior in the neighborhood region of the goal, indicating the drawback of the potential field design. This observation enlightens a potential direction for further research in this field.

4.6 Discussion

Overall, the experiments above show several major trends. (a) The CES algorithm shows its feasibility in smoothing highly sub-optimal and dynamically infeasible trajectories, which is verified by the significant reduction in trajectory length and traversal time, while the general optimization performance depends on the

property of the reference trajectory and the working scenario. (b) The optimization performance is more obvious in RRT trajectories than PRM and FTM trajectories in general since RRT trajectories are usually more sub-optimal in their shape and traversal time. (c) Computation complexity for this optimization algorithm is exponential to node number on the reference trajectory so that prior simplification on reducing the reference node number throughout the reference trajectory helps to improve the computation performance in practical scenarios. (d) Updating the problem domain that is presented by bubbles after each iteration helps to improve the optimality of the final trajectory. (e) Artificial potential field improves the trajectory optimization performance at some level, though its performance highly depends on the choice of trade-off coefficients and potential field function design.

5 Conclusion

In this project, we investigated the convex optimization algorithm on trajectory planning named Convex Elastic Smoothing (CES) and implemented this algorithm on trajectory optimization based on several common trajectory planning methods with modification. The result shows that this convex trajectory optimization algorithm reveals its ability in improving the optimality and efficiency in multiple scenarios, supported by the metrics in trajectory length and traversal time. The modification in the bubble updating mechanism helps to further improve the algorithm performance. The possibility of incorporating the artificial potential field in trajectory optimization is also included and investigated with several experiments, though the result still now is not satisfying. The investigation in this topic leaves the space for future extension.

Further research in this field includes the modification of the artificial potential field design and its trade-off optimization problem with the elastic hand stretching to further improve the trajectory optimality. Besides, applying this optimization framework to more realistic and complicated vehicle models and environments, like the multi-wheel vehicle model and the aircraft model, is another noteworthy topic for future exploration.

References

[1] S. M. La Valle, "Motion planning," *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 108–118, 2011.

[2] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[4] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[5] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[6] 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.

[7] F. Islam, J. Nasir, U. Malik, Y. Ayaz, and O. Hasan, "Rrt-smart: Rapid convergence implementation of rrt towards optimal solution," in *2012 IEEE international conference on mechatronics and automation*, pp. 1651–1656, IEEE, 2012.

[8] Z. Zhu, E. Schmerling, and M. Pavone, "A convex optimization approach to smooth trajectories for motion planning with car-like robots," in *2015 54th IEEE conference on decision and control (CDC)*, pp. 835–842, IEEE, 2015.

[9] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pp. 802–807, IEEE, 1993.

[10] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, 2014.

[11] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International journal of robotics research*, vol. 34, no. 7, pp. 883–921, 2015.

[12] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*, pp. 396–404, Springer, 1986.

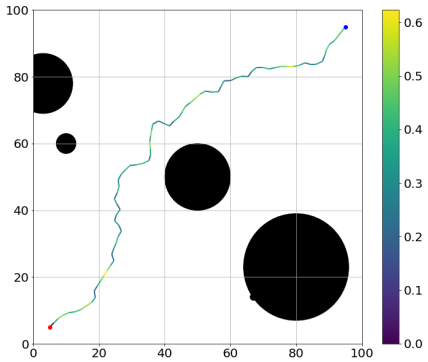
[13] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1." http://cvxr.com/cvx, Mar. 2014.

[14] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019.

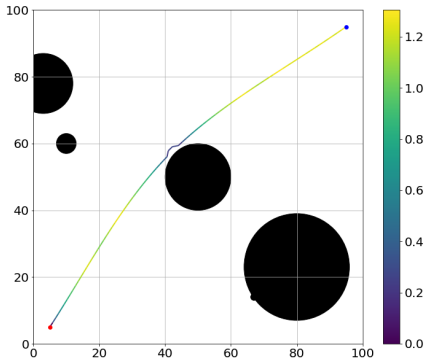
[15] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, (Taipei, Taiwan), 2004.

Appendix

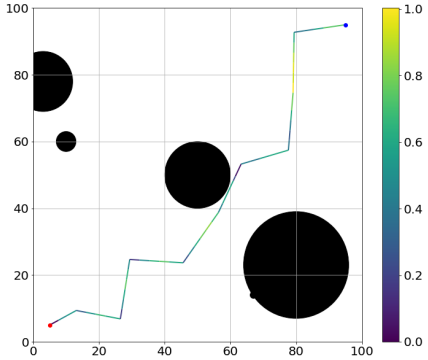
In this section, we will generally present figure that we are not able to cover previously.



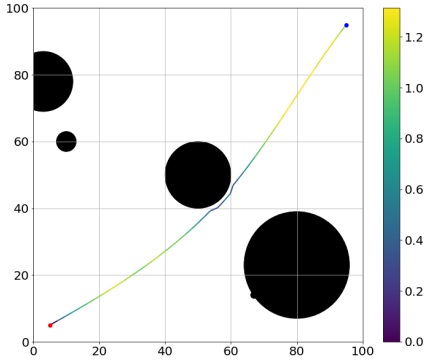
(a) RRT Reference



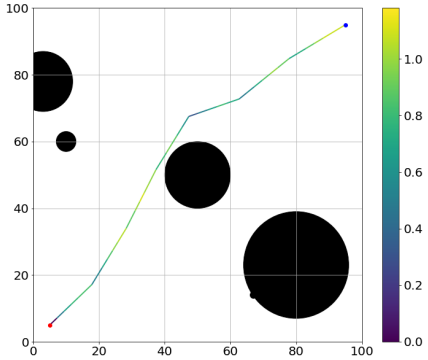
(b) RRT Optimized



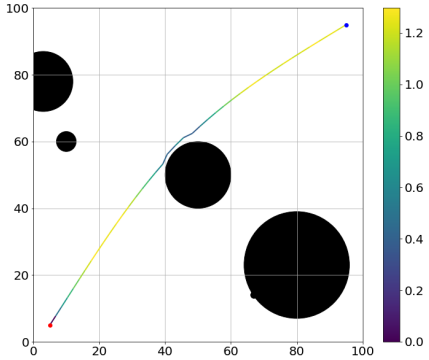
(c) PRM Reference



(d) PRM Optimized

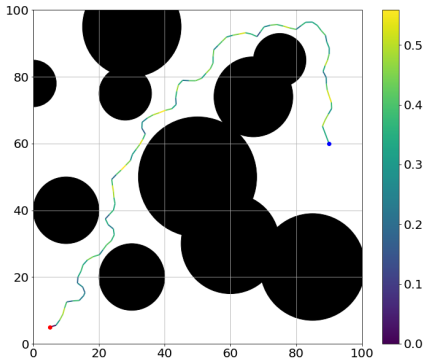


(e) FMT Reference

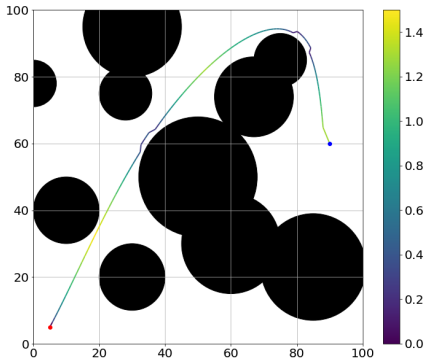


(f) FMT Optimized

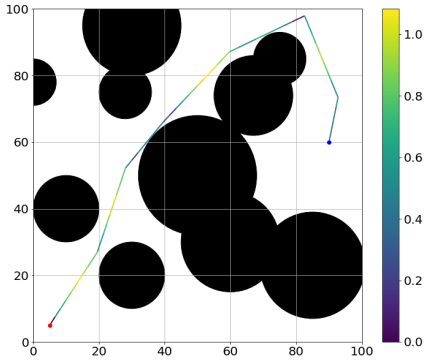
Figure 8: Velocity distribution over Reference and Optimized Trajectories under Sparse Circular Maze



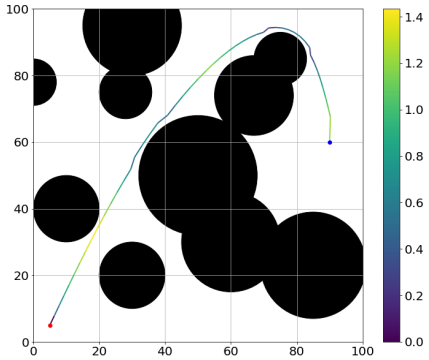
(a) RRT Reference



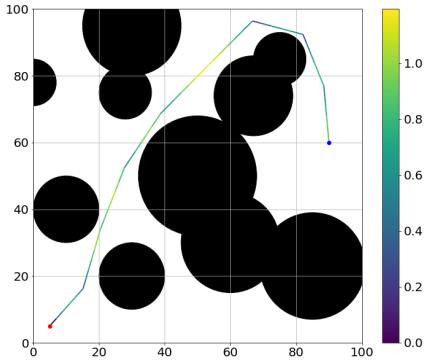
(b) RRT Optimized



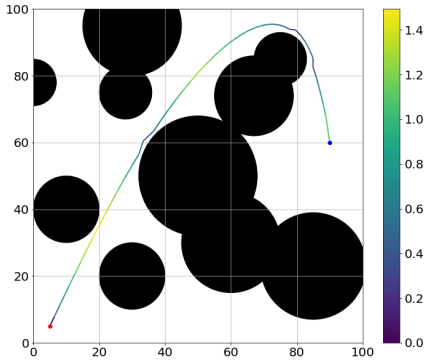
(c) PRM Reference



(d) PRM Optimized

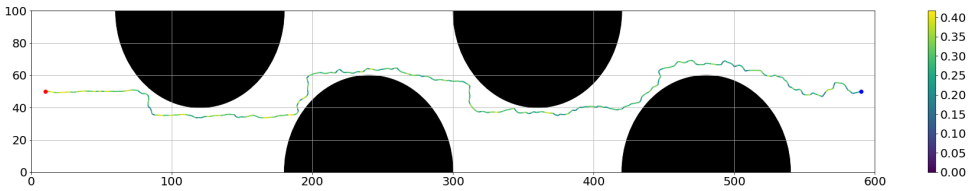


(e) FMT Reference

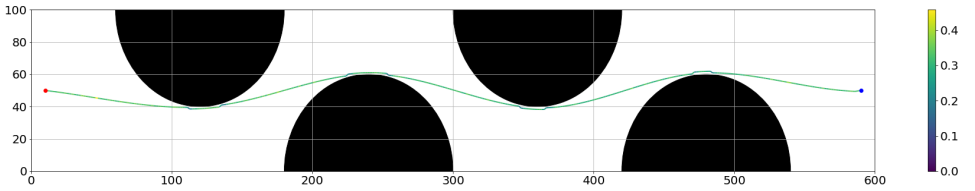


(f) FMT Optimized

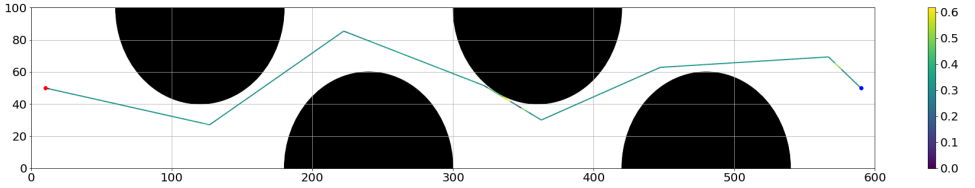
Figure 9: Velocity distribution over Reference and Optimized Trajectories under Compacted Circular Maze



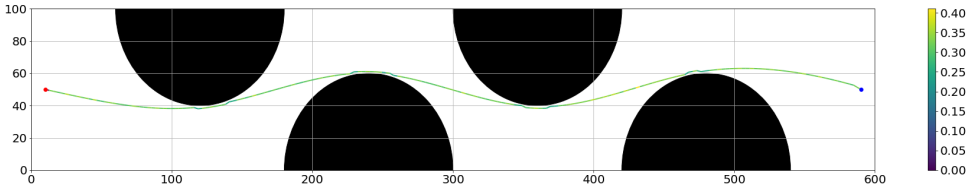
(a) RRT Reference



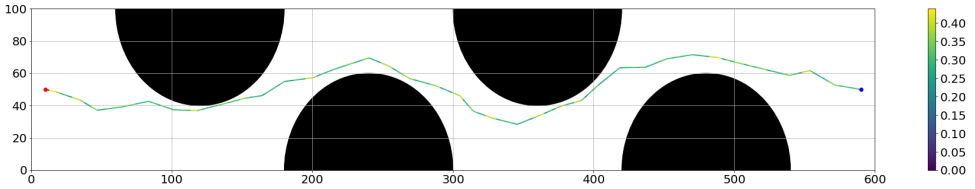
(b) RRT Optimized



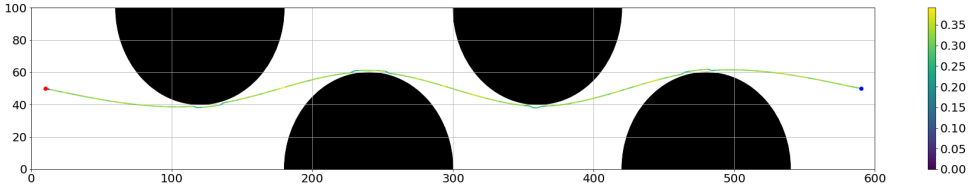
(c) PRM Reference



(d) PRM Optimized

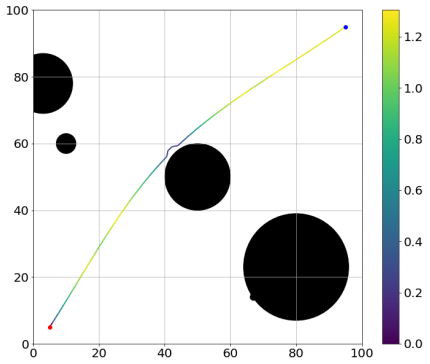


(e) FMT Reference

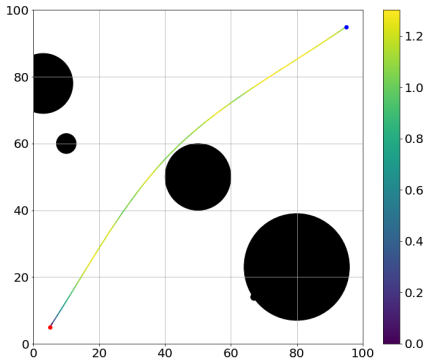


(f) FMT Optimized

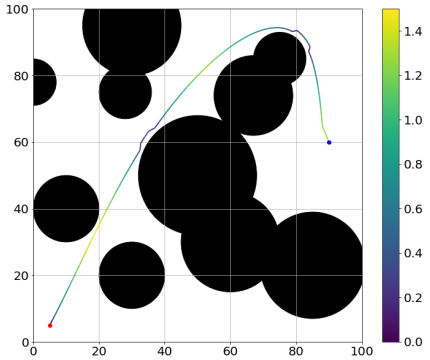
Figure 10: Velocity distribution over Reference and Optimized Trajectories under Lane Changing



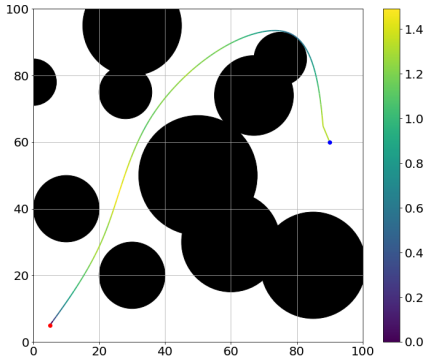
(a) Non-update Sparse



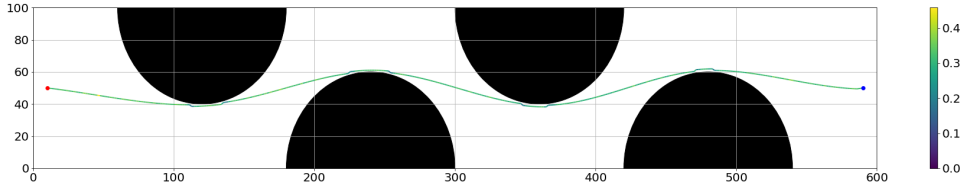
(b) Update Sparse



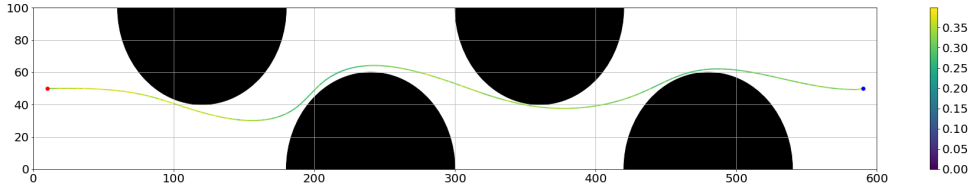
(c) Non-update Compacted



(d) Update Compacted

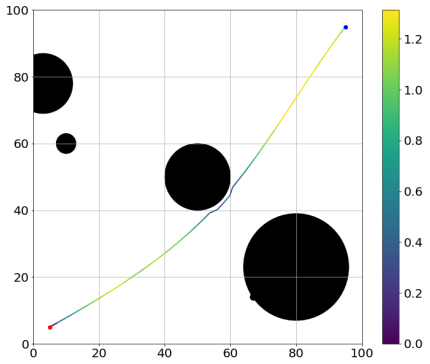


(e) Non-update Lane

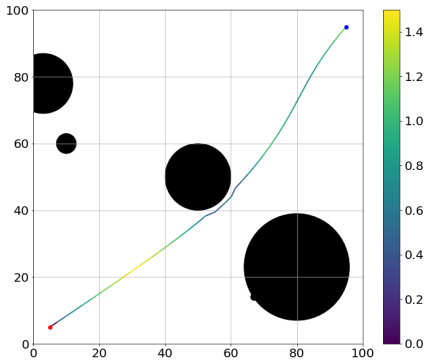


(f) Update Lane

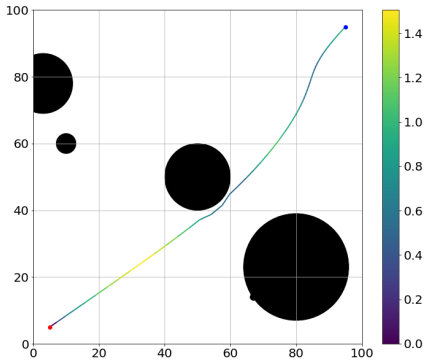
Figure 11: Velocity distribution over RRT Trajectories with and without Bubble Updating Mechanism under Each Scenario



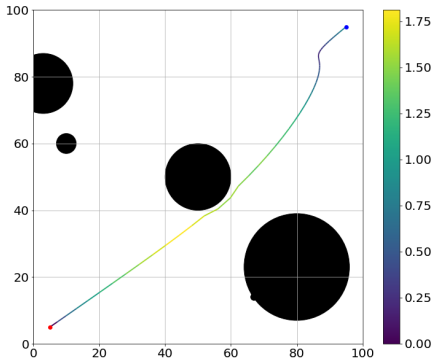
(a) $\gamma = 0$



(b) $\gamma = 5 \times 10^{-5}$

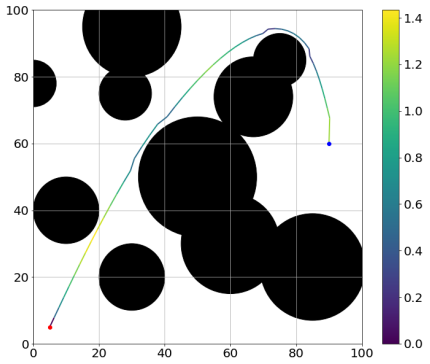


(c) $\gamma = 1 \times 10^{-4}$

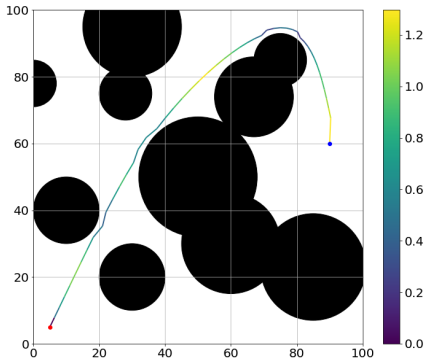


(d) $\gamma = 2 \times 10^{-2}$

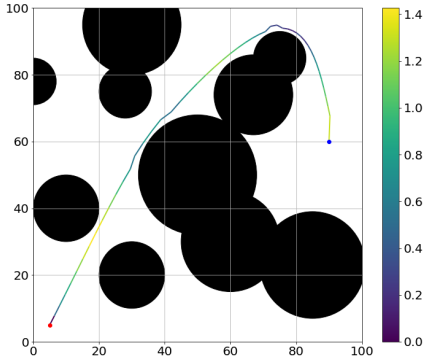
Figure 12: Velocity distribution over PRM Trajectories with Different Trade-off Coefficients with Artificial Potential Field under Sparse Circular Maze



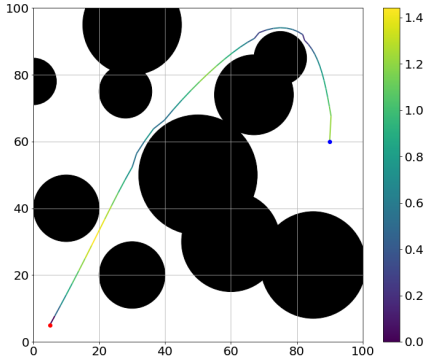
(a) $\gamma = 0$



(b) $\gamma = 5 \times 10^{-5}$



(c) $\gamma = 1 \times 10^{-4}$



(d) $\gamma = 2 \times 10^{-2}$

Figure 13: Velocity distribution over PRM Trajectories with Different Trade-off Coefficients with Artificial Potential Field under Compacted Circular Maze