Path planning with moving obstacles using stochastic optimal control

Seyyed Reza Jafari[1], Anders Hansson[1], Bo Wahlberg[2]

***Abstract*— Navigating a collision-free, optimal path for a robot poses a perpetual challenge, particularly in the presence of moving objects such as humans. In this study, we formulate the problem of finding an optimal path as a stochastic optimal control problem. However, obtaining a solution to this problem is nontrivial. Therefore, we consider a simplified problem, which is more tractable. For this simplified formulation, we are able to solve the corresponding Bellman equation. However, the solution obtained from the simplified problem does not sufficiently address the original problem of interest. To address the full problem, we propose a numerical procedure where we solve an optimization problem at each sampling instant. The solution to the simplified problem is integrated into the online formulation as a final-state penalty. We illustrate the efficiency of the proposed method using a numerical example.**

I. INTRODUCTION

There are numerous applications of autonomous mobile robots that are discussed in the literature. In general, these applications are divided into two groups: indoor applications, such as delivering packages, cargo, and cleaning large buildings [1], and outdoor field robotics application [2]. Finding an optimal path for the robot to reach its destination is a crucial task in these applications. Path planning is one of the primary challenges that must be solved before mobile robots can autonomously navigate and explore complex environments [3]. The primary goal of path planning is to ensure safe, efficient, and collision-free navigation in both static and dynamic environments [4].

Summarizing, the main contributions of this paper are:

1) We formulate an optimal path planning problem as a stochastic optimal control problem.
2) We show that the corresponding value function exhibits symmetry, enabling its representation with a reduced number of variables.
3) We use a numerical example to illustrate that the proposed framework can outperform the widely used $A^\star$ algorithm on the considered class of problems.

II. MATHEMATICAL MODEL

Consider the problem where there is a robot, a stochastically moving obstacle, and a static target in a 2-dimensional space. Let $r_k \in \mathbb{R}^2$ and $h_k \in \mathbb{R}^2$ represent the positions of the robot and the moving obstacle at time k, respectively. Moreover, assume that $t \in \mathbb{R}^2$ is the target position, where $\mathbb{R}$ is the set of real numbers.

The problem is to find an optimal policy that enables the robot to reach the final state, i.e. the target, while trying to avoid colliding with the moving obstacle. The mathematical description of this problem is as follows:

$$
\begin{aligned}
&\text{minimize} \lim_{N\to\infty} \mathbb{E}\left[\sum_{k=0}^{N-1} f(h_k, r_k)\right] \\
&\text{subject to} \quad \begin{cases} r_{k+1} = F_r(r_k, u_k) \triangleq r_k + u_k \\ h_{k+1} = F_h(h_k, w_k) \triangleq h_k + w_k \end{cases}, \quad k \geq 0 \\
&\qquad\qquad u_k \in \mathscr{U}, w_k \in \mathscr{W} \quad k \geq 0
\end{aligned} \tag{1}
$$

where $\mathbb{E}[\cdot]$ denotes mathematical expectation, and where $F_r : \mathbb{R}^2 \times \mathscr{U} \to \mathbb{R}^2$, $F_h : \mathbb{R}^2 \times \mathscr{W} \to \mathbb{R}^2$. Here $\mathscr{U} = \left\{R_u \left[\cos(\alpha) \quad \sin(\alpha)\right]^T \in \mathbb{R}^2 : R_u \in \mathbb{R}_+, \alpha \in [0, 2\pi]\right\}$, and $\mathscr{W} = \left\{R_w \left[\cos(\gamma) \quad \sin(\gamma)\right]^T \in \mathbb{R}^2 : R_w \in \mathbb{R}_+, \gamma \in [0, 2\pi]\right\}$. Where $\mathbb{R}_+$ is the set of nonnegative real numbers. We assume that the probability density function of w only depends on R_w, i.e. it is constant for each value of γ. We let $e = ||r-t||_2$, and $d = ||h-r||_2$. Moreover, $f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_+$ is the so-called incremental cost. We define this function for a given target position t as

$$
\begin{aligned}
f(h,r) &= \begin{cases} 0, & ||r-t||_2 \leq R \\ \lambda(||r-t||_2 - 1)^2 + \frac{1}{||h-r||_2 + \varepsilon}, & ||r-t||_2 > R \end{cases} \\
&= \begin{cases} 0, & e \leq R \\ \lambda(e-1)^2 + \frac{1}{d+\varepsilon}, & e > R \end{cases} \quad \triangleq \bar{f}(d,e)
\end{aligned} \tag{2}
$$

where $\bar{f} : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$, and where $\lambda \in \mathbb{R}_{++}$ is a tuning parameter, and $\varepsilon \in \mathbb{R}_{++}$ is a small number. Here, $\mathbb{R}_{++}$ is the set of positive real numbers. We will solve the infinite horizon optimal control problem (1) using the Bellman equation:

$$
V(h,r) = \min_{u \in \mathscr{U}} \mathbb{E}\left[f(h,r) + V\left(F_h(h,w), F_r(r,u)\right)\right] \tag{3}
$$

If we find a solution V to this equation, the optimal policy is the minimizing argument in the right-hand side of the equation. Thus the optimal policy is a function of (h,r). Note that, the Bellman equation in (3) must be solved for all (h,r) and for a given target position t. This problem leads to the 'curse of dimensionality' due to the large state space. We can use value iterations [5] to solve the Bellman equation in (3):

This work was supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

[1]Seyyed Reza Jafari, and Anders Hansson are with the Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden `seyyed.reza.jafari@liu.se`, `anders.g.hansson@liu.se`

[2]Bo Wahlberg is with the Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden `bo@kth.se`

$$V_{k+1}(h,r) = \min_{u\in\mathscr{U}} \mathbb{E}\left[f(h,r) + V_k\left(F_h(h,w), F_r(r,u)\right)\right] \quad (4)$$

where $V_0(h,r) = 0, \forall h, r \in \mathbb{R}^2$. The limit of this sequence as k goes to infinity satisfies the Bellman equation.

III. Geometric Symmetry

In this section we will show that the value function has a symmetry, and we will use this property to reduce the domain of the value function.

Definition 1: Consider the two pairs, (h_1, r_1) and (h_2, r_2), in Fig. 1. If $d_1 = d_2$, $e_1 = e_2$, and $\theta_1 = \pm\theta_2$, then these pairs are said to be symmetric around t .

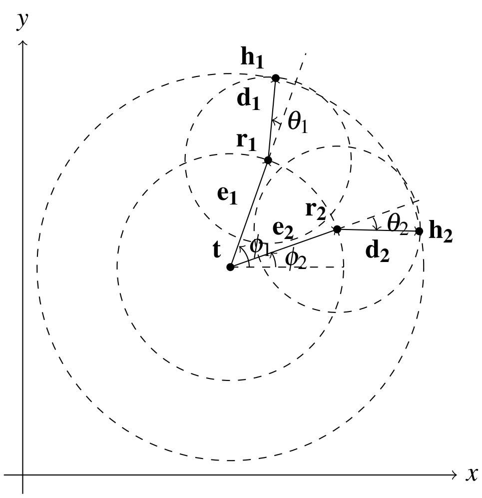

Fig. 1: Geometrical representation of position of robot, and moving obstacle for two symmetric pairs (h_1, r_1), and (h_2, r_2)

We can show that for (4), if (h_1, r_1) and (h_2, r_2) are symmetric, then $V_{k+1}(h_1, r_1) = V_{k+1}(h_2, r_2)$, regardless of the values of ϕ_1 and ϕ_2. We now define

$$W_k(d,e,\theta) = V_k\left(t + e\begin{bmatrix}1\\0\end{bmatrix} + d\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix}, t + e\begin{bmatrix}1\\0\end{bmatrix}\right) \quad (5)$$

where $W_k : \mathbb{R}_+ \times \mathbb{R}_+ \times [0, 2\pi] \to \mathbb{R}$. Then we can equivalently write (4) as

$$W_{k+1}(d,e,\theta) = \min_{u\in\mathscr{U}} \mathbb{E}\left[\bar{f}(d,e) + W_k\left(d_+, e_+, \theta_+\right)\right] \quad (6)$$

where d_+, e_+, and θ_+ are functions of d, e, θ, u, and w. The limit of this recursion will define a solution to (3). This limit is denoted by $W(d,e,\theta)$, where $W : \mathbb{R}_+ \times \mathbb{R}_+ \times [0, 2\pi] \to \mathbb{R}$.

IV. Fitted Value Iteration

We propose to solve the recursion for W_k in (6) using fitted value iteration [5]. Our goal is to approximate the function $W_k(d,e,\theta)$, with $\tilde{W} : \mathbb{R}_+ \times \mathbb{R}_+ \times [0,\pi] \times \mathbb{R}^p \to \mathbb{R}$. One possibility is to use a linear regression model as follows:

$$\tilde{W}(d,e,\theta,a_k) = a_k^T \varphi(d,e,\theta) \quad (7)$$

We consider a piecewise constant approximation of W_k by assuming that the i-th component of φ, $i \in \{1, 2, \cdots, p\}$ is defined as

$$\varphi_i(d,e,\theta) = \begin{cases} 1, & (d,e,\theta) \in \mathscr{S}_i \\ 0, & (d,e,\theta) \notin \mathscr{S}_i \end{cases} \quad (8)$$

where $\mathscr{S}_i$ is a partition of $\mathbb{R}_+ \times \mathbb{R}_+ \times [0,\pi]$.

We also restrict ourselves to the case that $R_u \in \{0, \bar{R}_u\}$, and $R_w \in \{0, \bar{R}_w\}$ to have a tractable problem.

V. Effect of Probability Distribution and Constraints

Now, we relax the condition regarding to probability distribution of w, and also consider constraints on the states. We use the optimal value function V that satisfies the Bellman equation in (3) and formulate the following problem:

$$\begin{aligned} &u_k^\star = \operatorname{argmin}_{u_k} \mathbb{E}\left[f(h_k, r_k) + V(h_{k+1}, r_{k+1})\right] \\ &\text{subject to} \quad \begin{cases} r_{k+1} = F_r(r_k, u_k) \\ h_{k+1} = F_h(h_k, w_k) \end{cases}, \quad k \geq 0 \\ &\qquad u_k \in \mathscr{U}, w_k \in \mathscr{W}, x_k = (r_k, h_k) \in \mathscr{X} \quad k \geq 0 \end{aligned} \quad (9)$$

where $\mathscr{X}$ is the set of all constraints on x_k. The expectation is with respect to w, and the probability distribution of w is arbitrary. At each time step, we solve the optimization problem (9) in real-time to find the control signal $u_k^\star$, and then we apply this control signal.

VI. Numerical Results

We consider a numerical example to show the efficiency of the proposed method. We let $h_0 = \begin{bmatrix}2 & 6\end{bmatrix}^T$, $r_0 = \begin{bmatrix}4 & 12\end{bmatrix}^T$, and $t = \begin{bmatrix}4 & 3\end{bmatrix}^T$. We consider 10000 different realizations of w, and we compare the results of the proposed method with receding horizon $A^\star$. Table I shows the cost, percentage of collisions, and execution time per stage for these methods, averaged over all different realizations of w. The proposed method has a lower cost than receding horizon $A^\star$. Additionally, the average computation time per stage and percentage of collisions for the proposed method are less than those for receding horizon $A^\star$.

TABLE I: Comparison of proposed method with receding horizon $A^\star$

Method	Cost	Computation Time Per Stage [sec]	Percentage of Collision
Proposed Method	3.1168	0.0107	0.01
Receding Horizon $A^\star$	4.1683	0.2183	46.07

References

[1] Haruna, Zaharuddeen, et al. ”Path Planning Algorithms for Mobile Robots: A Survey.” Motion Planning for Dynamic Agents. IntechOpen, 2023.

[2] Mester, Gyula. ”Applications of mobile robots.” Proc. International Conference on Food Science. 2006.

[3] Zhang, Han-ye, Wei-ming Lin, and Ai-xia Chen. ”Path planning for the mobile robot: A review.” Symmetry 10.10 (2018): 450.

[4] Karur, Karthik, et al. ”A survey of path planning algorithms for mobile robots.” Vehicles 3.3 (2021): 448-468.

[5] Hansson, Anders, and Martin Andersen. Optimization for Learning and Control. John Wiley & Sons, 2023.