

# Hybrid Dynamic Moving Obstacle Avoidance Using a Stochastic Reachable Set-Based Potential Field

Nick Malone, *Member, IEEE*, Hao-Tien (Lewis) Chiang, *Student Member, IEEE*, Kendra Lesser, *Member, IEEE*, Meeko Oishi, *Member, IEEE*, and Lydia Tapia, *Senior Member, IEEE*

**Abstract**—One of the primary challenges for autonomous robotics in uncertain and dynamic environments is planning and executing a collision free path. Hybrid dynamic obstacles present an even greater challenge as the obstacles can change dynamics without warning and potentially invalidate paths. Artificial Potential Field (APF) based techniques have shown great promise in successful path planning in highly dynamic environments due to their low cost at runtime. We utilize the APF framework for runtime planning but leverage a formal validation method, Stochastic Reachable (SR) sets, to generate accurate potential fields for moving obstacles. A small number of SR sets are computed a priori, then used to generate a potential field which represents the obstacle’s stochastic motion for online path planning. Our method is novel and scales well with the number of obstacles, maintaining a relatively high probability of reaching the goal without collision, as compared to other traditional Gaussian APF methods. Here, we demonstrate our method with up to 900 hybrid dynamic obstacles and show that it outperforms the traditional Gaussian APF method by up to 60% in the holonomic case and up to 20% in the unicycle case.

## I. INTRODUCTION

Navigation in dynamic, uncertain environments is a difficult yet ubiquitous problem in transportation systems (e.g., autonomous driving, shipping lanes near ports, air traffic management) and distributed robotic systems (vehicle swarms in air, ground, or water environments), with application to problems in search and rescue, coordinated movement, distributed monitoring and surveillance, and others. We consider the problem of motion planning in environments with hundreds of stochastic, dynamic obstacles, in which the obstacles can arbitrarily switch between trajectories that follow a constant radius arc or a straight line, with stochastic angular or translational speeds, respectively. This kind of switching is a realistic abstraction of planar vehicle dynamics, and is commonly used in models of multivehicle conflict resolution [63], [64], [7], [54], [20], aircraft target tracking [30], [8], [47], and in other

This work was supported in part by the National Science Foundation (NSF) under Grant Numbers IIS-1528047, IIS-1553266 (Tapia, CAREER), CMMI-1254990 (Oishi, CAREER) and CNS-1329878. Tapia and Malone are supported in part by the National Institutes of Health (NIH) Grant P2ORR018754 to the Center for Evolutionary and Theoretical Immunology. We thank Nathan Rackley and Steven Cutlip for the modification of ORCA and for the calculation of the hybrid SR set, respectively.

Nick Malone is with Tau Technologies, Albuquerque, NM 87106 (email: nick.malone@tautechnologies.com). Hao-Tien Chiang and Lydia Tapia are with the Computer Science Department of the University of New Mexico, Albuquerque, NM 87131, USA (email: lewispro@unm.edu; tapia@cs.unm.edu). Kendra Lesser is with Verus Research, Albuquerque, NM 87110 (email: kendra.lessner@verusresearch.net). Meeko Oishi is with the Electrical and Computer Engineering Department of the University of New Mexico, Albuquerque, NM 87131, USA (email: oishi@unm.edu). Contributions by Malone and Lesser were made while at the University of New Mexico.

applications [18] which require mathematically tractable yet realistic models of complex behavior.

While motion planning algorithms are highly effective in static environments and can be adapted for moving obstacles with easily predictable trajectories, effective solutions in environments that are both dynamic and stochastic are elusive. Common approaches to solving the motion planning problem for dynamic obstacles include Artificial Potential Field (APF) methods [26], [11], [35], [38], tree based planners [40], [49], Probabilistic Roadmap Methods (PRMs) [43], [67], [57], [32], and several variants which use heuristics [3], [10]. APF methods create a potential landscape and use gradient descent for navigation, plan locally, and can be dynamically reactive to unexpected obstacles. These methods generate an artificial potential in the robot’s workspace, which repels the robot from obstacles and attracts the robot to the goal [34]. Recent work has extended the capabilities of APF methods by addressing several known limitations, e.g., traps due to local minima and difficulty with narrow passages [26], [19]. Some work has been done to extend APFs to moving obstacles [38], by considering the trajectories of the obstacles. However, methodological approaches that address obstacles that are both stochastic and dynamic are lacking.

While control theoretic methods have been developed to provide mathematical assurances of safety despite stochasticity in low dimensional systems, they are computationally infeasible when the environment has tens to hundreds of dynamic obstacles. Stochastic reachability analysis provides offline verification of dynamical systems to assess whether the state of the system will, with a certain likelihood, remain within a desired subset of the state-space for some finite time, or avoid an undesired subset of the state-space [2]. To solve problems in collision avoidance, the region in the relative state-space, which constitutes collision, is defined as the set of states the system should avoid [60], [33], and those configurations which will lead to collision with a given likelihood are described by the Stochastic Reachable set (SR set). Unfortunately, the computation time for SR sets is exponential in the dimension of the continuous state, hence assessment of collision probabilities with many simultaneously moving obstacles is not feasible.

We propose a solution that combines ad-hoc and formal methods, to incorporate the effect of likely obstacle motion into the desired path, and exploit computationally efficient paradigms that can be used in real time. Our method weights an artificial potential field with stochastic reachable sets, computed pairwise between the robot and each stochastic, dynamic obstacle. Computational efficiency is achieved by

pre-computing the SR sets offline for a finite set of obstacle types, then querying those sets at run-time to construct a repulsive potential field around each obstacle.

Preliminary versions of this method were implemented via roadmap methods for dynamic path queries (SR-Query) [43] and APF with SR sets (APF-SR) for stochastic obstacles that followed simple line or arc trajectories [13] (but did not switch between these trajectories). SR-Query was more successful in identifying collision-free paths in environments with 50 moving obstacles than a roadmap-based approach that simply pruned invalid edges during dynamic path queries [32], but was susceptible to fast unseen moving obstacles, due to limited reactivity and required navigation on the roadmap edges. In [13], we demonstrated the advantages of APF methods over roadmap-based methods in environments with up to 300 stochastic, dynamic obstacles. We also evaluated the improved performance of incorporating SR sets with APF methods, as opposed to ad-hoc methods [38], [66], [27] for computing repulsion fields.

In this paper, our main contributions are to a) extend APF-SR to accommodate stochastic *hybrid* obstacles, which are more realistic and capture behavior that is much more representative of real-world dynamic obstacles, b) present a thorough parameter exploration of APF-SR and comparative analysis between APF-SR and other methods, and c) demonstrate our method in environments with up to 900 obstacles (approximately an order of magnitude more obstacles than we considered in previous implementations, while remaining real-time capable). Our method has several advantages over existing moving obstacle solutions and over SR sets alone. First, at runtime, the method is fast because it does not have to make expensive collision detection calls and simply queries the SR sets. Second, it scales well with many obstacles. Furthermore, it provides a framework in which multiple SR sets can be combined to generate approximate collision probabilities with many moving obstacles, which would otherwise be impossible using a single SR set that accounts for all obstacles simultaneously. Finally, by using SR sets for the underlying collision probability calculation, the method provides an upper bound on the probability of collision, which can be used to select the best path.

In Section II, we discuss related work in combining formal and ad-hoc methods for autonomous navigation. Section III describes the problem formulation, modeling, and stochastic reachability analysis. Section IV presents APF-SR. In Section V, we begin by conducting an extensive parameter evaluation. We then perform a comparison of success rate, path length and cause of failure with a holonomic and a unicycle robot in environments with up to 900 obstacles. We show that APF-SR outperforms the Gaussian APF comparison methods by up to 60% in the holonomic case and up to 20% in the unicycle case with 900 obstacles. Lastly, Section VI summarizes our contribution.

## II. RELATED WORK

The field of motion planning is vast, and there are countless approaches, both formal and ad-hoc, for designing strategies to

navigate robots through complex environments while achieving certain goals. Ad-hoc methods typically demonstrate their effectiveness through simulated performance, while formal methods use verification and controller synthesis techniques to provide theoretical performance guarantees.

APF methods are a popular ad-hoc approach for solving path planning problems, because of their simplicity, fast execution time, and applicability to several robotic problems, including unmanned aerial vehicles [11], [35], robot soccer [71], and mobile robots [26], [19], [65], [59], [46]. APF methods work by creating an artificial repulsion field around obstacles and an attraction vector towards the goal. The robot then navigates by following the gradient. For example, a recently developed APF method assigns non-uniform repulsive bubbles around moving human obstacles to prevent robots from moving in front of a walking human [38].

Recent work has extended the APF method to account for cases in which the goal is not reachable due to obstacle proximity [26], and navigation in narrow passages is required [19]. Other recent work has focused on modification of the computation of the potential field through fuzzy [59] and evolutionary [65] APFs. Another branch of work on APFs utilizes the repulsive and attractive concepts of APFs, but also integrates another path planning method [31], [55], [14]. For example, [31] uses a user-defined cost-map to influence node placement in a Rapidly exploring Random Tree (RRT) algorithm. The cost-map dictates a repulsiveness or attractiveness factor for every region. Similarly, Navigation Fields [55] assign a gradient that agents follow, and is used for crowd modeling. Finally, path guided APF methods utilize a global path planning method to find a desired path from start to goal, but then use APF for local planning decisions [14].

Many APF methods, however, do not exploit information about the possible future behavior of dynamic obstacles. One class of methods that does exploit possible future behavior is multi-resolution planning. Traditionally, multi-resolution planning uses a coarse grid to restrict the planning space on a global scale and then uses finer resolution grids in the restricted area [9], [5]. This allows for future states to be anticipated and planned for in the coarse restricted grid. However, recent work has started to combine high level global planning with varying underlying local planners. Some proposed methods use a global planner with a variant of APF for local planning [52], [28], [50], [51]. Voronoi decomposition was used in [52] to generate a global plan and then uses APF for local planning along the Voronoi edges. Similarly, a probabilistic roadmap was used in [28] to generate a global guidance path, and an APF-based local planner avoids dynamic obstacles and smooths the path. An APF variant was used in [50], [51] for reactive planning, but they also construct a hierarchy of successively more forward-looking path planners (eventually ending with a mission planner). While these planners can handle certain classes of moving obstacles, they do not directly consider the likelihood of robot-obstacle collisions.

Velocity obstacles (VO) [24] are widely used for path planning with moving obstacles. It is suitable for fast online planning in environments with hundreds of moving obsta-

cles. ORCA (Optimal Reciprocal Collision Avoidance) [69] extended this idea to multi-agent planning and was efficient enough to be implemented by mainstream video games. Additional VO formulation incorporated arcing motion obstacles, allowing an algorithm to handle both linear and arcing motion obstacles [39]. Frameworks have also been proposed for planning with stochastically moving obstacles by iterating through all possible robot velocities in order to approximate the true VO and find the optimal velocity [36]. The main drawback of VO-based collision avoidance is that the cost of VO calculation becomes prohibitively high for obstacles with complex geometry and stochastic motion.

Formal methods, in contrast, consider the likelihood that a path will result in a collision. By using a mathematical model of the system, and hence capturing possible future behavior, formal methods synthesize provably-correct paths using either optimal control or model checking algorithms, at the expense of scalability. Recent work has focused on logic languages (i.e., linear temporal logic) to express complex specifications for the robot, rather than simple criteria, such as “reach the goal while avoiding obstacles.” For example, the authors in [22] consider the temporal logic for motion planning with dynamic robots and static, known obstacles, and use formal methods to construct paths that are provably guaranteed to meet the desired specifications. Extensions of this type of technique to uncertain (but still static) environments are given in, e.g., [44] and [4]. A robot whose motion is probabilistic, and modeled as a Markov decision process (MDP), is considered in [37], and the optimal strategy and maximal probability of successful navigation in a static environment are computed.

Reachability analysis is another formal method, based on optimal control techniques, that is well-suited to *dynamic* environments. Reachability analysis can be used to generate optimal strategies for avoiding a certain region of the state space, while reaching a target set, through the use of a Hamilton-Jacobi-Isaacs (HJI) formulation that allows for both control and disturbance inputs [63]. The result is a maximal set of states (the reachable set), that when applied to collision avoidance problems, represents the set within which collision between two objects is guaranteed, subject to optimal robot actions and worst-case disturbance (i.e. obstacle) inputs [48], [45], [29]. In [62], reachable sets are calculated to assure a robot safely reaches a target while avoiding a single obstacle, whose motion is chosen to maximize collision, and the robot cannot modify its movements based on subsequent observations. A similar approach is taken in [17], but reachable sets are computed iteratively so that the robot can modify its actions. In [42], multiple obstacles that act as bounded, worst-case disturbances are avoided online, based on pre-computed reachable sets.

Stochastic reachability analysis extends the deterministic setting to allow for stochastic dynamics, and thus generates *probabilistic* reachable sets, which are the set of initial states from which a robot can reach a target while avoiding certain regions of the state space with a certain probability [56], [2], [61]. In [60], the desired target set is known, but the undesired sets that the robot should avoid are random and must be propagated over time. In [16], a two-player stochastic

dynamical game is applied to a target tracking application in which the target acts in opposition to the tracker. The computational techniques used to solve these reachability problems rely on solving a dynamic program by modeling the system as a finite state Markov decision process (MDP) [1], [21]. Such an approach suffers from an explosion in the number of states as the dimension of the system grows, and therefore for higher dimensional problems, this approach is not feasible.

Because the strengths of ad-hoc versus formal methods (speed and scalability versus performance guarantees, respectively) seemingly complement each other, there has recently been considerable interest in combining formal methods with ad-hoc planning methods. Luders et al. [41] applies robust control in the expansion step of the RRT algorithm. The algorithm constructs an RRT and performs a robust planning check along each new edge, which takes into account the possible obstacle locations and rejects non-robust paths. Somewhat similarly, [70] uses both a global and a local RRT to find guaranteed successful paths that satisfy both local and global objectives specified in linear temporal logic. The work in [58] uses a reactive controller which is tuned to avoid certain classes of pop-up obstacles, such as cylinders, combined with way-point following to guide a holonomic micro-UAV. While the method does not have a global path planning component, it still combines the ad-hoc way-points with a formal obstacle avoidance control method.

Similar to our work, Frazzoli et al. [25] uses a global path planner with an underlying control method. The authors propose using an RRT to build a feasible tree of paths from a start configuration to a goal configuration and then utilize an online controller to move between configurations in the tree. This online controller is capable of avoiding static obstacles. Alternately, [53] extends RRTs to dynamic obstacles by repairing subtrees invalidated by moving obstacles. Unlike our method, the computational cost of this RRT-based method is directly related to environment size, and stochastically moving obstacles are not handled. Lastly, [15] and [12] predict obstacle motion by Monte Carlo simulations and forward reachability analysis, respectively. Partial RRTs (which may not reach the goal) in the state-time space are grown in order to identify collision-free paths. However, unlike APF-SR, which has limited tunable parameters, the performance of these methods are sensitive to many scenario specific parameters.

### III. MODELING AND STOCHASTIC REACHABILITY ANALYSIS

#### A. Robot Dynamics

We consider two models for the dynamics of the robot: 1) a holonomic point-mass model, and 2) a non-holonomic unicycle model, with state  $\bar{x}^r = [x^r, y^r, \theta^r] \in \mathbb{R}^3$  representing the robot position and heading angle. The holonomic model

$$\begin{aligned} \dot{x}^r &= u^x \\ \dot{y}^r &= u^y \\ \dot{\theta}^r &= 0 \end{aligned} \quad (1)$$

has velocity control input  $u = [u^x, u^y] \in U_h \subset \mathbb{R}^2$ . The input is bounded, with  $U_h = [u_{\min}^x, u_{\max}^x] \times [u_{\min}^y, u_{\max}^y]$ . The non-holonomic unicycle model

$$\begin{aligned}\dot{x}^r &= u^s \cos \theta^r \\ \dot{y}^r &= u^s \sin \theta^r \\ \dot{\theta}^r &= u^w\end{aligned}\quad (2)$$

has control input  $u = [u^s, u^w] \in U_{nh} \subset \mathbb{R}^2$ , with speed  $u^s$  and angular velocity  $u^w$ . The input is bounded, with  $U_{nh} = [u_{\min}^s, u_{\max}^s] \times [u_{\min}^w, u_{\max}^w]$ . Discretizing the robot dynamics (1) and (2) using an Euler approximation with time step  $\Delta$  results in  $\bar{x}_{n+1}^r = \bar{x}_n^r + \Delta f^r(u_n, \theta_n^r)$ , with

$$f^r(u_n, \theta_n^r) = \begin{bmatrix} u_n^x \\ u_n^y \\ 0 \end{bmatrix}\quad (3)$$

for the holonomic robot and

$$f^r(u_n, \theta_n^r) = \begin{bmatrix} u_n^s \cos \theta_n^r \\ u_n^s \sin \theta_n^r \\ u_n^w \end{bmatrix}\quad (4)$$

for the unicycle robot.

## B. Obstacle Dynamics

Each obstacle is represented as a point mass with state  $\bar{x}^o = [x^o, y^o, \theta^o] \in \mathbb{R}^3$ , that follows either a straight-line or constant-arc trajectory with stochastic velocity  $w$  that represents linear or angular velocity, respectively. The random variable  $w$  takes values in the finite sets  $\mathcal{W}^{\text{line}}$  and  $\mathcal{W}^{\text{arc}_i}$ , respectively, with probability mass functions described by  $p^{\text{line}}(w)$  and  $p^{\text{arc}_i}(w)$  for  $i \in \{1, 2, 3\}$ . The obstacle dynamics discretized with time step  $\Delta$  are  $\bar{x}_{n+1}^o = \bar{x}_n^o + \Delta \cdot f^o(q_{n+1}, w_n, \theta_n^o)$ , with

$$f^o(\text{line}, w_n, \theta_n^o) = \begin{bmatrix} w_n \\ \gamma w_n \\ 0 \end{bmatrix}\quad (5)$$

for straight-line motion, with line slope  $\gamma \in \mathbb{R}$  determined by the heading angle  $\theta_n^o$  (i.e.  $\gamma = \tan \theta_0^o$ ), and

$$f^o(\text{arc}_i, w_n, \theta_n^o) = \begin{bmatrix} r_i w_n \cos \theta_n^o \\ r_i w_n \sin \theta_n^o \\ w_n \end{bmatrix}\quad (6)$$

for constant-arc movement, with radius  $r_i \in \mathbb{R}^+$ .

We allow the obstacles to switch between a straight line trajectory and one of three arc trajectories. Hence at any instant, the obstacle may take on continuous dynamics associated with one of four modes,  $\mathcal{Q} = \{\text{line}, \text{arc}_1, \text{arc}_2, \text{arc}_3\}$ , with arc trajectories distinguished correspondingly by distinct radii  $r_1 < r_2 < r_3$ . Further, we presume continuity of the heading angle, such that the angle  $\gamma$  of the line trajectory is completely specified by the obstacle heading at the previous instant, upon exiting an arc trajectory.

The switching dynamics are described by a stochastic process, such that the duration of time spent in a given mode is modeled similarly to an exponential distribution. We presume

that the likelihood of switching from mode  $q_i$  to mode  $q_j$ ,  $q_i, q_j \in \mathcal{Q}$ , at time  $n$  is given by

$$\begin{aligned}p_Q(\text{line}, \text{arc}_i) &= \frac{1}{3} (1 - \beta_n^{\text{line}}) \quad \forall i \in \{1, 2, 3\} \\ p_Q(\text{line}, \text{line}) &= \beta_n^{\text{line}} \\ p_Q(\text{arc}_i, \text{line}) &= 1 - \beta_n^{\text{arc}} \quad \forall i \in \{1, 2, 3\} \\ p_Q(\text{arc}_i, \text{arc}_i) &= \beta_n^{\text{arc}} \quad \forall i \in \{1, 2, 3\} \\ p_Q(\text{arc}_i, \text{arc}_j) &= 0 \quad \forall i, j \in \{1, 2, 3\}, i \neq j\end{aligned}\quad (7)$$

with  $\beta_n^{\text{line}} = e^{-\frac{\Delta(n-n_s)}{S}(1-R_{\text{line}})}$ ,  $\beta_n^{\text{arc}} = e^{-\frac{\Delta(n-n_s)}{S}(1-R_{\text{arc}})}$ ,  $\Delta \cdot n_s$  the time that the obstacle last switched,  $S$  the switching time parameter,  $R_{\text{line}}$  the fraction of obstacles in the planning space that are following line trajectories, and  $R_{\text{arc}} = 1 - R_{\text{line}}$ .

The switching time parameter allows for the tuning of the switching rates of obstacles, such that lower values of  $S$  increase the switching rate and higher values of  $S$  decrease the switching rate. For example, a switching time parameter set to a value much greater than the simulation running time will produce a negligibly small probability of an obstacle switching. This process assures that switching occurs roughly every  $S$  seconds, and also that the total number of obstacles in the planning space following arc and line trajectories remains approximately constant.

## C. Relative robot-obstacle dynamics

We model the relative dynamics between the robot and a single obstacle by examining the motion of the obstacle with respect to a coordinate frame affixed to the robot, via standard kinematic analysis. We define the relative state

$$\tilde{x} = \begin{bmatrix} R^T(\theta^r) \begin{bmatrix} x^o - x^r \\ y^o - y^r \end{bmatrix} \\ \theta^o - \theta^r \end{bmatrix} \in \mathbb{R}^3\quad (8)$$

in terms of a standard rotation matrix  $R(\cdot)$ , with dynamics

$$\begin{aligned}\tilde{x}_{n+1} &= \tilde{x}_n + \Delta \cdot \tilde{f}(q_{n+1}, u_n, w_n, \theta_n^r, \theta_n^o), \\ \tilde{f}(q_{n+1}, u_n, w_n, \theta_n^r, \theta_n^o) &= \begin{bmatrix} R^T(\theta_r) & | & 0_{1 \times 2} \\ \hline 0_{2 \times 1} & | & 1 \end{bmatrix} \cdot \\ &\quad (f^r(u_n, \theta_n^r) - f^o(q_{n+1}, w_n, \theta_n^o))\end{aligned}\quad (9)$$

For the purpose of computing the SR sets (but not for use in later simulation), we approximate the Poisson-like distribution (7) by a Markov process by presuming constant values for the switching likelihood, that is,

$$\begin{aligned}\beta_n^{\text{line}} &= \beta^{\text{line}} \\ \beta_n^{\text{arc}} &= \beta^{\text{arc}}\end{aligned}\quad (10)$$

This approximation is computed empirically for a given Poisson-like distribution by finding the average switching rate per time-step over 10,000 trials. The approximation enables us to express the resulting system as a discrete-time stochastic hybrid system (DTSHS), described by the tuple  $\mathcal{H} = (\tilde{\mathcal{X}}, \mathcal{Q}, \mathcal{U}, T_x, T_q)$ , with

- $\tilde{\mathcal{X}} \subseteq \mathbb{R}^3$  the set of continuous states representing relative position and heading

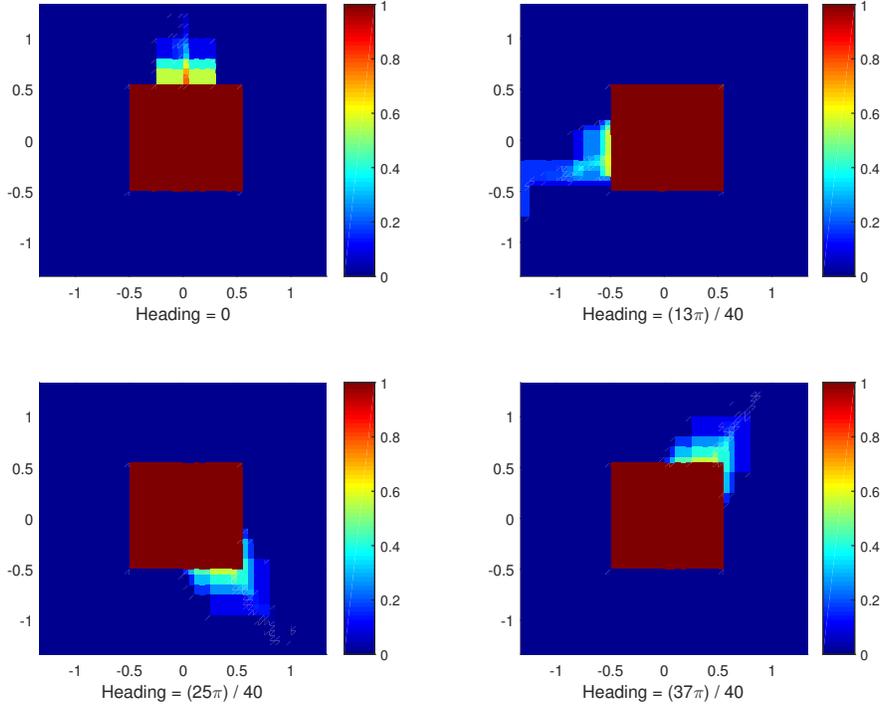


Fig. 1: Stochastic reachable set for relative robot-obstacle dynamics (9) with Markov switching (7), (10) in an arc mode. Since the SR set is  $3D_2$  we visualize the probability of collision with respect to relative position  $(\tilde{x}, \tilde{y})$  for four selected values of relative heading  $\theta$ .

- $\mathcal{Q} = \{\text{line}, \text{arc}_1, \text{arc}_2, \text{arc}_3\}$  a finite set of discrete modes, with  $\mathcal{S} = \mathcal{X} \times \mathcal{Q}$  the hybrid state space,
- $\mathcal{U} \subseteq \mathbb{R}^2$  a compact Borel space which contains all possible control inputs,
- $T_x : \mathbb{R}^3 \times \mathcal{Q} \times \mathcal{S} \times \mathcal{U} \rightarrow [0, 1]$  a stochastic transition kernel  $T_x(\tilde{x}_{n+1}|\tilde{x}_n, q_{n+1}, u_n)$  that assigns a probability distribution to  $\tilde{x}_{n+1}$  conditioned on  $\tilde{x}_n$ ,  $q_{n+1}$ , and  $u_n$ , and
- $T_q : \mathcal{Q} \times \mathcal{Q} \rightarrow [0, 1]$  a discrete transition kernel  $T_q(q_{n+1}|q_n)$  that assigns a probability distribution to  $q_{n+1}$  conditioned on  $q_n$ .

We assume the sets  $\mathcal{W}^{\text{line}}$  and  $\mathcal{W}^{\text{arc}}$  are finite, and therefore define the transition kernel  $T_x$  as

$$T_x(\tilde{x}_{n+1}|\tilde{x}_n, q_{n+1}, u_n) = \begin{cases} p^{\text{line}}(w_n) & \text{for } \tilde{x}_{n+1} = \tilde{x}_n + \Delta \tilde{f}(\text{line}, u_n, w_n, \theta_n^r, \theta_n^o) \\ p^{\text{arc}_i}(w_n) & \text{for } \tilde{x}_{n+1} = \tilde{x}_n + \Delta \tilde{f}(\text{arc}_i, u_n, w_n, \theta_n^r, \theta_n^o) \end{cases} \quad (11)$$

The transition kernel  $T_q$  is described by  $p_Q$  (7), with  $\beta_n^{\text{line}}$  and  $\beta_n^{\text{arc}}$  as in (10).

We combine the continuous and discrete state transition kernels for ease of notation, such that

$$\tau(\tilde{x}_{n+1}, q_{n+1}|\tilde{x}_n, q_n, u_n) = T_x(\tilde{x}_{n+1}|\tilde{x}_n, u_n, q_{n+1}) \cdot T_q(q_{n+1}|q_n) \quad (12)$$

#### D. Stochastic Reachable Sets for Collision Avoidance

We presume a collision occurs between the robot and a single obstacle whenever

$$\|\bar{x}_n^r - \bar{x}_n^o\|_1 \leq \epsilon \quad (13)$$

for some  $n$  and some constant distance  $\epsilon$ , and define the *avoid set*,  $\bar{K}$ , as the set of states in which (13) holds.

We generate collision avoidance probabilities through stochastic reachability analysis. To avoid collision with the obstacle, the robot should remain within  $K$ , the complement of  $\bar{K}$ . The probability that the robot remains within  $K$  over  $N$  time steps, with initial relative position  $\tilde{x}_0$ , can be calculated using dynamic programming [6], introduced for SR sets in [2]. To compute the SR set, we iterate a value function backwards in time from  $n = N$  to time  $n = 0$ ,

$$V_N^*(\tilde{x}, q) = \mathbf{1}_K(\tilde{x}) \quad (14)$$

$$V_n^*(\tilde{x}, q) = \max_{u \in \mathcal{U}} \mathbf{1}_K(\tilde{x}) \sum_Q \int_{\mathcal{X}} V_{n+1}^*(\tilde{x}', q') \cdot \tau(\tilde{x}', q' | \tilde{x}, u, q) d\tilde{x}' \quad (15)$$

in which an indicator function  $\mathbf{1}_K(x)$  is equal to 1 if  $x \in K$  and equal to 0 otherwise. The value function  $V_0^*(\tilde{x}_0, q_0)$  at time  $n = 0$  describes the probability of **avoiding** collision over  $N$  time steps when starting in some initial state  $\tilde{x}_0$  and initial mode  $q_0$ .

Note that (14)–(15) generally do not have a closed form expression, and must be evaluated for all possible  $(\tilde{x}_n, q_n) \in \tilde{\mathcal{X}} \times \mathcal{Q}$ . A discretization of  $\tilde{\mathcal{X}} \subseteq \mathbb{R}^3$  to a finite number of  $(\tilde{x}_n, q_n)$  results in an approximate solution, with smaller error corresponding to finer resolution of the discretization. We assume our discretization is sufficiently refined that we do not consider any errors in the resulting SR set because of this approximation, or the Markov switching approximation (10), and treat  $V_0^*(\tilde{x}_0, q_0)$  as the actual probability of collision.

Figure 1 depicts  $V_0^*(\tilde{x}_0, \text{arc})$ , the SR set for relative robot-obstacle dynamics (9) with Markov switching (7), (10) in an arc mode with a unicycle robot. The heat maps for four selected values of relative heading,  $\tilde{\theta}$ , show a higher probability of collision when the robot is in line with the obstacle’s trajectory. Intuitively, the closer the robot is to the obstacle, the higher the probability of collision. On a single core of an Intel 3.40 GHz CORE i7-2600 CPU with 8 GB of RAM, the SR set in Figure 2a took 1727.25 seconds to compute, over a horizon of  $N = 30$  steps, with time step of length  $\Delta = 1$ . We observed convergence in the stochastic reachable sets for  $N > 5$  since the robot and obstacle traveled sufficiently far apart within this time frame. The precomputed SR sets have a memory storage requirement of about 4.8MB for obstacles with hybrid dynamics. Using single mode dynamics (arc/line) as in [13], the memory requirement is 117KB. The SR set is computed over a grid which has  $121 \times 121 \times 18$  elements.

When used in environments with a single obstacle,  $V_0^*(\tilde{x}_0, q_0)$  (14) is the maximum probability of avoiding a collision and a tight upper bound. To consider environments with multiple obstacles, let  $B_i$  correspond to the event that the robot avoids collision with obstacle  $i \in \{1, \dots, M\}$ . We presume that collision avoidance probabilities are calculated separately for each obstacle, that is,  $V_0^{*,1}(\tilde{x}_0^1, q_0^1)$ ,  $V_0^{*,2}(\tilde{x}_0^2, q_0^2)$ ,  $\dots$ ,  $V_0^{*,M}(\tilde{x}_0^M, q_0^M)$  for relative position  $\tilde{x}_0^i$  with respect to obstacle  $i$  in mode  $q_0^i$ . The probability of avoiding collision with *all* obstacles is

$$\mathbb{P}[B_1 \cap B_2 \cap \dots \cap B_M] \leq \min\{V_0^{*,1}(\tilde{x}_0^1, q_0^1), V_0^{*,2}(\tilde{x}_0^2, q_0^2), \dots, V_0^{*,M}(\tilde{x}_0^M, q_0^M)\} \quad (16)$$

Hence, by computing the minimum value over all probabilities of collision avoidance with each obstacle individually, we obtain an upper bound to the total collision avoidance probability. While this upper bound does not provide a guarantee of safety, it can inform which paths are relatively more likely to avoid collision. Since our focus is on finding paths with higher success rates, rather than theoretically guaranteed collision-free paths, the upper bound (16) is appropriate. Further discussion and the derivation of (16) is given in [43].

#### IV. METHODS

In this section, we present a novel method for integrating SR sets with APF methods. First, the SR sets are calculated with a time step of  $\Delta$  seconds, however the robot and obstacle states are updated at a time step of  $\delta$  seconds such that  $\delta < \Delta$ . One hurdle in using SR sets to inform the potential field is

---

#### Algorithm 1 APF-SR

---

**Input:** obstacles  $O$  with state  $\bar{x}^o$  and mode  $q^o$ , robot  $r$  with state  $\bar{x}^r$  and mode  $q^r$ , smoothed SR sets  $V_0^{*'}(\tilde{x}, q^o)$

---

```

1: for  $t = 0$ ;  $t < \text{maxTime}$ ;  $t = t + \delta$  do
2:    $APF_{vector} = (0, 0)$ 
3:   for all obstacle  $o \in O$  do
4:     if  $\text{dist}(\bar{x}_t^o, \bar{x}_t^r) < d_{min}$  then
5:        $APF_{vector} = APF_{vector} +$ 
          $\text{getRepulsiveGradient}(\bar{x}_t^o, \bar{x}_t^r, q_t^o; V_0^{*'}(\tilde{x}_t, q_t^o))$ 
6:     end if
7:   end for
8:    $APF_{vector} = APF_{vector} + \text{goal-vector}$ 
9:    $u = \text{calcControl}(APF_{vector})$ 
10:   $\bar{x}_{t+\delta}^r = \bar{x}_t^r + \delta \cdot f^r(u, \theta_t^r)$ 
11: end for

```

---

the possibility of non-smoothness of  $V_0^*(\tilde{x}, q)$  in (14)–(15). In general, no guarantees of smoothness are possible. Since APF methods use a gradient as a warning that the robot is about to collide with an obstacle, we first smooth the SR set by convolving the set with a Gaussian,  $\mathcal{N}(\mu = 0, \sigma^2)$ , to produce  $V_0^{*'}(\tilde{x}, q)$ . While there are many forms of smoothing that can be used, we chose a Gaussian smoothing due its common incorporation in APF methods, simplicity of computation, and effectiveness for this application. Figure 2 shows two examples of smoothing SR sets. Figure 2a is the SR set for a holonomic robot and Figure 2c is the raw SR set for a unicycle robot. Figures 2b and 2d show the respective resulting sets after convolution with a Gaussian. As expected, the discontinuity from 0 to 1 at the obstacle boundary in Figures 2a and 2c is smoothed in Figures 2b and 2d, respectively.

Algorithm 1 calculates the APF gradient by summing the obstacle gradients, calculated in *getRepulsiveGradient* (Algorithm 2), and the goal-vector (Lines 3–8). The combination of the contributions of the individual obstacles provides the upper bound defined by (16). The gradient of the attractive and repulsive potentials is then used by *calcControl* to construct the control input  $u$  (Line 9). The goal-vector is a small magnitude vector which always points towards the goal relative to the robot’s current position. Thus, the APF gradient is the direction the robot should move in to avoid obstacles and reach the goal. Finally, the control law for the robot is updated with the control input  $u$  (Line 10).

The *getRepulsiveGradient* function, described in Algorithm 2, calculates the repulsive APF gradient for an obstacle with state  $\bar{x}_t^o$  and mode  $q_t^o$ . This gradient is calculated by first finding which cell of the smoothed SR set the relative state  $\tilde{x}_t$  is in for mode  $q_t^o$ , and denoting this cell as  $p_{i,j,k}$ . Thus,  $SR(p_{i,j,k})$  is the value in the stochastic reachable set at cell  $p_{i,j,k}$ . The gradient is then calculated by the (planar) second order central finite difference centered at  $i, j$ , by examining a ‘slice’ of the SR set for a given relative heading  $\tilde{\theta}_t$  with cell index  $k$ .

Once the  $APF_{vector}$  is calculated, then the control input  $u$  is calculated by the *calcControl* function. For the holonomic

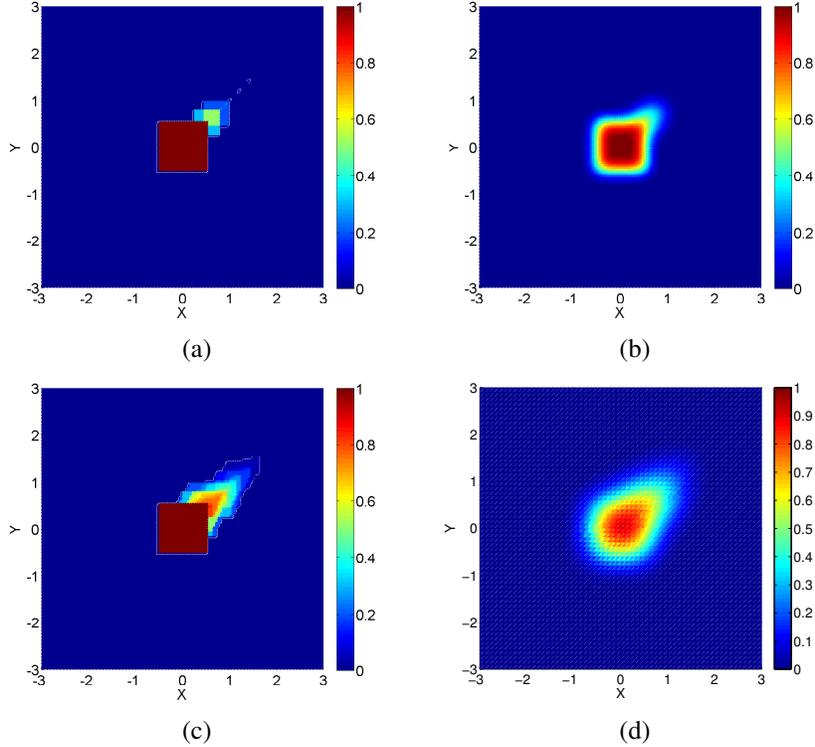


Fig. 2: SR sets for relative robot-obstacle dynamics in line mode; the color represents probability of collision. (a-b) Holonomic robot: (a) raw SR set (Discontinuities exist where color changes suddenly from red to blue.) (b) smoothed SR set after convolution with a Gaussian,  $\mathcal{N}(\mu = 0, \sigma^2)$ ,  $\sigma = 0.15$ . (c-d) Unicycle robot: (c) raw SR set (d) smoothed SR set after convolution with a Gaussian,  $\mathcal{N}(\mu = 0, \sigma^2)$ ,  $\sigma = 0.15$

---

**Algorithm 2** getRepulsiveGradient
 

---

**Input:** robot position  $\bar{x}_t^r$ , obstacle position  $\bar{x}_t^o$  and mode  $q_t^o$

**Output:** gradient  $G$

---

- 1:  $\{i, j, k\} = \text{cell}(\tilde{x}_t)$
  - 2:  $G_x = \frac{1}{2} (o.SR(p_{i-1,j,k}, o.q) + o.SR(p_{i-2,j,k}, o.q)) - \frac{1}{2} (o.SR(p_{i+1,j,k}, o.q) + o.SR(p_{i+2,j,k}, o.q))$
  - 3:  $G_y = \frac{1}{2} (o.SR(p_{i,j-1,k}, o.q) + o.SR(p_{i,j-2,k}, o.q)) - \frac{1}{2} (o.SR(p_{i,j+1,k}, o.q) + o.SR(p_{i,j+2,k}, o.q))$
  - 4:  $G = (G_x, G_y)$
- 

case,  $u = APF_{vector}$ . However, for the non-holonomic case a heading and speed must be extracted from the  $APF_{vector}$  to construct  $u = (u_s, u_w)$ . This is done by first setting  $u_w$  to the maximum turn rate in the direction of the  $APF_{vector}$ , then setting  $u_s$  to the maximum speed in the direction of the  $APF_{vector}$ . The maximum speed of the unicycle is the same as the maximum speed used in the SR calculation. Finally,  $u$  is the updated the control law for the robot.

## V. EXPERIMENTS

### A. Experimental Setup

Evaluation is performed in environments with up to 900 moving obstacles. In our evaluation, we define successful navigation as the ability to find a path from a start state to

goal state, without any collisions and within a specified time horizon. Two environments are used for evaluation. In the first environment, randomized initial obstacle start locations are placed in a circle of radius 50m. To maintain consistent obstacle density, an obstacle exiting the circle wraps around the boundary of the environment, re-entering  $\pi$  radians away from the point of exit, with the same velocity as upon its exit. Figure 3 shows the first environment. Black squares represent obstacles in line mode, and blue squares represent obstacles in arc mode. The robot (represented as the orange circle) must navigate from the box labeled ‘S’ (Start) to the box labeled ‘G’ (Goal). In the second environment, a regular pattern of obstacles is used, e.g., traffic lanes. The results for the second environment are shown in Section V-H.

We maintain the same values for model parameters in all experiments. For the obstacles, the set of velocities in line mode are  $\mathcal{W}^{\text{line}} = \{0.1, 0.2, 0.5, 0.7\}$  m/s, with corresponding probabilities  $p^{\text{line}}(w) = \{0.3, 0.2, 0.3, 0.2\}$ . In the three arc modes, the set of angular velocities are  $\mathcal{W}^{\text{arc}_i} = \frac{1}{5^i} \cdot \left\{ \frac{1.08}{2\pi}, \frac{1.622}{2\pi}, \frac{2.432}{2\pi}, \frac{3.24}{2\pi} \right\}$  radian/s for arcs of radius 5m, 10m, and 15m, respectively, with corresponding probabilities  $p^{\text{arc}_i}(w) \in \{0.2, 0.2, 0.3, 0.3\}$  that are the same in each mode. The angular velocities of arc obstacles are chosen such that the average linear speed is identical to line obstacles, regardless of arc radius.

The collision distance is determined by the obstacle body, and hence  $\epsilon = 1$ . The time step for the experiments is  $\delta = 0.1s$

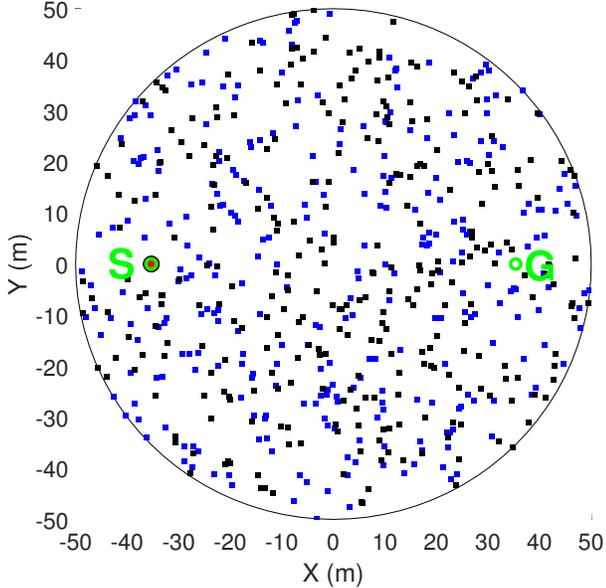


Fig. 3: Example of the circular environment with 600 obstacles. The red circle represents the robot’s location. S is the start, G is the goal, black boxes are line obstacles, and blue boxes are arc obstacles.

and for the stochastic reachable set calculations, we presume  $\Delta = 1s$ . The distance around the robot in which obstacles will affect the selection of the APF gradient is  $d_{min} = 3m$ . To implement the 3D stochastic reachable set calculations, we discretize relative heading in increments of  $\frac{\pi}{20}$  and in the experiment, use the corresponding planar stochastic reachable set that is closest to the current value of relative heading. This is the same discretization resolution used to compute the SR set.

We compare our method to other published methods that address moving obstacles. First, a Gaussian APF method was used, [46], with two parameterizations:  $\mathcal{N}(0m, (0.15m)^2)$  (Gaussian  $\sigma = 0.15$ ) and  $\mathcal{N}(0m, (0.45m)^2)$  (Gaussian  $\sigma = 0.45$ ). The Gaussian methods place a Gaussian potential field around moving obstacles. We selected two different standard deviation values to demonstrate the impact of increasing the safety margin around obstacles, but at the expense of making some paths infeasible due to the large repulsion area. Other parameters were evaluated but were found to not provide high success rates (results not shown). For example, a low  $\sigma$  value, such as 0.05m, allowed the robot to get too close to obstacles and often resulted in collision with fast moving obstacles. Contrarily, high values, such as 0.9m, resulted in overly conservative behaviors that hinder the robot’s ability to progress toward the goal in crowded environments. Second, a VO [24] was used that computed an avoidance vector based on the current state of obstacles in the environment. The Velocity Obstacle (VO) algorithm was adapted from the RVO2 C++ code base [68] implementation of the Optimal Reciprocal Collision Avoidance (ORCA) algorithm. This algorithm [69] was modified to allow for single-agent collision avoidance,

removing the reciprocal aspect of ORCA while maintaining many of ORCA’s linear programming optimizations. All experiments were run on a single core of an Intel i7-3720QM at 2.6GHz with 16GB of RAM, and were repeated 100 times. In Figures 4, 5, 7 and 12, the uncertainty in success rate is captured using the 99% confidence level derived from the central limit theorem and the variation in path length is depicted by standard deviation.

### B. Model Assumptions and Limitations

During the computation of the gradient during planning, we assume knowledge of the obstacle’s position  $\bar{x}_t^o$  and mode  $q_t^o$ , robot position  $\bar{x}_t^r$ , and goal position. These values can be obtained and updated anytime, and, for APF methods, frequent updates are common since accurate values are critical for finding the gradient. In this paper, we update their positions before each gradient computation. We assume obstacle dynamics are not impacted by outside forces, e.g., other obstacles. Therefore, in the highly crowded experiments shown, obstacle trajectories do not interact.

There are limitations inherent to APFs and SR sets that are also inherent to APF-SR. First, the point-mass robot model is a simplification of actual robot motion. However, methods such as [26] and [19] exist, which extend APF methods to non-point robots. Also, a more realistic robot model can be easily incorporated into the SR set calculation. Another common implicit limitation of APF methods is that small, non-convex obstacles are required to minimize the possibility of local minima in the potential field that may entrap the robot. One possible way to address this limitation is to combine the local method of APF-SR with a global planning method that considers possible local minima that occur in the static environment offline [14], [28]. In planning methods for moving obstacle avoidance, there is often an implicit assumption that obstacles move sufficiently slow. In this work, the size and possible velocities of the obstacle and robot are directly incorporated into the SR set computation. Therefore, the size and shape of the SR set reflects those velocities and enables planning even in the presence of fast-moving obstacles.

SR set computation is an expensive process that requires knowledge of the stochastic dynamics. We address this with an approximation that considers modes individually that performs similarly to the actual dynamics model in Section V-C. Beyond the scope of this work, this approximation facilitates the integration of new methods that might maintain a SR set database which could be used to match obstacle motion to sets (as was done with funnel libraries [42]) or predictions of dynamic obstacle motions [23].

### C. Stochastic Reachable Set Approximation

We first consider an approximation to the hybrid model in Section II. In many applications, the mode of the obstacle may not be known at all instants. Hence we consider an approximation to the SR set for the hybrid system described by (5), (6), (7), in which we compute the SR sets for non-switched systems with dynamics and distributions as described for each mode individually, and select the appropriate SR set in the

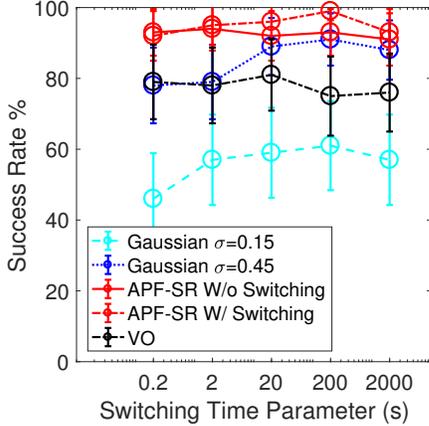


Fig. 4: Success rate vs. the switching time parameter appearing in (7). Smaller switching time parameter values indicate faster obstacles switching between a line and arc (or vice versa).

APF-SR algorithm. We believe that this approach would be more computationally efficient in realistic scenarios.

We evaluate in Figure 4 the efficacy of this approximation by comparing the performance of APF-SR over several switching time parameter values with the true SR set (denoted APF-SR w/o approx) and with the approximate SR sets, that are derived from constant-mode obstacles (denoted APF-SR w/ approx). The  $\beta^{\text{line}}$  and  $\beta^{\text{arc}}$  in APF-SR w/ approx have the values  $\{0.53, 0.18, 0.56, 0.02, 0.01\}$  for switching time parameter values  $S = \{0.2s, 2s, 20s, 200s, 2000s\}$ . This plot shows that the approximation produces success rates that are similar to without approximation. Due to this similarity, the approximation is used in all subsequent experiments.

The switching time parameter parameter can impact the planning complexity, with faster switching obstacles increasing problem complexity. Figure 4 shows that APF-SR is minimally impacted by the parameter. On the other hand, the Gaussian methods, which do not take into account the trajectories of the obstacles, are heavily affected by a rapid switching rate. Success rate is reduced by 10% for the fastest switching rate from the slowest switching rate. For the remaining experiments, we select a switching time parameter of 20s.

#### D. Method and Environmental Parameter Evaluation

In this set of experiments, we explore in detail the method and the environmental parameters that can impact APF-SR. All experiments are run in an environment with 300 obstacles, a goal-vector magnitude of 0.01, and a holonomic robot.

Table I shows the impact of the  $\sigma$  used for the Gaussian that smoothed the SR Set in APF-SR. The best performance occurs at  $\sigma = 0.15m$ . Thus, this value for  $\sigma$  will be used for smoothing all SR sets for APF-SR in the remaining experiments.

The ratio of line obstacles to arc obstacles is another important environmental parameter as arc obstacles have empirically been shown, for many methods, to be more complicated to plan for than line obstacles. Figure 5 shows the impact of

$\sigma$	0.05	0.15	0.45
Success Rate	$35 \pm 12\%$	$95 \pm 5\%$	$86 \pm 9\%$

TABLE I: Success rate vs.  $\sigma$  (in meters) used for smoothing the SR set in APF-SR. (goal-vector magnitude of  $g = 0.01$ )

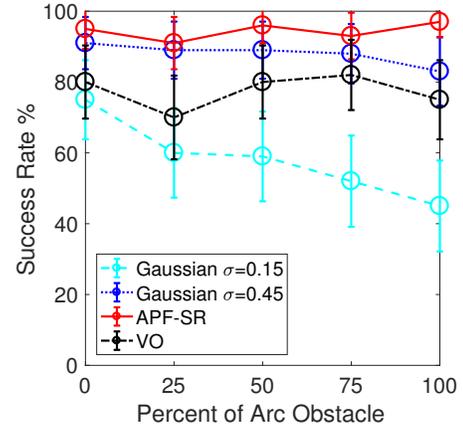


Fig. 5: Success rate vs. the percentage of obstacles moving in an arc trajectory, with a goal-vector magnitude of 0.01 and 300 obstacles.

varying the ratio of obstacle types on the success rates of the various planning methods. Unlike the other experiments, the obstacles are not allowed to switch dynamics; instead, the ratio of obstacles types are varied from 0 to 100%. For APF-SR, the success rate is approximately constant around 95% regardless of the percentage of line obstacles. It is important to note that the possible radii of the arc obstacles were chosen such that the difference between the line trajectory and the arc is large. Unlike APF-SR, the success rate for Gaussian  $\sigma = 0.15$  is reduced by 6% with 100% arcs, and the success rate for Gaussian  $\sigma = 0.45$  is reduced by 34% with 100% arcs. Similarly, VO hovers around 80% success rate for all the switching time parameter values (15% less than APFSR), which indicates that VO is affected more by the number of obstacles than by their trajectories.

#### E. Holonomic Robot Experiments

Figure 6 demonstrates the success rate versus number of obstacles (300 to 900) for the APF-based methods when varying goal-vector magnitude with a holonomic robot. APF-SR (Figure 6a and 6b) and the two Gaussian APFs (Figures 6c-6f) show that the goal-vector has a consistent effect across all three methods with goal-vector magnitude of 0.01 providing on average the best overall balance between success rate and path length regardless of the number of obstacles. An additional test (labeled “off” in Figure 6) is run with the goal-vector magnitude set to 0.01 when all obstacles are more than 3m units away and set to 0 when at least one obstacle is closer than 3m away. This test allows the algorithm to attempt maximal avoidance when obstacles are nearby. Second, APF-SR does better than either of the Gaussian method parameterizations and VO. The slopes in Figure 6 are approximately the same for each method. This indicates that

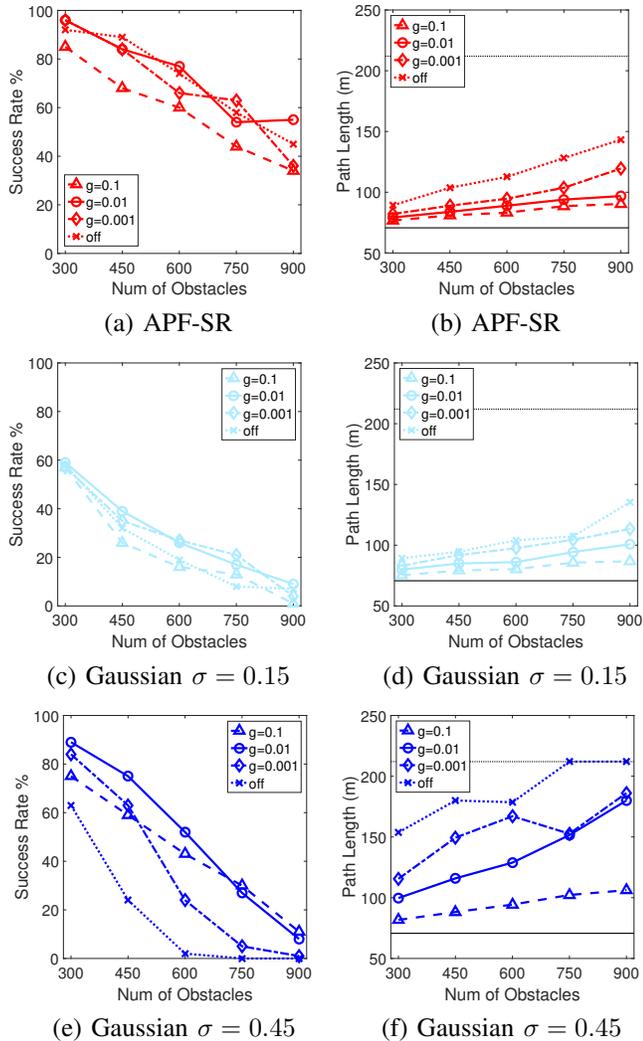


Fig. 6: Holonomic robot success rate and path length for various methods.  $g$  is the goal-vector, and  $g = \text{off}$  indicates that the goal-vector is 0 when the robot is under the influence of an obstacle’s APF and is 1 otherwise. The solid line at path length 70m indicates the theoretical shortest path possible (i.e., straight line from the start to the goal). The dotted line at path length 210m indicates a cutoff point where the run is considered a failure.

the difficulty of the problem increases linearly with the number of obstacles. APF-SR has a similar slope to the Gaussian  $\sigma = 0.15$ , but for 300 obstacles the success rate is higher (95% compared to 60% for a goal-vector magnitude of 0.01). Interestingly, the slope for Gaussian  $\sigma = 0.45$  is steeper than the other methods, but it has a success rate of 89% for 300 obstacles, which indicates that the greater repulsion region aids in path planning for sparse environments but prevents the robot from navigating in cluttered environments.

Total path length captures how much the robot is forced to deviate from the straight-line path because of obstacles. Figures 6b, 6d and 6f show the average path length versus the number of obstacles for APF-SR and the two Gaussian comparison methods (only recorded for collision-free runs).

These figures show that, as expected, the path length increases as the goal-vector gets smaller. This indicates that if the goal-vector is too strong the robot does not react enough to the obstacles’ potential fields, but if the goal-vector is too small (the black dotted line) the robot does not make enough progress towards the goal and spends too much time in the obstacle field which increases the probability of colliding. This trend holds for APF-SR and the two Gaussians, however this trend is scaled relative to the potential fields used. Gaussian  $\sigma = 0.45$  has the widest field and, as such, is affected the most by the obstacles. Gaussian  $\sigma = 0.15$  is more similar in size to APF-SR, but it does not consider the obstacle trajectory. Thus, its path lengths are similar, but its success rate is lower. The shaped potential field of APF-SR allows the robot to navigate around obstacles in a safe manner by avoiding entering the obstacle trajectory (like the large Gaussian method) but still have a relatively small field which allows it to move between dense obstacle clusters.

After selecting a goal-vector of 0.01 for the APF-based methods, we evaluated success rate as a function of the number of obstacles in Figure 7 for all comparison methods. Figure 7a shows that APF-SR has the highest success rate over all quantities of obstacles. VO has the second highest success rate, in most cases. The Gaussian methods are the most impacted by the increasing number of obstacles. Figure 7b shows APF-SR has a similar path length to Gaussian  $\sigma = 0.15$ . This could be because SR sets were convolved with a Gaussian function with the same sigma. VO provides a slightly shorter path length in most cases.

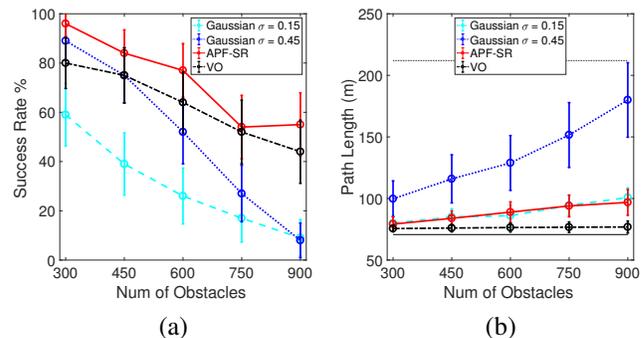


Fig. 7: Holonomic robot (a) success rate and (b) path length. The goal-vector is 0.01 for all APF-based methods.

We also investigate what situations cause APF-SR to fail. Figure 8 shows the multi-obstacle collision percentage, that is, the percent of collisions due to either, one, two or three nearby obstacles (not percentage of interactions which led to collision). The failure case occurrences are normalized for each planner/robot combination. The total number of failures for holonomic, unicycle dynamics, respectively are: 4, 14 for APF-SR, 40, 66 for Gaussian  $\sigma=0.15$ , and 11, 40 for Gaussian  $\sigma=0.45$ . The test is conducted in the circular environment with 300 moving obstacles and the goal-vector is set to 0.01 for all methods. The value of  $d_{min}$  is used to define proximity. Gaussian  $\sigma = 0.15$  collides mostly with single obstacles. This is likely due to the fact that low  $\sigma$  allows the robot to get too

close to obstacles, and thus is unable to avoid collision with fast moving obstacles. In contrast, a larger Gaussian,  $\sigma = 0.45$ , more often collides with two obstacles. The wider Gaussian provides a larger buffer between the robot and single obstacles, thus reducing the collisions with a single obstacle.

SR sets provide probabilistic collision avoidance when the corresponding optimal control for that SR set is implemented. This means that applying the optimal control will almost surely result in collisions when enough trajectories are simulated. This is likely the phenomena underlying the collisions with a single obstacle. Another possible source of the collision is due to the fact that the controller implemented is not necessarily the optimal control as computed by (14), (15). In addition, we can provide no guarantee of success when planning for more than one nearby obstacle, since the SR sets are only computed for single obstacle interactions. Figure 8 confirms this, as the APF-SR method fails most often when multiple obstacles are nearby. Essentially, conflicting APF gradients from multiple obstacles can cause the APF-SR method to fail.

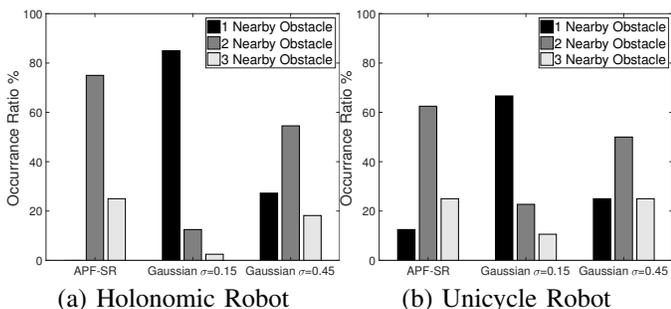


Fig. 8: Analysis of collisions. Percent of collisions due to 1, 2 or 3 nearby obstacles for various methods, for (a) holonomic and (b) unicycle robots. An obstacle is nearby if the distance between the robot and the obstacle is less than 3m.

Table II indicates that APF-SR has very low computation time per planning step. In the extremely crowded 900 obstacles environment, the 0.13ms computation time per planning step for APF-SR is comparable to the Gaussian APFs and is much faster than VO (16ms). APF-SR has a low computation time since the algorithm simply queries the precomputed SR set for the construction of repulsive potentials. Therefore, the computation time is very low. In this problem with obstacles that are distributed uniformly at random, we observe an expected linear scaling in computation time to the number of obstacles. The high computation time of VO compared to APF methods is due to implementation [68]; re-building the K-D tree every planning step is computationally intensive.

# of Obstacles	300	450	600	750	900
Gaussian $\sigma = 0.15$	0.036	0.052	0.068	0.083	0.10
Gaussian $\sigma = 0.45$	0.036	0.051	0.068	0.083	0.10
APF-SR	0.045	0.065	0.087	0.11	0.13
VO	5.1	7.8	11.0	13.5	16.0

TABLE II: Computation time per planning step in ms of APF-SR and comparison methods.

## F. Unicycle Robot Experiments

In this set of experiments, we presume unicycle robot dynamics. The robot is limited to a turning rate of  $\frac{\pi}{12}$  per time step  $\Delta$ . This increases the difficulty of the problem as the robot cannot instantly change heading to avoid a collision.

Figure 9 shows that both APF-SR and the Gaussian methods all have relatively lower success rates than in the holonomic case. However, the Gaussians suffer much more than APF-SR. The best Gaussian success rate (for 300 obstacles) reduces from 90% in the holonomic case to 62% in the unicycle case, but APF-SR reduces from 95% to 84%. Furthermore, as the number of obstacles increase, the Gaussians rapidly approach 0% success rate. Since [68] cannot be directly applied to non-holonomic robots without significant modifications, we do not include a comparison to VO.

For the holonomic case a goal-vector magnitude of 0.01 was on average better than any other goal attraction (Figures 6). However, for the unicycle case (Figures 9), success rates for a goal-vector magnitude of 0.01 and 0.1 oscillate. This is likely due to the increased difficulty of the problem which greatly increases the probability of collision the longer the robot is in the environment. Furthermore, the success rate slope for APF-SR is the steepest. While this indicates that APF-SR's success degrades faster with increasing number of obstacles, the success rate is still higher than the Gaussian comparison methods.

Figures 9b, 9d and 9f show the path length versus the number of obstacles. Again, as expected the path length increases as the goal-vector decreases, and the path length increases as the number of obstacles increases. As seen in the Gaussian  $\sigma = 0.45$  path length plot (Figure 9f), the path length increases substantially for the few collision-free paths in highly crowded environments (more than 600 obstacles). This is due to paths that were unable to arrive at the goal before an imposed cutoff. For Gaussian  $\sigma = 0.15$ , there was only one collision-free path in the 900 obstacle environment with goal-vector 0.01 (Figure 9d).

These experiments have shown that APF-SR is able to path plan in environments that have up to 900 hybrid dynamic moving obstacles with a high success rate. Furthermore, APF-SR is significantly more robust to the hybrid dynamics than the comparison methods, and the increased success is due to encoding the relative obstacle robot dynamics in the SR set used to produce the potential fields for the obstacles. Thus, APF-SR is able to make more informed path planning decisions to avoid moving obstacles.

Figure 8b shows the number of nearby obstacles when each method fails. The same trend as the holonomic case is exhibited. Specifically, there are more collision events for all methods since the unicycle robot dynamics is more difficult, but APF-SR rarely collides with only one nearby obstacle.

## G. APF-SR in Environments with Sensor Noise

Real-world robots often face environment sensing uncertainty. In this experiment, we simulate uncertainty via randomly sampling the perceived obstacle position every time step with a bounded uniform distribution centered around

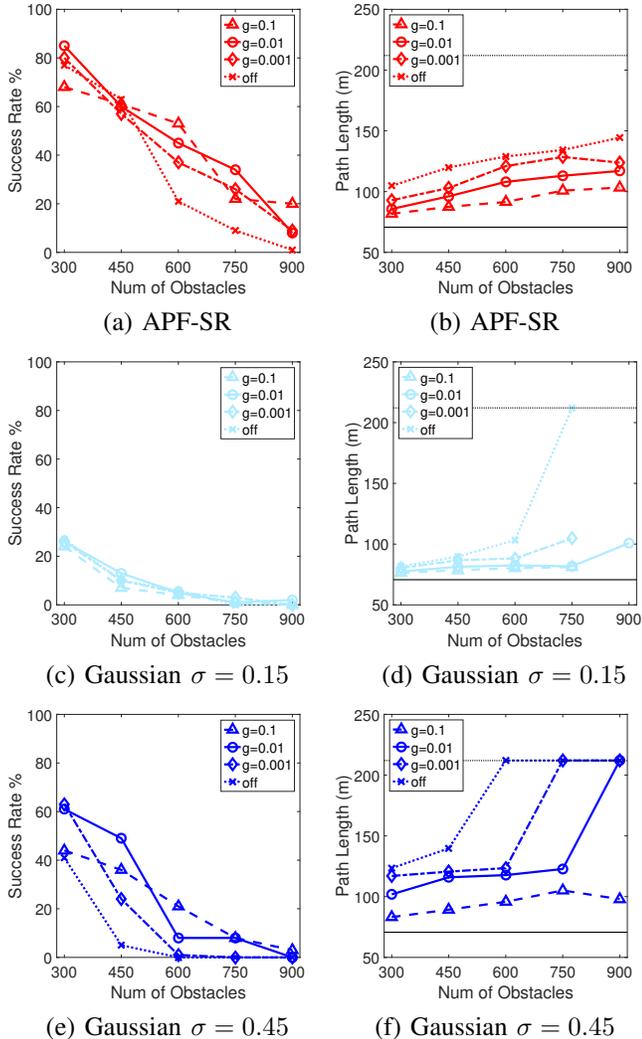


Fig. 9: Unicycle robot success rate and path length for various methods.  $g$  is the goal-vector, and  $g = \text{off}$  indicates that the goal-vector is 0 when the robot is under the influence of an obstacle's APF and is 1 otherwise.

the true position of the obstacle. The uniform distribution is bounded by 10, 25 and 50% of the obstacle width (1m). For example, if the noise is bounded by 50% and the obstacle's true position is at the origin, the perceived position of that obstacle can be anywhere within the box of (0.5m, 0.5m), (-0.5m, 0.5m), (-0.5m, -0.5m) and (0.5m, -0.5m). The SR sets used was described in Section III, which does not consider robot sensor uncertainty. The remaining experimental setup is the same as Section V-E.

Figure 10 shows the success rate of APF-SR is largely unaffected by sensor uncertainty, except in highly crowded cases with the largest amounts of noise. For all noise levels, APF-SR has a higher success rate than the Gaussians despite the lack of noise in the SR set computation. All methods are likely assisted by frequent observation of nearby obstacles and frequent replanning (100 Hz). Therefore, incorrect obstacle position observations may be averaged out.

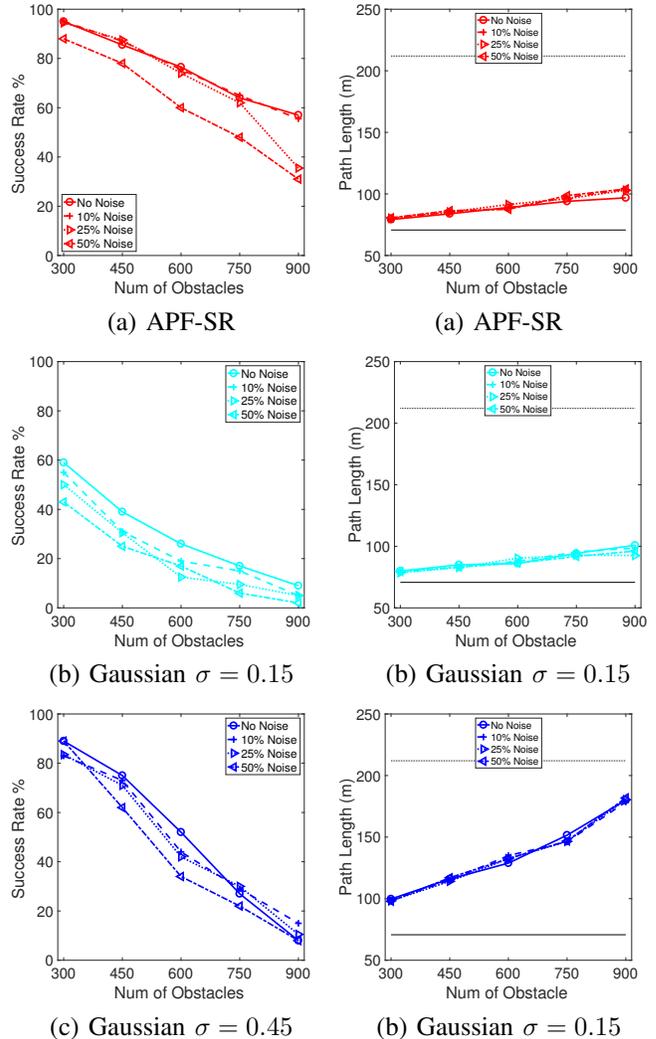


Fig. 10: Success rate and path length vs number of obstacles in the presence of obstacle position noise. A 10% noise level means the sensed position of the obstacle can deviate from the true position according to a uniform random distribution up to 10% of the obstacle width.

#### H. Lanes Environment

In this experiment, we tested APF-SR in an environment with structured obstacle motions. This contrasts the random obstacle placement in the previous experiments. This new environment, Lanes, requires the holonomic robot to travel across 18 lanes of moving obstacles from a start (-15m, -7.5m) to a goal (15m, 7.5m) (Figure 11). The dynamic obstacles travel in lanes, and the speed of each obstacle is sampled stochastically using line dynamics (5). Lane direction alternates sequentially. For example, the obstacles in the bottom lane travel from left to right while the obstacles in the lane above travel from right to left. In order to maintain the obstacle density, an obstacle is transported to the opposite side of its lane upon exiting the domain boundary. The robot is a holonomic point robot with a maximum speed of 0.36m/s, identical to previous experiments.

Figure 12 shows APF-SR has a much higher success rate

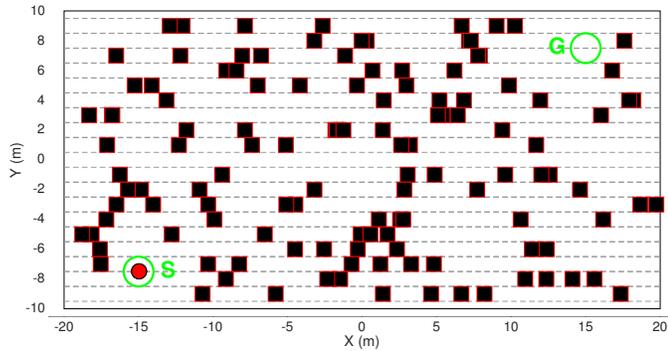


Fig. 11: Lanes environment. The robot must travel from S to G by crossing lanes without colliding with moving obstacles (black squares) or leaving the domain (solid black boundary). 7 obstacles per lane is shown.

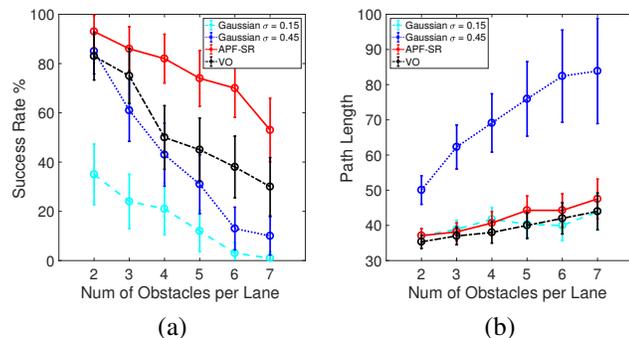


Fig. 12: Lanes environment (a) success rate and (b) path length with varied number of obstacles per lane. The goal-vector is 0.01 for all APF methods.

compared to the other methods (over 20% higher for 7 obstacles per lane). As seen in the reduced success rate, VO struggles in this problem due to the density and proximity of obstacles. This factor also impacts the Gaussians whose success rates are the lowest of all the methods. All path lengths are similar, except Gaussian  $\sigma = 0.45$ . This method exhibits a path length that is at least 20% longer than the other methods. This is expected since it attempts to maintain a high clearance from obstacles in this crowded environment.

## VI. CONCLUSION

The incorporation of the formal SR sets into the ad-hoc APF method produces a more accurate representation of the relative robot-obstacle dynamics, which leads to an increased success rate during path planning. Our experiments indicate that APF-SR has a 10%-30% higher success rate than comparison methods in all cases, including problems with 900 moving obstacles. The SR set informs APF-SR of the direction and velocity of the obstacle, which is used to generate a repulsive potential that reflects the probability of collision. Hence, the APF-SR algorithm can make informed planning decisions even in the presence of multiple moving obstacles. We showed that APF-SR is robust to the primary parameters in the method, and demonstrated that the method is capable of path planning

in highly complex and dynamic environments with obstacles that can switch between line- and arc-following dynamics.

## REFERENCES

- [1] A. Abate, J.-P. Katoen, J. Lygeros, and M. Prandini. Approximate model checking of stochastic hybrid systems. *European Journal of Control*, 16(6):624–641, 2010.
- [2] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.
- [3] R. Al-Hmouz, T. Gulrez, and A. Al-Jumaily. Probabilistic road maps with obstacle avoidance in cluttered dynamic environment. In *IEEE Intelligent Sensors, Sensor Networks and Information Processing Conf.*, pages 241–245, 2004.
- [4] A. I. M. Ayala, S. Andersson, and C. Belta. Temporal logic motion planning in unknown environments. In *IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 5729–5284, 2013.
- [5] S. Behnke. Local multiresolution path planning. In *Robocup 2003: Robot Soccer World Cup VII*, pages 332–343. Springer, 2004.
- [6] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Sci., 2005.
- [7] A. Bicchi and L. Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):221–231, 2000.
- [8] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988.
- [9] R. Bohlin. Path planning in practice; lazy evaluation on a multi-resolution grid. In *IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 49–54, 2001.
- [10] R. Bohlin and L. E. Kavraki. Path planning using Lazy PRM. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 521–528, 2000.
- [11] O. Cetin, S. Kurnaz, O. Kaynak, and H. Temeltas. Potential field-based navigation task for autonomous flight control of unmanned aerial vehicles. *Int. J. of Autom. and Cont.*, 5(1):1–21, 2011.
- [12] H.-T. Chiang, B. HomChaudhuri, A. P. Vinod, M. Oishi, and L. Tapia. Dynamic risk tolerance: Motion planning by balancing short-term and long-term stochastic dynamic predictions. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, To appear, 2017.
- [13] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia. Aggressive moving obstacle avoidance using a stochastic reachable set based potential field. In *Algorithmic Foundations of Robotics XI*, pages 73–89. Springer, 2015.
- [14] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia. Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2347–2354, 2015.
- [15] H.-T. Chiang, N. Rackley, and L. Tapia. Stochastic ensemble simulation motion planning in stochastic dynamic environments. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 3836–3843, 2015.
- [16] J. Ding, M. Kamgarpour, S. Summers, A. Abate, J. Lygeros, and C. Tomlin. A stochastic games framework for verification and control of discrete time stochastic hybrid systems. *Automatica*, 49(9):2665–2674, 2013.
- [17] J. Ding, E. Li, H. Huang, and C. Tomlin. Reachability-based synthesis of feedback policies for motion planning under bounded disturbances. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2160–2165, 2011.
- [18] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin. Reachability calculations for automated aerial refueling. In *IEEE Conf. on Decision and Cont.*, pages 3706–3712, 2008.
- [19] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *Int. J. Robot. Res.*, 29(5):485–501, 2010.
- [20] J. M. Duperret, M. R. Hafner, and D. D. Vecchio. Formal design of a provably safe robotic roundabout system. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 2006–2011, 2010.
- [21] S. Esmail Zadeh Soudjani and A. Abate. Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. *SIAM Journal on Applied Dynamical Systems*, 12(2):921–956, 2013.
- [22] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. Pappas. Temporal logic motion planning for dynamic robots. *Automatica*, 45(2):343–352, 2009.

- [23] S. Ferguson, B. Luders, R. C. Grande, and J. P. How. Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions. In *Algorithmic Foundations of Robotics XI*, pages 161–177. Springer, 2015.
- [24] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.*, 17:760–772, 1998.
- [25] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002.
- [26] S. S. Ge and Y. J. Cui. New potential functions for mobile robot path planning. *IEEE Trans. Robot. Automat.*, 16(5):615–620, 2000.
- [27] S. S. Ge and Y. J. Cui. Dynamic motion planning for mobile robots using potential field method. *Auto. Robots*, 13(3):207–222, 2002.
- [28] R. Geraerts and M. H. Overmars. The corridor map method: Real-time high-quality path planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1023–1028, 2007.
- [29] J. H. Gillula, G. M. Hoffmann, H. Haomiao, M. P. Vitus, and C. J. Tomlin. Applications of hybrid reachability analysis to robotic aerial vehicles. *Int. J. Robot. Res.*, 30(3):335–354, 2011.
- [30] I. Hwang and C. E. Seah. Intent-based probabilistic conflict detection for the next generation air transportation system. *Proceedings of the IEEE*, 96(12):2040–2059, 2008.
- [31] L. Jaillet, J. Cortés, and T. Siméon. Sampling-based path planning on configuration-space costmaps. *IEEE Trans. Robot.*, 26(4):635–646, 2010.
- [32] L. Jaillet and T. Siméon. A PRM-based motion planner for dynamically changing environments. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 1606–1611, 2004.
- [33] M. Kamgarpour, J. Ding, S. Summers, A. Abate, J. Lygeros, and C. Tomlin. Discrete time stochastic hybrid dynamical games: Verification and controller synthesis. In *IEEE Conf. on Decision and Cont.*, pages 6122–6127, 2011.
- [34] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.*, 5(1):90–98, 1986.
- [35] T. Khuswendi, H. Hindersah, and W. Adiprawita. UAV path planning using potential field and modified receding horizon A\* 3D algorithm. In *Int. Conf. on Electrical Eng. and Informatics (ICEEI)*, pages 1–6, 2011.
- [36] B. Kluge. Recursive agent modeling with probabilistic velocity obstacles for mobile robot navigation among humans. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 376–380, 2003.
- [37] M. Lahijanian, S. B. Andersson, and C. Belta. Temporal logic motion planning and control with probabilistic satisfaction guarantees. *IEEE Trans. Robot.*, 28(2):396–409, 2012.
- [38] C.-P. Lam, C.-T. Chou, K.-H. Chiang, and L.-C. Fu. Human-centered robot navigation towards a harmoniously human–robot coexisting environment. *IEEE Trans. Robot.*, 27(1):99–112, 2011.
- [39] F. Large, S. Sckhavat, Z. Shiller, and C. Laugier. Using non-linear velocity obstacles to plan motions in a dynamic environment. In *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 734–739, 2002.
- [40] H.-C. Lee, T. Yaniss, and B.-H. Lee. Grafting: A path replanning technique for rapidly-exploring random trees in dynamic environments. *Advanced Robotics*, 26(18):2145–2168, 2012.
- [41] B. Luders and J. How. An optimizing sampling-based motion planner with guaranteed robustness to bounded uncertainty. In *American Control Conference (ACC)*, 2014, pages 771–777, June 2014.
- [42] A. Majumdar and R. Tedrake. Robust online motion planning with regions of finite time invariance. In *Algorithmic Foundations of Robotics X*, pages 543–558. Springer, 2013.
- [43] N. Malone, K. Lesser, M. Oishi, and L. Tapia. Stochastic reachability based motion planning for multiple moving obstacle avoidance. In *Hybrid Systems: Computation and Control (HSCC)*, pages 51–60, 2014.
- [44] M. Maly, M. Lahijanian, L. E. Kavradi, H. Kress-Gazit, and M. Y. Vardi. Iterative temporal motion planning for hybrid systems in partially unknown environments. In *Hybrid Systems: Computation and Control (HSCC)*, pages 353–362, 2013.
- [45] K. Margellos and J. Lygeros. Hamilton-Jacobi formulation for reach-avoid problems with an application to air traffic management. In *American Control Conference (ACC)*, pages 3045–3050, 2010.
- [46] M. Massari, G. Giardini, and F. Bernelli-Zazzera. Autonomous navigation system for planetary exploration rover based on artificial potential fields. In *Proceedings of Dynamics and Control of Systems and Structures in Space (DCSSS)*, pages 153–162, 2004.
- [47] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, 1998.
- [48] I. Mitchell, A. Bayen, and C. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *Trans. on Auto. Cont.*, 50(7):947–957, 2005.
- [49] V. Narayanan, M. Phillips, and M. Likhachev. Anytime safe interval path planning for dynamic environments. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 4708–4715, 2012.
- [50] M. Nieuwenhuisen and S. Behnke. Hierarchical planning with 3D local multiresolution obstacle avoidance for micro aerial vehicles. In *Proceedings of the Joint Int. Symposium on Robotics (ISR) and the German Conference on Robotics (ROBOTIK)*, pages 1–7, 2014.
- [51] M. Nieuwenhuisen and S. Behnke. Layered mission and path planning for mav navigation with partial environment knowledge. In *Intelligent Autonomous Systems*, pages 307–319. Springer, 2016.
- [52] K. Ok, S. Ansari, B. Gallagher, W. Sica, F. Dellaert, and M. Stilman. Path planning with uncertainty: Voronoi uncertainty fields. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4596–4601, 2013.
- [53] M. Otte and E. Frazzoli. RRT-X: Real-time motion planning/replanning for environments with unpredictable obstacles. In *Algorithmic Foundations of Robotics XI*, pages 461–478. Springer, 2015.
- [54] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli. Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE Trans. Robot.*, 23(6):1170–1183, 2007.
- [55] S. Patil, J. Van Den Berg, S. Curtis, M. C. Lin, and D. Manocha. Directing crowd simulations using navigation fields. *Trans. on Visualization and Computer Graphics*, 17(2):244–254, 2011.
- [56] M. Prandini and J. Hu. *Stochastic Reachability: Theoretical Foundations and Numerical Approximation*, pages 107–139. Lecture Notes in Control and Information Sciences. Springer Verlag, 2006.
- [57] S. Rodriguez, J.-M. Lien, and N. M. Amato. A framework for planning motion in environments with moving obstacles. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 3309–3314, 2007.
- [58] R. Sharma, J. B. Saunders, and R. W. Beard. Reactive path planning for micro air vehicles using bearing-only measurements. *Journal of Intelligent & Robotic Systems*, 65(1-4):409–416, 2012.
- [59] Q. Song and L. Liu. Mobile robot path planning based on dynamic fuzzy artificial potential field method. *Int. J. of Hybrid Info. Tech.*, 5(4), 2012.
- [60] S. Summers, M. Kamgarpour, J. Lygeros, and C. Tomlin. A stochastic reach-avoid problem with random obstacles. In *Hybrid Systems: Computation and Control (HSCC)*, pages 251–260, 2011.
- [61] S. Summers and J. Lygeros. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Automatica*, 46(12):1951–1961, 2010.
- [62] R. Takei, H. Huang, J. Ding, and C. Tomlin. Time-optimal multi-stage motion planning with guaranteed collision avoidance via an open-loop game formulation. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 323–329, 2012.
- [63] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi. Computational techniques for the verification and control of hybrid systems. *Proceedings of the IEEE*, 91(7):986–1001, 2003.
- [64] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.
- [65] P. Vadakkepat, K. C. Tan, and W. Ming-Liang. Evolutionary artificial potential fields and their application in real time robot path planning. In *Proceedings of Congress on Evolutionary Computation*, pages 256–263, 2000.
- [66] K. P. Valavanis, T. Hebert, R. Kolluru, and N. Tsourveloudis. Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field. *IEEE Trans. Sys., Man, Cybern.*, 30(2):187–196, 2000.
- [67] J. van den Berg, D. Ferguson, and J. Kuffner. Anytime path planning and replanning in dynamic environments. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2366–2371, 2006.
- [68] J. van den Berg, S. J. Guy, J. Snape, M. C. Lin, and D. Manocha. RVO2 library: Reciprocal collision avoidance for real-time multi-agent simulation. <http://gamma.cs.unc.edu/RVO2/>.
- [69] J. van ven Berg, S. J. Guy, M. C. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *International Symposium on Robotics Research*, pages 3–19, 2009.
- [70] C. I. Vasile and C. Belta. Reactive sampling-based temporal logic path planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4310–4315, 2014.
- [71] S. Weijun, M. Rui, and Y. Chongchong. A study on soccer robot path planning with fuzzy artificial potential field. In *Computing, Control and Industrial Engineering (CCIE)*, volume 1, pages 386–390, 2010.



**Nick Malone** is a Ph.D. graduate of the Computer Science Department at the University of New Mexico. He earned a M.S. in Computer Science from the University of Tulsa and a Bachelors in Computer Science at the University of Tulsa.



**Lydia Tapia** received a Ph.D. in Computer Science from Texas A&M University and was a Computing Innovation Postdoctoral Fellow at the University of Texas at Austin. She is an Assistant Professor of Computer Science with the University of New Mexico, Albuquerque, NM, USA. Her research has been recognized through the NSF CAREER Award (2016), the 2015 Denise Denton Emerging Leader Award from the Anita Borg Institute, and the 2017 Computer Research Association Committee on the Status of Women in Computing Research Borg Early Career Award. She is a former Sloan Scholar and NIH Molecular Biophysics Training Grant Fellow.



**Hao-Tien (Lewis) Chiang** received a M.S. in Physics from the University of Mexico and worked on quantum algorithms. He is currently a Ph.D student in the Computer Science Department of the University of New Mexico. Since joining the program, he was a student organizer of the Becoming a Robot Guru 2 workshop at RSS 2016. His research focuses on reinforcement learning and motion planning, specifically for high dimensional and dynamically constrained robots in highly dynamic environments.



**Kendra Lesser** received her Ph.D. in Electrical Engineering in 2014, and an M.S. in Mathematics in 2010 from the University of New Mexico. She received her B.Sc. in Mathematics from McGill University in 2007. She is currently a Senior Controls Engineer at Verus Research in Albuquerque, NM, and prior to that was a Marie Curie Research Fellow in the Computer Science Department at the University of Oxford. Her main research interests include stochastic optimization, planning under uncertainty, verification, and learning algorithms, with

a focus on safe, provably correct planning and estimation. Applications include optimized maintenance and control of building heating systems, motion planning algorithms, and guidance and control of spacecraft.



**Meeko Oishi** received the Ph.D. (2004) and M.S. (2000) in Mechanical Engineering from Stanford University (Ph.D. minor, Electrical Engineering), and a B.S.E. in Mechanical Engineering from Princeton University (1998). She is an Associate Professor of Electrical and Computer Engineering at the University of New Mexico. Her research interests include hybrid dynamical systems, control of human-in-the-loop systems, reachability analysis, and modeling of motor performance and control in Parkinson's disease. She previously held a faculty position at the University of British Columbia at Vancouver, and postdoctoral positions at Sandia National Laboratories and at the National Ecological Observatory Network. She is the recipient of the UNM Regents' Lectureship, the NSF CAREER Award, the UNM Teaching Fellowship, the Peter Wall Institute Early Career Scholar Award, the Truman Postdoctoral Fellowship in National Security Science and Engineering, and the George Bienkowski Memorial Prize, Princeton University. She was a Summer Faculty Fellow at AFRL Space Vehicles Directorate, and a Science and Technology Policy Fellow at The National Academies.