Autonomous Robots (2019) 43:575–588
https://doi.org/10.1007/s10514-018-9739-0

Scalable Markov chain approximation for a safe intercept navigation in the presence of multiple vehicles

Alexey A. Munishkin[1] · Araz Hashemi[2] · David W. Casbeer[2] · Dejan Milutinović[1]

Received: 15 February 2017 / Accepted: 2 April 2018 / Published online: 9 April 2018

Abstract
This paper studies a safe intercept navigation which accounts for the uncertainty of other vehicles' trajectories, avoids collisions and any other positions in which vehicle safety is compromised. Since the number of vehicles can vary with time, it is important that the navigation strategy can quickly adjust to the current number of vehicles, i.e, that it scales well with the number of vehicles. The scalable strategy is based on a stochastic optimal control problem formulation of safe navigation in the presence of a single vehicle, denoted as the one-on-one vehicle problem. It is shown that safe navigation in the presence of multiple vehicles can be solved exactly as an auxiliary Markov decision problem. This allows us to approximate the solution based on the one-on-one vehicle optimal control solution and achieve scalable navigation. Our work is illustrated by a numerical example of safely navigating a vehicle in the presence of four other vehicles and by a robot experiment.

Keywords Autonomous navigation · Dubins vehicles · Stochastic optimal control

1 Introduction

In the context of this paper, safe navigation is the one providing that a vehicle that moves along its path (a) avoids collisions with other vehicles; (b) performs maneuvers to reach a safe configuration with regard to other vehicles; and (c) navigates while taking into account uncertainties in the trajectories of other vehicles.

To study the concept of safe navigation and propose a design method, we introduce in this paper an unmanned aerial vehicle scenario in which a single fixed wing vehicle is tasked to intercept one of multiple other vehicles, avoid collisions and any unsafe positions from which other vehicles can enter its regions of vulnerability. This scenario emphasizes that safety is not only about avoiding collisions, for example, in a car traffic case, the presence of a vehicle in the car driver's blind spot does not immediately lead to a collision, but creates a threat for the car safety, and for the safety reasons, the car driver would try to avoid such configurations. Our multiple-aerial vehicle scenario is related to work on collision-free navigation of mobile robots in complex cluttered environments (Hoy et al. 2015), i.e., collision avoidance in multi-robot and swarm like systems (Alonso-Mora et al. 2013; Panagou et al. 2016; Wang et al. 2017), but it accounts as well for safety consideration that is beyond the risk of collision only.

The study of navigation against threats created by other vehicles is tightly interwoven with the development of game theory (Isaacs 1965) and the two-target game problem (Ardema et al. 1985; Getz and Leitmann 1979; Getz and Pachter 1981). The game includes two vehicles that navigate around each other until one of the vehicles, the winner of the game, enters its target set. A stochastic variant of such two-target games is considered in Yavin (1988) and Yavin and Villers (1988). These and other earlier theoretical works have been surveyed in Grimm and Well (1991). Other lines of work have been focused on applications (Eklund et al. 2005), as well as various other approaches to the problem (Israelsen et al. 2017; McGrew et al. 2010; Virtanen et al. 2006).

In all these works, the two vehicles are opponents with the intent to harm the other vehicle and win the game. Obviously, the safe navigation inspired by the game theory would

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10514-018-9739-0) contains supplementary material, which is available to authorized users.

✉ Dejan Milutinović
dmilutin@ucsc.edu

[1] Computer Engineering, University of California, Santa Cruz, CA 95064, USA

[2] Control Science Center of Excellence, Air Force Research Laboratory, Wright-Patterson AFB, Dayton, OH 45433, USA

essentially mean that every vehicle in the surrounding is considered as an adversary, which is the worst case scenario, and it would result in conservative navigation strategies. Therefore, in this paper we propose a stochastic approach to safe navigation.

In the proposed stochastic approach, we anticipate that vehicles in the surroundings of a safely navigated vehicle *may, or may not* have bad intents. The lack of information about their navigation is modeled by a stochastic process and safety is addressed by a computationally defined avoidance set. In the deterministic problems (Huang et al. 2015) the boundaries between reachable and unreachable state space regions are sharp and avoidance set can be computed using the deterministic optimal control approach. However, in the stochastic problems, reachable and unreachable regions are defined in terms of probability and without sharp boundaries. In our preliminary work (Munishkin et al. 2016), we introduced the concept of avoidance set based on expected times. *The novelty of the work presented here is in an iterative algorithm for the avoidance set computations. Then we use the avoidance set to compute the solution for a safe vehicle navigation in the proximity of a single vehicle and propose an approach to apply this one-on-one vehicle safe navigation to the multiple vehicle case, which scales well with the number of vehicles, and is therefore suitable for real time applications.*

Extending the game theory or optimal control solution to multiple agents is difficult because of the so-called curse-of-dimensionality, due to the number of agents. In the discrete domain, there has been extensive work in solving game theory problems on graphs (Aigner and Fromme 1984), which are traditionally called cops and robbers, and in Vieira et al. (2009) a scalable solution of the game was proposed. For the continuous domain, Vidal et al. (2002) provides a framework for combining the kinematic models of various pursuing agents for a real-time implementation of a chase and search problem. In Li et al. (2008), a hierarchical game extension with a finite time look ahead to the stochastic setting has been proposed, while in Festa and Vinter (2016), a game theory problem is partitioned into smaller problems that are then solved separately, and the solution of the original problem is determined as the lower bound of the smaller problems. However, none of these works considered the two target problem in a multi-vehicle scenario.

Our attempt to deal with multi-agent, two-target problems and safe navigation is based on the stochastic optimal control solution of the one-on-one vehicle problem, which is to some extent similar to Panagou et al. (2016), Wang et al. (2017) and other works that use Lyapunov functions to achieve a collision free motion. Instead of guessing a suitable Lyapunov function, in our approach (Anderson and Milutinović 2011, 2014) we numerically compute it as the value function resulting from the solution of stochastic optimal control, which is tightly connected with the nonholonomic kinematics of the vehicles, as well as anticipated uncertainties. In our preliminary work along this line in Hashemi et al. (2016), we found that the value function of the one-on-one vehicle problem can be represented as a sum of the expected time and hazard function components. In the work presented here, the computed components are integrated in the navigation to replicate the performance of the one-on-one vehicle solution when the safely navigated vehicle is close to its goal and the position of other vehicles is irrelevant.

The paper is organized as follows. Section 2 discusses the problem we are solving with multiple Dubins vehicles. In Sect. 3 we present the iterative algorithm for computing the avoidance set and develop an optimal control strategy for the Blue agent to navigate in the presence of a single Red agent. Then, we extend in Sect. 4 the problem to multiple Red agents and a single faster Blue agent, which navigates to enter the tail sector of one of the Red agents while avoiding positions from which any of the Red agents can enter its tail. After our discussion on the optimal control, we discuss a method of simplifying the computations to provide scalability and preserving optimality in the limiting case of a single Blue chasing a single Red agent. Our results are illustrated by a simulation in Sect. 5, and a robot experiment in Sect. 6. Section 7 gives conclusions.

2 Problem formulation

Consider a scenario with a single blue agent B and N red agents $R_1, R_2, \ldots R_N$, as depicted in Fig. 1. *Our goal is to design a control policy for B, which will allow it to navigate into the tail (vulnerable position) of one of the red agents as quickly as possible while simultaneously avoiding collisions with the red agents.*

Agent B has Dubins vehicle kinematics described by

$$dx_B = v_B \cos\theta_B dt \quad (1)$$

$$dy_B = v_B \sin\theta_B dt \quad (2)$$

$$d\theta_B = u_B dt \quad (3)$$

where (x_B, y_B) is the vehicle's position, v_B is the velocity and θ_B is the heading angle. The control variable $u_B \in \mathcal{U}$ is the heading rate which is bounded and takes the value in the set $\mathcal{U} = [-u_{max}, u_{max}]$.

B is cognizant of its own state (x_B, y_B, θ_B), as well as each red agent's position and heading angle $(x_{R_i}, y_{R_i}, \theta_{R_i})$, $i = 1 \ldots N$. However, the intent of the red agents is unknown to B. To account for this uncertainty each red agent, R_i, is modeled as a Dubins vehicle with stochastic heading rate as follows:

$$dx_{R_i} = v_R \cos\theta_{R_i} dt \quad (4)$$

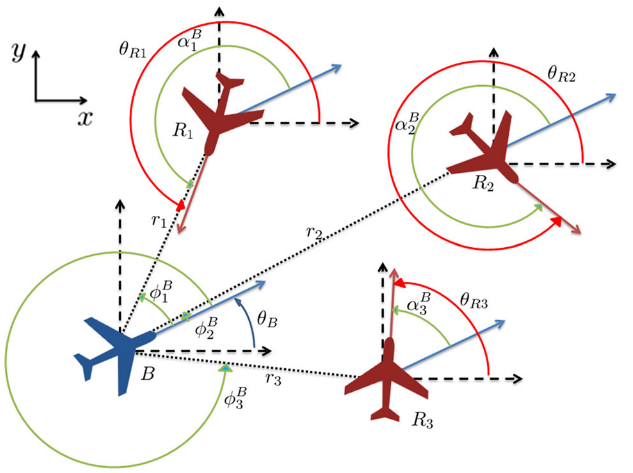

Fig. 1 Geometry of the multi vehicle tail chase problem. θ_B, θ_{R_i} are the heading angles of the blue (faster) agent B and the red (slower) agents R_i. The relative coordinates are the distance r_i^B, the alignment angle $\alpha_i^B = \theta_{R_i} - \theta_B$ and the bearing angle $\phi_i^B = \psi_i^B - \theta_B$ (Color figure online)

$$dy_{R_i} = v_R \sin\theta_{R_i} dt \tag{5}$$

$$d\theta_{R_i} = \sigma_R dw_i \tag{6}$$

where v_R is the velocity ($v_R < v_B$), σ_R is the scaling parameter, and dw_i, $i = 1 \ldots N$, are standard, unit intensity, mutually independent Wiener processes. Hence, for an infinitesimal time-step dt, R_i's change in heading $\theta_{R_i}(t+dt) - \theta_{R_i}(t)$ has a normal distribution with a mean zero and variance $\sigma_R^2 dt$. Therefore, the parameter σ_R describes the agility of the vehicle.

The objective of B is more easily expressed mathematically using the relative coordinates $\mathbf{X}_i = (r_i^B, \phi_i^B, \alpha_i^B)^T$ for each R_i, where r_i^B is the distance, ϕ_i^B is the bearing angle and α_i^B is the alignment angle as depicted in Fig. 1. These relative coordinates are given by

$$r_i^B = \sqrt{(x_{R_i} - x_B)^2 + (y_{R_i} - y_B)^2} \tag{7}$$

$$\phi_i^B = \psi_i^B - \theta_B \tag{8}$$

$$\alpha_i^B = \theta_{R_i} - \theta_B \tag{9}$$

By applying the Itô's Lemma to the relative coordinates (7)–(8) and kinematic models (1)–(6), we obtain the dynamics of the relative coordinates as

$$dr_i^B = b_{r_i} dt; \quad d\phi_i^B = b_{\phi_i} dt; \quad d\alpha_i^B = b_{\alpha_i} dt + \sigma_R dw_i \tag{10}$$

where

$$b_{r_i} = v_R \cos(\phi_i^B - \alpha_i^B) - v_B \cos\phi_i^B; \quad b_{\alpha_i} = -u_B \tag{11}$$

$$b_{\phi_i} = -u_B + \frac{1}{r_i^B}(v_B \sin\phi_i^B - v_R \sin(\phi_i^B - \alpha_i^B)). \tag{12}$$

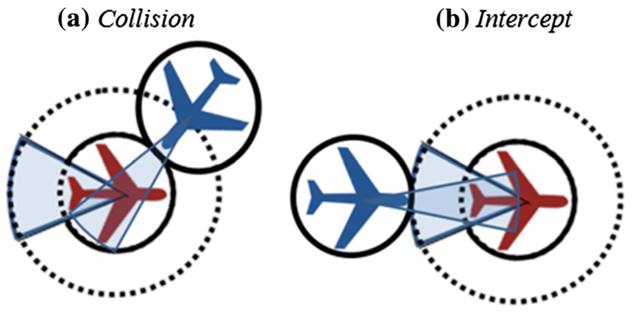
(a) *Collision* **(b)** *Intercept*

Fig. 2 The tail sector of the red vehicle is the target tail sector of B (angular width ϕ). The sector in front of B (angular width α) indicates the range in which B and the red vehicle should be aligned. **a** A collision occurs when B reaches the proximity of the red vehicle (thick arc line). **b** An intercept occurs when B is in the tail sector of the red vehicle and aligned with the red vehicle (Color figure online)

Using the vector of relative coordinates $\mathbf{X}_i$ and $\mathbf{b}(\mathbf{X}_i, u_B) = (b_{r_i}, b_{\phi_i}, b_{\alpha_i})^T$, we can re-write (10) in the vector form as

$$d\mathbf{X}_i = \mathbf{b}(\mathbf{X}_i, u_B)dt + (0, 0, \sigma_R)^T dw_i. \tag{13}$$

With respect to each agent R_i, the goal of B is to reach the *target set*

$$\begin{aligned} \mathcal{T}_i = \big\{\mathbf{X}_i : \mathbf{X}_i = (r_i^B, \phi_i^B, \alpha_i^B)^T, r_i^B \in (\underline{r}, \bar{r}], \\ \phi_i^B \in (-\bar{\phi}, \bar{\phi}], \alpha_i^B \in (-\bar{\alpha}, \bar{\alpha}]\big\}, \end{aligned} \tag{14}$$

for some $\underline{r}, \bar{\phi}, \bar{\alpha} > 0$ while avoiding the collision set

$$\mathcal{C}_i = \{\mathbf{X}_i : \mathbf{X}_i = (r_i^B, \phi_i^B, \alpha_i^B)^T, r_i^B \le \underline{r}\}. \tag{15}$$

The target set $\mathcal{T}_i$ requires not only that B is in the tail sector of R_i at the distance r_i^B between the *collision distance* $\underline{r}$ and the "length" of the tail $\bar{r}$, but also that its heading θ_B is aligned in the general direction with R_i's heading θ_{R_i}, as described by the relative heading bounds $\underline{\alpha}$ and $\bar{\alpha}$. The collision set $\mathcal{C}_i$ describes a ball about R_i which B must avoid. A depiction of these geometries is shown in Fig. 2.

Since the intent of R_i is unknown, and it *may, or may not* have the intent to enter the tail (vulnerable position) of B, it is also reasonable for B to avoid configurations which facilitate R_i reaching its tail. Any such configuration would be considered unsafe by B. We define the set of unsafe configurations $\bar{\mathcal{S}}_i$ with respect to R_i as the set of relative coordinates from which the expected time T_{B_i} for B to enter the tail of R_i is longer or equal to the expected time T_{R_i} for R_i to enter the tail of B, i.e., $\bar{\mathcal{S}}_i = \{\mathbf{X}_i : T_{B_i} \ge T_{R_i}\}$. Therefore, with regard to the single R_i, the set that B should avoid, i.e., the *avoidance* set, is $\mathcal{A}_i = \mathcal{C}_i \cup \bar{\mathcal{S}}_i$.

Let us define the set $\mathcal{G}_i \subset \mathbb{R}^3$ of B's relative coordinates to the red agent R_i, which excludes interiors of the target $\mathcal{T}_i$

and avoidance set $\mathcal{A}_i$, i.e., $\mathrm{int}(\mathcal{A}_i)$ and $\mathrm{int}(\mathcal{T}_i)$, respectively. In other words,

$$\mathcal{G}_i = (\mathbb{R}_+ \times [-\pi, \pi) \times [-\pi, \pi)) \setminus (\mathrm{int}(\mathcal{A}_i) \cup \mathrm{int}(\mathcal{T}_i)) \quad (16)$$

Let τ_i be the time when B first enters the avoidance set $\mathcal{A}_i$ or target set $\mathcal{T}_i$. In other words, $\tau_i = \inf\{t : \mathbf{X}_i(t) \in \partial\mathcal{G}_i\}$ is the time when B reaches the boundary $\partial\mathcal{G}_i$.

For any admissible pure Markov control policy u_B and any initial condition $\mathbf{X}_i(0) \in \mathrm{int}(\mathcal{G}_i)$, the drift and diffusion terms of (13) are Lipschitz for $\mathbf{X}_i \in \mathrm{int}(\mathcal{G}_i)$ and thus (13) has a unique solution $\mathbf{X}_i(t) \in \mathrm{int}(\mathcal{G}_i)$ (in the sense of distribution) until the time point τ_i when we can assume that the process stops; therefore, τ_i is also called the *stopping time*. The generator corresponding to the relative dynamics (13) for the control policy u_B is given by the differential operator

$$\mathcal{L}^{u_B} f(\mathbf{X}_i) = \mathbf{b}(\mathbf{X}_i, u_B)^T (\nabla_{\mathbf{X}_i} f) + \frac{\sigma_R^2}{2} \frac{\partial^2 f}{(\partial [\mathbf{X}_i]_3)^2} \quad (17)$$

where $[\cdot]_3$ denotes the third component of $\mathbf{X}_i$, which is α_i, $f : \mathcal{G}_i \to \mathbb{R}$ and $f \in C^2$ in its domain $\mathcal{G}_i$.

The domain $\mathcal{G} \subset \mathbb{R}^{3N}$ of the control policy u_B, which takes into account all red agents, is $\mathcal{G} = \prod_{i=1}^N \mathcal{G}_i$. The configuration of all agents at a time t is defined by a vector $X(t) \in \mathcal{G}$ as $\mathbf{X}(t) = (\mathbf{X}_1(t)^T, \mathbf{X}_2(t)^T, \ldots \mathbf{X}_N^T(t))^T, \mathbf{X}_i \in \mathcal{G}_i$. If we introduce

$$\mathbf{b}(\mathbf{X}, u_B) = (\mathbf{b}(\mathbf{X}_1, u_B)^T, \ldots \mathbf{b}(\mathbf{X}_N, u_B)^T)^T \quad (18)$$

then the dynamics of $\mathbf{X}$ is

$$d\mathbf{X} = \mathbf{b}(\mathbf{X}, u_B)dt + \sum_{i=1}^N (\mathbf{e}_i \otimes (0, 0, \sigma_R)^T) dw_i \quad (19)$$

where $\mathbf{e}_i = (0, 0, \ldots 1, \ldots 0)^T$ is the N-dimensional standard basis vector with zeros everywhere, except 1 in the ith component. The symbol $\otimes$ represents the Kronecker matrix product, and the result of the operation in the brackets between the two-column vectors of the dimensions $N \times 1$ and 3×1, respectively, is the column vector of the dimension $3N \times 1$. The generator corresponding to the dynamics (19) and control policy u_B is given by the differential operator

$$\mathcal{L}^{u_B} f(\mathbf{X}) = \sum_{i=1}^N \mathcal{L}_i^{u_B} f(\mathbf{X}) \quad (20)$$

where $f : \mathcal{G} \to \mathbb{R}$ and $f \in C^2$ in its domain $\mathcal{G} \subset \mathbb{R}^{3N}$.

Using the relative dynamics (19), we proceed to define a stochastic optimal control problem as in Fleming and Rishel (1975) and Kushner and Dupuis (2001), for any admissible, pure Markov control policy u_B and any initial condition $\mathbf{X} \in \mathcal{G}$. The time at which B enters for the first time any avoidance $\mathcal{A}_i$ or target set $\mathcal{T}_i$, $i = 1, \ldots N$, i.e., when it reaches the boundary $\partial\mathcal{G}$ of $\mathcal{G}$, is $\tau = min(\tau_1, \ldots, \tau_N) = \inf\{t : \mathbf{X}(t) \in \partial\mathcal{G}\}$. Therefore, our goal is to define the control policy $u_B(\mathbf{X}) : \mathcal{G} \to [-u_{max}, u_{max}]$ that minimizes the cost function

$$W(\mathbf{X}, u_B) = \mathbb{E}_{\mathbf{X}}^{u_B} \left\{ g(\mathbf{X}(\tau)) + \int_0^\tau dt \right\} \quad (21)$$

where $\mathbb{E}_{\mathbf{X}}^{u_B}\{\cdot\}$ is the expectation operator with respect to the probability distribution of the realization of the trajectory $\mathbf{X}(t)$ which starts at $\mathbf{X}(0) = \mathbf{X} \in \mathrm{int}(\mathcal{G})$ and terminates in $\mathbf{X}(\tau) \in \partial\mathcal{G}$ under the control u_B, and the terminal cost

$$g(\mathbf{X}) = \begin{cases} M, \text{ if } \ \mathbf{X} \text{ is such that any of } \mathbf{X}_i \in \mathcal{A}_i \\ 0, \ \text{ otherwise} \end{cases} \quad (22)$$

where $M \gg 0$ is a large constant which applies a penalty for entering the boundary $\partial\mathcal{G}$ with any of $\mathbf{X}_i$ in the avoidance set $\mathcal{A}_i$, i.e., outside the target set $\mathcal{T}_i$. The optimal control policy u_B, which minimizes (21) for any initial condition $\mathbf{X} \in \mathrm{int}(\mathcal{G})$, results in the optimal cost-to-go function $V(\mathbf{X}) = \inf_{u_B} W(\mathbf{X}, u_B)$, which is the solution of the dynamic programming (HJB) equation

$$\begin{cases} \inf_{u_B}\{\mathcal{L}^{u_B} V(\mathbf{X}) + 1 = 0\}, & \mathbf{X} \in \mathrm{int}(\mathcal{G}) \\ V(\mathbf{X}) = g(\mathbf{X}), & \mathbf{X} \in \partial\mathcal{G} \end{cases} \quad (23)$$

While the HJB solution defines theoretically the optimal control u_B, finding its solution is practically difficult and we need to resort to approximate computational methods (Kushner and Dupuis 2001). The approach we propose here is inspired by the form of (20), whose structure is a sum of $\mathcal{L}_i^{u_B}$. Therefore, we propose to solve a one-on-one problem involving B and a single red agent ($N = 1$), and use the solution to approximate the solution of (23) for multiple red vehicles ($N > 1$).

3 Intercept of a single vehicle with avoidance of unsafe configurations (one-on-one solution)

In this section $N = 1$; therefore, without loosing generality, we set $i = 1$. To numerically solve the single-target stochastic tail chase problem, we utilize the locally consistent Markov chain approximation method (Kushner and Dupuis 2001). In the first part of this section, we briefly describe this method.

3.1 Locally consistent Markov chain approximation method

We discretize the continuous domain $\mathcal{G}_i \subset \mathcal{R}^3$ with small discrete steps Δr, $\Delta\phi$, $\Delta\alpha$ for each component of $\mathbf{X}_i \in \mathcal{G}_i$. With this we obtain a discrete state space, denoted by $\mathcal{G}_i^h$, which defines the states of the discrete time Markov chain approximation $\mathbf{X}_i^h(n)$, where n is the index of discrete steps. The locally consistent Markov chain approximation $\mathbf{X}_i^h(n)$ of $\mathbf{X}_i(t)$ requires that increments $\Delta\mathbf{X}_i^h(n) = \mathbf{X}_i^h(n+1) - \mathbf{X}_i^h(n)$ satisfy

$$\mathbb{E}\left\{\Delta\mathbf{X}_i^h(n)\right\} = \mathbf{b}(\mathbf{X}_i^h, u_B)\Delta t^h + o(\Delta t^h) \tag{24}$$

$$Cov\left\{\Delta\mathbf{X}_i^h(n)\right\} = \mathbf{a}(\mathbf{X}_i^h)\Delta t^h + o(\Delta t^h) \tag{25}$$

where the discrete steps are separated by interpolation times Δt^h, $\mathbb{E}\{\cdot\}$ represents the conditional expectation given the discretization steps, control action u_B and state $\mathbf{X}_i^h$, $Cov\{\mathbf{X}\} = \mathbb{E}\{\mathbf{X}\mathbf{X}^T\}$, and the matrix $\mathbf{a}(\mathbf{X}_i^h) = diag(0, 0, \sigma_R^2)$.

By defining the transition probabilities, $p^h(\mathbf{X}_i^h(n+1)| \mathbf{X}_i^h(n), u_B, \Delta t^h)$, in such a way that the discrete chain $\mathbf{X}_i^h(n)$ is locally consistent (Grimm and Well 1991, Sec. 4.1) with the original process $\mathbf{X}_i(t)$, it can be shown that the optimal cost and control of the discrete problem, which satisfy the discrete dynamic programming equation,

$$V_i^h(\mathbf{X}_i^h) = \min_{u_B \in \mathcal{U}} \left\{ \sum_{\mathbf{Y}_i^h \in \mathcal{N}(\mathbf{X}_i^h)} p^h(\mathbf{Y}_i^h|\mathbf{X}_i^h, u_B) V_i^h(\mathbf{Y}_i^h) + \Delta t^h \right\} \tag{26}$$

with boundary condition

$$V_i(\mathbf{X}_i^h) = g(\mathbf{X}_i^h), \quad \mathbf{X}_i^h \in \partial\mathcal{G}_i^h \tag{27}$$

converge, as $\Delta r, \Delta\phi, \Delta\alpha \to 0$, with the known rate of convergence (Song and Yin 2010) to the optimal cost and control of the continuous optimal control problem, given from the solution of (23).

In expression (26), $\mathcal{N}^h(\mathbf{X}_i^h)$ denotes the set of six neighbor states $\mathcal{N}^h(\mathbf{X}_i^h) = \{(\mathbf{X}_i^h \pm (\Delta r, 0, 0)^T), (\mathbf{X}_i^h \pm (0, \Delta\phi, 0))^T, (\mathbf{X}_i^h \pm (0, 0, \Delta\alpha))^T\}$ which are the only possible transition states. To abbreviate notation, we write the two transition probabilities along the r_i component as $p_r^{\pm}(\mathbf{X}_i, u_B) = p^h(\mathbf{X}_i^h \pm (\Delta r, 0, 0)^T|\mathbf{X}_i^h, u_B)$ and we do it similarly for $p_\phi^{\pm}(\mathbf{X}_i^h, u_B)$ and $p_\alpha^{\pm}(\mathbf{X}_i^h, u_B)$. Based on the locally consistent Markov chain approximation, the transition probabilities are

$$p_r^{\pm}(\mathbf{X}_i^h, u_B) = t^h(\mathbf{X}_i, u_B) b_r^{\pm}(\mathbf{X}_i)/\Delta r \tag{28}$$

$$p_\phi^{\pm}(\mathbf{X}_i^h, u_B) = t^h(\mathbf{X}_i, u_B) b_\phi^{\pm}(\mathbf{X}_i, u_B)/\Delta\phi \tag{29}$$

$$p_\alpha^{\pm}(\mathbf{X}_i^h, u_B) = t^h(\mathbf{X}_i, u_B) b_\alpha^{\pm}(u_B)/\Delta\alpha + \sigma_R^2/(\Delta\alpha)^2 \tag{30}$$

where $b_r^{\pm}(\mathbf{X}_i) = \max\{0, \pm b_r(\mathbf{X}_i)\}$ is used with the '+' sign for the step Δr and with the '−' sign for the step $-\Delta r$. The values $b_\phi^{\pm}(\mathbf{X}_i, u_B)$ and $b_\alpha^{\pm}(u_B)$ are defined in the same way. The transition probabilities are based on the state and control dependent interpolation time

$$\Delta t^h(\mathbf{X}_i, u_B) = \left(\frac{|b_{r_i}(\mathbf{X}_i)|}{\Delta r} + \frac{|b_{\phi_i}(\mathbf{X}_i, u_B)|}{\Delta\phi} + \frac{|b_{\alpha_i}(u_B)|}{\Delta\alpha} + \frac{\sigma_R^2}{(\Delta\alpha)^2} \right)^{-1} \tag{31}$$

Note that the discretization scheme is based on fixed steps Δr, $\Delta\phi$, $\Delta\alpha$, while the interpolation interval is defined by the problem parameters; therefore, this type of discretization is called *time implicit discretization* (Kushner and Dupuis 2001).

Using the discrete approximation (26)–(31), and the so-called value iterations, we can obtain an approximate solution of (23) for a single target. For brevity, we will denote this numerical method as

$$(u_B^h, V_i^h) \leftarrow HJBSolution(v_B, v_R, \sigma_R, \mathcal{T}_1, \mathcal{A}_i, M) \tag{32}$$

where the superscript h indicates that the results of the computation are in the form of a lookup table corresponding to the discrete computational domain and the problem input parameters are listed in the brackets.

The expected time to reach the target, $\mathcal{T}_i$, can be computed (Gardiner 2009) from the backwards Kolmogorov equation,

$$\begin{cases} \mathcal{L}^{u_B} T_{B_i}(\mathbf{X}_i) + 1 = 0\}, & \mathbf{X}_i \in \text{int}(\mathcal{G}_i) \\ T_{B_i}(\mathbf{X}_i) = 0, & \mathbf{X}_i \in \mathcal{T}_i \end{cases} \tag{33}$$

Comparing the above with (23), we see that once the optimal u_B is fixed (is computed), the cost-to-go function $V_i(\mathbf{X}_i)$ and expected time $T_{B_i}(\mathbf{X}_i)$ solve the same PDE with different boundary conditions. Therefore, once we numerically solve (26) to find the optimal cost and policy, we can apply the same discretization and compute the expected time as

$$T_{B_i}^h(\mathbf{X}_i^h) = \min_{u_B \in \mathcal{U}} \left\{ \sum_{\mathbf{Y}_i^h \in \mathcal{N}(X_i^h)} p^h(\mathbf{Y}_i^h|\mathbf{X}_i^h, u_B) T_{B_i}^h(\mathbf{Y}_i^h) + \Delta t^h \right\} \tag{34}$$

$$T_{B_i}(\mathbf{X}_i^h) = 0, \quad \mathbf{X}_i^h \in \partial\mathcal{T}_i^h \tag{35}$$

We denote the numerical method of computing the expected time as

$$T_{B_i}^h \leftarrow BKGSolution(v_B, v_R, \sigma_R, \mathcal{T}_i, \mathcal{A}_i, u_B). \tag{36}$$

3.2 Avoidance of unsafe configurations

Under our problem formulation, the red agent *may*, or *may not* have the goal to enter the tail of B, which is defined in the same way as the tail of the red agent. The threat from the red agent entering the tail of B should be anticipated by the navigation of B. We account for this threat by assuming that if R_i has the goal to enter the tail of B, then it wants to achieve it without colliding with B and in the shortest possible time. Since R_i does not know the navigation strategy of B, the problem of defining its navigation is similar to the one posed in Sect. 3.1, with the roles of R_i and B reversed. In this case, the relative coordinates of B relative to R_i, (r^R, ϕ^R, α^R), are the distance, the bearing angle and alignment angle, respectively, defined from the perspective of R. At any time point, their relation to $(r^B_{R_i}, \phi^B_{R_i}, \alpha^B_{R_i})$ is

$$r^R = r^B_{R_i};\ \ \phi^R = \pi + \phi^B_{R_i} - \alpha^B_{R_i};\ \ \alpha^R = -\alpha^B_{R_i}. \tag{37}$$

For the purpose of computing the control u^R, the dynamics of these relative coordinates have to account that u_R is the control variable, i.e., $d\theta_R = u_R$, and that control u_B is unknown, i.e., $d\theta_B = \sigma_B dw_B$. Based on the fact that these "reversed role" relative dynamics are similar to (10), we can compute the control u_R using the same technique described in Sect. 3. However, we set $\sigma_B = \sigma_R$ stating that the amount of uncertainty that B and R have about each other's control is the same. With the tail of B defined in the same way as the tail of R, the optimal control policy u_R can be computed as

$$(u^h_R, V^h_i) \leftarrow HJBSolution(v_R, v_B, \sigma_R, \mathcal{T}_i, \mathcal{C}_i, M) \tag{38}$$

which is different from (32) only because of the switched order of v_R and v_B. We can also evaluate the expected time to reach the target

$$T^h_R \leftarrow BKGSolution(v_R, v_B, \sigma_R, \mathcal{T}_i, \mathcal{C}_i, u_R) \tag{39}$$

which is the information we need to compute the avoidance set $\mathcal{A}_i$.

The iterative procedure for computing B's avoidance set, which takes into account the threat of hostile red agents, is provided in Fig. 3. The first line computes the initial optimal control u_B taking into account only the collision set $\mathcal{C}_i$; therefore, initially the avoidance set $\mathcal{A}_i = \mathcal{C}_i$. The expressions (38) and (39) are included as the second and the third lines of the algorithm. Inside the *repeat* loop we evaluate the expected time of reaching the tail of R, and any point with $T^h_R < T^h_{B_i}$ is included in the unsafe set $\bar{\mathcal{S}}_i$, which is then included in the avoidance set $\mathcal{A}^k_i$. The avoidance set is updated in each iteration and, after each update, the optimal control is re-computed. The iterations stop once the set of unsafe configurations is empty and the last updated avoidance set is returned as the result.

```
Algorithm AvoidanceSet $(v_B, v_R, \sigma_R, \mathcal{T}_i, \mathcal{C}_i)$
  $(u^0_B, V_i) \leftarrow HJBSolution(v_B, v_R, \sigma_R, \mathcal{T}_i, \mathcal{C}_i, M)$
  $(u_R, V_i) \leftarrow HJBSolution(v_R, v_B, \sigma_R, \mathcal{T}_i, \mathcal{C}_i, M)$
  $T_R \leftarrow BKGSolution(v_R, v_B, \sigma_R, \mathcal{T}_i, u_R)$
  $\mathcal{A}^0_i = \mathcal{C}_i,\ k = 0$
  $repeat:$
      $k = k + 1,\ \bar{\mathcal{S}}_i = \emptyset$
      $T^{k-1}_B \leftarrow BKGSolution(v_B, v_R, \sigma_R, \mathcal{T}_i, u^{k-1}_B)$
      $for\ all:\ (r^h, \alpha^h, \phi^h)$
          $\alpha^h_R = -\alpha^h$
          $\phi^h_R = \pi + \phi^h - \alpha^h$
          $if\ \overline{T}_R(r^h, \alpha^h_R, \phi^h_R) < \overline{T}^{k-1}_B(r^h, \alpha^h, \phi^h)\ then$
              $\bar{\mathcal{S}}_i = \bar{\mathcal{S}}_i \cup (r^h, \alpha^h, \phi^h)$
          $end\ if$
      $end\ for$
      $\mathcal{A}^k_i = \mathcal{A}^{k-1}_i \cup \bar{\mathcal{S}}_i$
      $(u^k_B, V_1) \leftarrow HJBSolution(v_B, v_R, \sigma_R, \mathcal{T}_1, \mathcal{A}^k_i, M)$
  $until\ \bar{\mathcal{S}}_i == \emptyset$
  $\mathcal{A}_i = \mathcal{A}^k_i$
  $return\ \mathcal{A}_i$
```

Fig. 3 Pseudocode for computing the avoidance set $\mathcal{A}_i$, which is a union of the collision $\mathcal{C}_i$ and unsafe configuration $\bar{\mathcal{S}}_i$ sets. The optimal control u^k_B and the avoidance set $\mathcal{A}^k_i$ are updated in each iteration until the set of unsafe configurations $\bar{\mathcal{S}}_i$ is empty

Let us denote with $Card(\cdot)$ the number of elements in a set. Therefore, the iterations stop when $Card(\bar{\mathcal{S}}_i) = 0$ and through the iterations $Card(\mathcal{A}^k_i) = Card(\mathcal{A}^{k-1}_i) + Card(\bar{\mathcal{S}}_i)$, which implies

$$0 \leq Card(\mathcal{A}^{k-1}_i) < Card(\mathcal{A}^k_i) \leq Card(\mathcal{G}_i) \tag{40}$$

The sequence $Card(\mathcal{A}^k_i)$ is monotonically increasing and limited from above. Therefore, only two outcomes are possible. First, the iterations stop for $\mathcal{A}^k_i$ corresponding to the whole space, in which case, we conclude that the control u_B does not exist since B has to avoid the whole space, i.e., $\mathcal{G}_i$. Second, $\mathcal{A}^k_i \subset Card(\mathcal{G}_i)$.

In the second case, the control can be computed using (30) and the corresponding expected time can be evaluated by (34).

3.3 Hazard and expected time

Let us denote the optimal control defined by (23) as $u^*_{B_i}$. Therefore, when $u_B = u^*_{B_i}$, the two partial differential equations (23) and (33) are the same, except for their difference in the boundary conditions. Since the differential operator $\mathcal{L}^{u_B}_i$ is linear, we can define the hazard $H(\mathbf{X}_i)$ as

$$V(\mathbf{X}_i) = H(\mathbf{X}_i) + T_{B_i}(\mathbf{X}_i) \tag{41}$$

where both $V(\mathbf{X}_i)$ and $T_{B_i}(\mathbf{X}_i)$ are the solutions of (23) and (33), respectively, corresponding to the optimal control, i.e., $u_{B_i} = u^*_{B_i}$. If we substitute $V(\mathbf{X}_i)$ from (41) into (23), we obtain that $H(\mathbf{X}_i)$ satisfies the same partial differential equation as (23), i.e., (33) with the boundary condition $H(\mathbf{X}_i) = M$ for $\mathbf{X}_i \in \partial\mathcal{G}_i$. The state X_i dependent probability $\mathbb{P}^{u_{B_i}}(\mathbf{X}_i)$ of reaching the boundary under the feedback control u_{B_i} can be computed as $\mathbb{P}^{u_{B_i}}(\mathbf{X}_i) = \frac{1}{M}H(\mathbf{X}_i)$. This can be verified by the fact that $\mathbb{P}^{u_{B_i}}$ also satisfies (33) with the boundary condition $\mathbb{P}^{u_B}(\mathbf{X}_i) = 1$ for $\mathbf{X}_i \in \partial\mathcal{G}_i$. This short analysis also indicates that the effect of a large M is to reduce the probability of reaching the boundary $\mathcal{G}_i$.

4 Scalable navigation strategy

Theoretically, one can apply the same discretized value iteration scheme used in the previous section to the optimal control with multiple red vehicles ($N > 1$). However, the dimension of the state space $\mathcal{G} \subset \mathbb{R}^{3N}$ is large in the sense that if we discretize each dimension with D_s discrete steps, the number of grid cells over which we have to compute the value iterations is D_s^{3N}. Furthermore, the control computed that way would not be able to address the change in the number of red vehicles. To circumvent this challenge in this section, we propose a control based on the one-on-one solution (see Sect. 3.2) which requires the value iteration computations with only D_s^3 grid cells. To achieve that, we introduce an auxiliary Markov decision problem which is locally consistent with the original problem and derive the form of its value function. Then we use its one-step look-ahead approximation which approximates its value function via lookup tables for the expected time and the hazard of the one-on-one solution. These lookup tables have to be computed only once and stored in the memory, and allow the computation of a scalable navigation strategy.

4.1 Auxiliary Markov decision problem

Instead of dealing with the discretized state space $\mathcal{G}^h$ of $\mathcal{G} \subset \mathbb{R}^{3N}$ and the corresponding locally consistent Markov chain $\mathbf{X}^h(n)$ we could obtain by the discretization from (23), here we deal with the discrete state space $\tilde{\mathcal{G}}^h = \prod_{i=1}^N \mathcal{G}_i^h$, where $i = 1, \ldots N$. The state space is illustrated in Fig. 4 in which each plane represents $\mathcal{G}_i^h$. Let us denote with $p^h(\tilde{\mathbf{X}}^h(n+1)|\tilde{\mathbf{X}}^h(n), u_B)$ the transition probability of the Markov chain $\tilde{\mathbf{X}}(n)$ over the discretized space $\tilde{\mathcal{G}}^h$. Since our model of R_i for the control B design is (4)–(6), i.e., the motion of the red agents R_i is independent, we conclude that

$$p^h(\tilde{\mathbf{X}}^h(n+1)|\tilde{\mathbf{X}}^h(n), u_B) = \prod_{i=1}^N p^h(\mathbf{X}_i^h(n+1)|\mathbf{X}_i^h(n), u_B) \tag{42}$$

where $\tilde{\mathbf{X}}(n) = (\mathbf{X}_1^h(n), \ldots \mathbf{X}_N^h(n))^T$ with component evolutions resulting from the discretization scheme based on (26) and applied for each $i = 1, \ldots N$. Note that every state of a Markov chain $\mathbf{X}^h(n)$ resulting from (23) would have $6N$ adjacent states, 2 along each of $3N$ dimensions of the state space $\mathcal{G}^h$. However, in our case, each state of the auxiliary Markov chain $\tilde{\mathbf{X}}(n)$ has 6^N, 2 along each of 3 relative dimensions in each subspace $\mathcal{G}_i^h$. The reason for constructing the auxiliary chain is that, given a control, the transitions in each subspace are independent of each other. Now, one may question whether the auxiliary Markov chain $\tilde{\mathbf{X}}(n)$ is locally consistent with the original continuous process $\mathbf{X}(t)$. The following theorem resolves this issue.

Theorem 1 *For $\widetilde{\mathbf{X}}^h = (\mathbf{X}_1^h, \mathbf{X}_2^h, \ldots \mathbf{X}_N^h)$ and u_B, define the multi-target interpolation interval $\widetilde{\Delta t}^h$ as*

$$\widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B) = \min_{i=1,\ldots,N} \Delta t^h(\mathbf{X}_i^h, u_B) \tag{43}$$

where $\Delta t^h(\mathbf{X}_i^h, u_B)$ is defined by (31). Then $\widetilde{\mathbf{X}}^h(n)$ is locally consistent with the process $\mathbf{X}(t)$ in (19) so that

$$\begin{aligned} \mathbb{E}\left\{\Delta\widetilde{\mathbf{X}}^h(n)\right\} &= \mathbf{b}(\widetilde{\mathbf{X}}^h(n), u_B)\widetilde{\Delta t}^h + o(\widetilde{\Delta t}^h) \\ \mathbb{C}\text{ov}\left\{\Delta\widetilde{\mathbf{X}}^h(n), \Delta\widetilde{\mathbf{X}}^h(n)\right\} &= \widetilde{\mathbf{a}}\widetilde{\Delta t}^h + o(\widetilde{\Delta t}^h) \end{aligned} \tag{44}$$

where $\Delta\widetilde{\mathbf{X}}(n) = \widetilde{\mathbf{X}}(n) - \widetilde{\mathbf{X}}(n-1)$, $\mathbf{b}$ is defined by (18), $\widetilde{\mathbf{a}} = diag(\mathbf{1}_N \otimes (0, 0, \sigma_R)^T$ and $\mathbf{1}_N$ is the column vector of 1.

Proof Let $\hat{\mathbf{1}}_i \in \mathbb{R}^{3N\times 3}$ be the block-stacked matrix with the 3×3 identity matrix in block i and zeros elsewhere. Then $\mathbb{E}\left\{\Delta\widetilde{\mathbf{X}}^h(n)\right\} = \mathbb{E}\left\{\sum_{i=1}^N \hat{\mathbf{1}}_j \Delta\mathbf{X}_i^h(n)\right\} = \sum_{i=1}^N \hat{\mathbf{1}}_i \mathbb{E}\left\{\Delta\mathbf{X}_i^h(n)\right\}$ since, by construction, the subchain $\mathbf{X}_i^h(n)$ is independent of $\mathbf{X}_j^h(n)$ for $i \neq j$.

Based on Lemma X from Sec. 5.3 of Kushner and Dupuis (2001), we have that $\mathbb{E}\{\Delta\mathbf{X}_i(n)\} = \mathbf{b}(\mathbf{X}_i, u)\ \Delta t^h(\mathbf{X}_i, u)$ where $\Delta t^h(\mathbf{X}_i, u)$ is as in (31). Without losing generality, we will as in Kushner and Dupuis (2001) assume that discretization steps are expressed based on a mesh size h, e.g., Δr, $\Delta\phi$ and $\Delta\alpha$ are expressed as $m_k h$, $k = 1, 2, 3$, respectively. With mesh-size h, and $\mathbf{b}_{\{1,2,3\}}(\mathbf{X}_j, u_B)$ which corresponds to $\mathbf{b}_{\{r_j,\phi_j,\alpha_j\}}(\mathbf{X}_j, u_B)$ we have $Q^h(\mathbf{X}_i, u_B) = \sigma_R^2 + h\sum_{k=1}^3 m_k|b_k(\mathbf{X}_j, u_B)|$, where therefore, $\Delta t^h(\mathbf{X}_i, u_B) = m_3^2 h^2 / Q^h(\mathbf{X}_i, u_B)$ is

$$\Delta t^h(\mathbf{X}_i, u_B) = \frac{m_3^2 h^2/\sigma_R^2}{1 + \frac{h}{\sigma_R^2}\sum_{k=1}^3 m_k|b_k(\mathbf{X}_i, u_B)|}$$

and

$$\widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B) = \frac{m_3^2 h^2/\sigma_R^2}{1 + \frac{h}{\sigma_R^2} \max_j \left\{ \sum_{k=1}^3 m_k |b_k(\mathbf{X}_j, u_B)| \right\}}$$

We can now exploit a general inequality $|\frac{1}{1+x} - \frac{1}{1+y}| \leq |x - y|$, $x, y > 0$ and by the boundedness of $\mathbf{b}(\cdot)$ on the domain $\mathcal{G}_i^h$ obtain

$$|\Delta t^h(\mathbf{X}_i, u_B) - \widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B)| \leq K \frac{m_3^2 h^3}{\sigma_R^4} \tag{45}$$

with

$$K \geq \left| \sum_{k=1}^3 m_k |b_k(\mathbf{X}_i, u_B)| - \max_j \left\{ \sum_{k=1}^3 m_k |b_k(\mathbf{X}_j, u_B)| \right\} \right| \tag{46}$$

It follows that $|\Delta t^h(\mathbf{X}_i, u_B) - \widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B)| / \widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B) = O(h)$, so the difference (45) is $o(\widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B))$ for all i.

Putting the above together, we have

$$\begin{aligned}\mathbb{E}\{\Delta \widetilde{\mathbf{X}}^h(n)\} &= \sum_{i=1}^N \hat{\mathbf{1}}_j \mathbf{b}(\mathbf{X}_j, u_B) \Delta t^h(\mathbf{X}_j, u_B) = \sum_{i=1}^N \hat{\mathbf{1}}_j \Big\{ \\ &\mathbf{b}(\mathbf{X}_j, u_B) \left[\widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B) + \left(\Delta t^h(\mathbf{X}_j, u_B) - \widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B) \right) \right] \Big\} \\ &= \mathbf{b}(\widetilde{\mathbf{X}}, u_B) \widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B) + o\left(\widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B) \right).\end{aligned}$$

The covariance follows in a similar manner. Writing $\overline{\Delta \widetilde{\mathbf{X}}^h}(n) = \Delta \widetilde{\mathbf{X}}^h(n) - \mathbb{E}\{\Delta \widetilde{\mathbf{X}}^h(n)\}$ for the (zero-mean) centered increment, the covariance from the theorem statement is

$$\begin{aligned}\mathbb{E}\{\overline{\Delta \widetilde{\mathbf{X}}^h}(n) \overline{\Delta \widetilde{\mathbf{X}}^h}(n)^T\} &= \mathbb{E} \left\{ \sum_{i=1}^N \sum_{j=1}^N \hat{\mathbf{1}}_i [\overline{\Delta \widetilde{\mathbf{X}}_i}][\overline{\Delta \widetilde{\mathbf{X}}_j}]^T \hat{\mathbf{1}}_j^T \right\} \\ &= \sum_{j=1}^N \hat{\mathbf{1}}_j \left[\mathbf{a}(X_j) \Delta t^h(\mathbf{X}_j, u_B) + o(\Delta t^h(\mathbf{X}_j, u_B)) \right] \hat{\mathbf{1}}_j' \\ &= \tilde{\mathbf{a}}(\widetilde{\mathbf{X}}) \widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B) + o\left(\widetilde{\Delta t}^h(\widetilde{\mathbf{X}}, u_B) \right).\end{aligned}$$

□

From Theorem 1, one can apply the results of Kushner and Dupuis (2001) to show the convergence of the auxiliary MDP to the original continuous-time problem associated with (18)–(21). Consequently, the original problem can be solved in the discrete space $\widetilde{\mathcal{G}}^h$ using

$$\begin{aligned}&\widetilde{V}^h(\widetilde{\mathbf{X}}^h) \\ &= \min_{u_B \in \mathcal{U}} \left\{ \sum_{\widetilde{\mathbf{Y}}^h \in \mathcal{N}(\widetilde{\mathbf{X}}^h)} \widetilde{p}^h(\widetilde{\mathbf{Y}}^h; \widetilde{\mathbf{X}}^h, u_B) \widetilde{V}^h(\widetilde{\mathbf{Y}}^h) + \widetilde{\Delta t}^h \right\}\end{aligned} \tag{47}$$

for $\widetilde{\mathbf{X}} \in \widetilde{\mathcal{G}}^h$ and $\widetilde{V}^h(\widetilde{\mathbf{X}}) = g(\widetilde{\mathbf{X}})$ on $\partial \widetilde{\mathcal{G}}^h$.

4.2 One-step look-ahead cost approximation

While the transition probabilities (42) define a locally consistent chain, solving the discrete HJB (47) for $\widetilde{V}^h(\widetilde{\mathbf{X}})$ and the corresponding optimal control u_B via value iteration still explodes in computation time as the number of targets increases.

Instead, we approximate these values via a one-step look-ahead cost approximation

$$\widehat{V} = \min_{u_B} \left\{ \min_i \left\{ \mathcal{T}_{B_i}(\mathbf{X}_i, u_B) \right\} + \sum_{j=1}^N \mathcal{H}(\mathbf{X}_j, u_B) \right\} \tag{48}$$

composed of the total expected risk with respect to all red vehicles

$$\mathcal{H}(\mathbf{X}_i, u_B) = \sum_{\mathbf{Y}^h \in \mathcal{N}(\mathbf{X}_i)} p^h(\mathbf{Y}^h | \mathbf{X}_i, u_B) H(\mathbf{Y}^h, u_B) \tag{49}$$

and the minimum expected time to the target i

$$\mathcal{T}_{B_i}(\mathbf{X}_i, u_B) = \Delta t_i^h + \sum_{\mathbf{Y}^h \in \mathcal{N}(\mathbf{X}_i)} p^h(\mathbf{Y}^h | \mathbf{X}_i, u_B) \mathcal{T}_{B_i}(\mathbf{Y}^h, u_B)$$

where $\Delta t_i^h = \Delta t^h(\mathbf{X}_i, u_B)$ and $\mathcal{N}(\mathbf{X}_i)$ are the 6^N neighbor states of $\mathbf{X}_i$. Based on this approximation, the control u_B minimizes the expression in the braces of (48). The approximation is scalable because all the values used in the approximations can be pre-computed and stored as lookup tables, and the number of computations linearly increases with the number of agents. Another important property of the approximation is that for $N = 1$, it is not an approximation, but an exact match to the optimal control solution. This method can be thought of as a form of approximate dynamic programming (Powell 2009).

For multiple red vehicles, the approximation includes the sum of expected risks $\mathcal{H}(\mathbf{X}_i, u_B)$. If the second part of the expression were the sum of the expected times, then our approximation could be written in terms of the value functions $V_i^h(\mathbf{X}_i, u_B)$, $i = 1, \ldots N$. However, the sum would include the expected times to the tails of all red agents and farther agents would contribute more to the sum. This is clearly unacceptable; at the state in which B is close to the tail sector of a specific red vehicle, the control should mainly depend on the expected time proximity to the tail sector and not on the expected times to the tails of distant vehicles. Consequently, in our approximation, we use the min operator for the expected times.

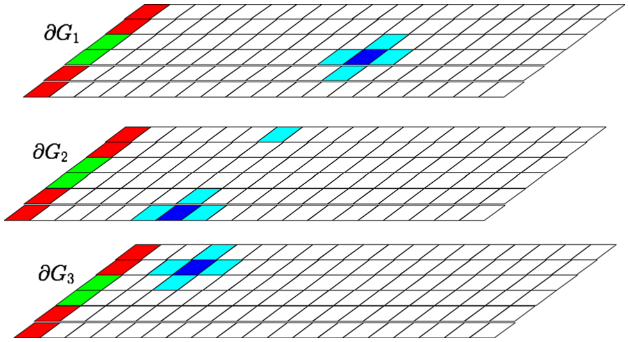

Fig. 4 The discretized auxiliary Markov decision problem state space. Each plane represents a (simplified, $\phi \equiv 0$) domain for the relative dynamics with respect to each boundary $\partial \mathcal{G}_i^h$ with the target set states colored green and the states to be avoided colored red. The relative position between B and R_i is represented in the ith plane and colored blue. The neighbor states in which the relative position can be in the next step are colored light blue (Color figure online)

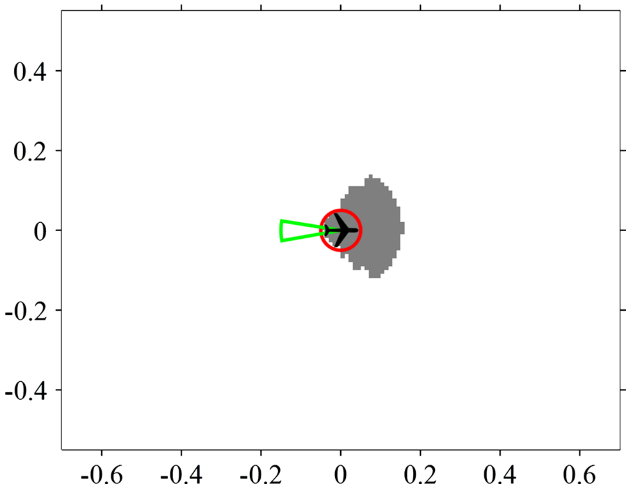

Fig. 5 The avoidance set for $v_B = 0.1$ and $v_R = 0.05$ is depicted by gray points. The center of the map depicts the red agent, its tail sector (green) and circular collision set (red) (Color figure online)

5 Results

The numerical simulation results of this section are based on the Dubins vehicle models (1)–(6) with the velocities $v_B = 0.1$ and $v_R = 0.05$ for B and R, respectively. The navigation uncertainties of R in (6) and of B in (38) are modeled with $\sigma_R = \sigma_B = \sqrt{10\pi/180}$ resulting in a standard deviation of the heading angle change of 10°/s. The target set $\mathcal{T}_i$ in (14) and the collision set $\mathcal{C}_i$ in (15) are based on the parameters $\underline{r} = 0.05$, $\bar{r} = 0.15$, $\bar{\phi} = 10°$ and $\bar{\alpha} = 20°$. The discretization steps in our computations are $\Delta r = (\underline{r} - \bar{r})/100$, $\Delta\phi = \Delta\alpha = 5°$. The M value, which is used in the iterative algorithm (see Fig. 4) for computing the avoidance set, is $M = 10^4$. Since the heading angle of B is the integral of u_B, i.e., $d\theta_B = u_B dt$, in our computations of optimal control for one-on-one solutions, we used the discrete values $u_{B_i} \in \{-0.5, 0, 0.5\}$, $i = 1, 2, 3, 4$ to speed up the computations. In the computation of u_B based on our algorithm, we allowed for values between − 0.5 and 0.5 with the step of 0.1.

To illustrate the avoidance set, we created a 2D map depicted in Fig. 5. The map coordinates are the x and y coordinates of B, relative to R, which is in the center of the map and pointed to the right as depicted in the figure together with the radius $\underline{r}$ and the tail sector in the B's target set. For every point (x, y) and for any heading angle of B at that point, i.e., the closest discrete point, we check if the value function $V_i^k = M$. If it is, then we know that the relative position is in the avoidance set and we mark that $x - y$ coordinate with gray color. Although the map is a projection of the avoidance set, it is likely that most of the positions to be avoided by B are in front of R and that the safest positions for B are behind R.

The avoidance set in Fig. 5 is computed after *two* iterations of the algorithm in Fig. 3. For the purpose of testing, we also computed the avoidance set for the velocity ratio between v_B and v_R that is closer to 1 ($v_B = 0.1$ and $v_R = 0.075$), and found that it required *four* iterations. That confirmed our expectation that the more competitive agents are, the more iterations it takes to compute the avoidance set. Once we reduced the discretization steps to $\Delta\phi = \Delta\alpha = 2°$, the algorithm took *six* iterations until the convergence, which suggests that the size of the discrete space also impacts the number of iterations.

In Fig. 6, we present an example of the simulation resulting from our navigation control. Figure 7 depicts the expected time toward each R agent and the corresponding hazard values. At the beginning, depicted in Fig. 6a, the B agent is in the position in which one of the red agents (R_4, gray) is the one towards which B has the smallest expected time, but it faces the other three red agents and the risk of collision. Therefore, the hazard values for R_1, R_2, R_3 are high in the time interval corresponding to the shaded segment A (see Fig. 7). To reduce the hazard values, the B agent maneuvers around the three R agents (see Fig. 6b), which results in the drop of the hazard values by the end of the time interval covered by the segment A. In the time interval between the shaded segments A and B, the expected time is reduced while the hazard to R_1 moderately increases (see Fig. 7). The position around which the hazard peaks is in Fig. 6b. Now B maneuvers to reduce the hazard, but at the cost of the steep increase of the expected times, which corresponds to the time covered by the shaded segment B. After the segment B, the expected time towards R_2 (green) is the smallest for the first time. It is the moment at which B starts cutting in front of R_1 and R_3. During the time interval covered by the shaded interval C, the expected time to R_2 is the shortest simultaneously with the highest hazard since B has already passed R_3 and is started

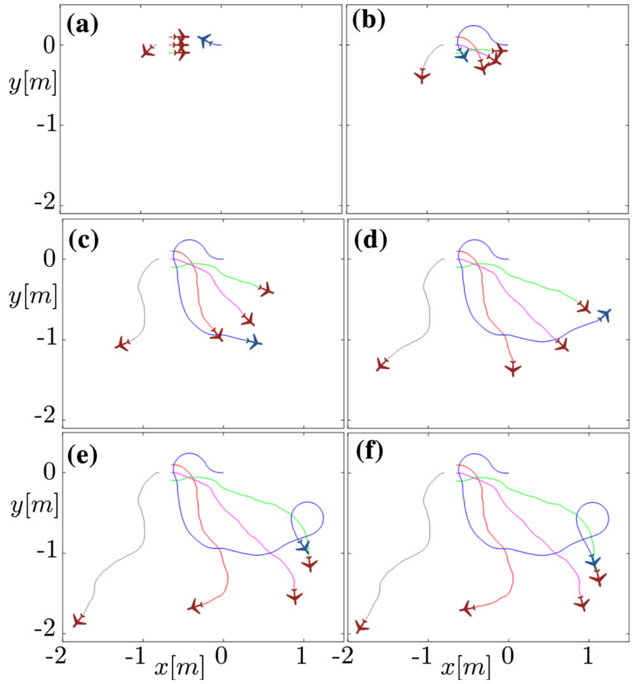

Fig. 6 Simulation result. Each panel corresponds to the following simulation times: **a** 1.32 s; **b** 9.32 s; **c** 25.00 s; **d** 34.00 s; **e** 45.29 s; and **f** 47.73 s, which is also the time at which B enters the tail sector of R_2. The B agent trajectory is colored blue and the R_1–R_4 trajectories are colored red, green, magenta and gray, respectively (see the supplementary movie) (Color figure online)

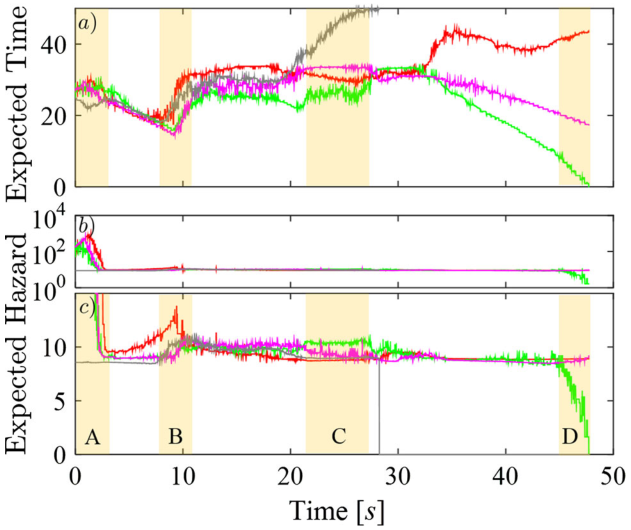

Fig. 7 **a** Expected time, **b** hazard values on the log scale and **c** hazard values on the linear scale. The diagrams corresponding to R_1–R_4 are colored red, green, magenta and gray, respectively (Color figure online)

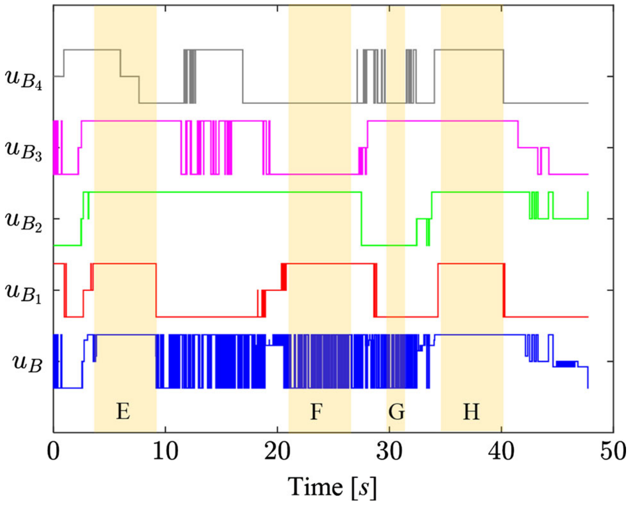

Fig. 8 The control u_B and control variables u_{B_i}, $i = 1, 2, 3, 4$ resulting from the solution of the one-on-one problem and relative positions between B and R_i at every time point of the simulation. All values are in the range $[-0.5, 0.5]$

cutting in front of R_2 as depicted in Fig. 6c. Shortly after the segment C time interval, the agent R_4 is out of range, the expected time for R_4 (gray) is too large to be plotted and the corresponding hazard drops to 0. At that point, the shortest expected time does not correspond to R_2 and it remains like that until the time point towards the middle between the segments C and D, i.e., the position depicted Fig. 6d. In the time interval covered by the segment D, the expected time towards R_2 continues to decrease, but there is also a steep decrease in the corresponding hazard value (see Fig. 6e). At the time point of the intercept depicted in Fig. 6f, both the hazard, as well as the expected time are 0.

The optimal control of B towards each R, which results from the solution of the one-on-one problem, as well as the optimal control u_B used in the navigation in our scenario are plotted in Fig. 8. The values of the control are in the range of minimal and maximal values of the turning rate. As we can see from the diagram, there are time instances when the control u_B looks like as if it were obtained by the control that is optimal towards the majority of R agents. The best examples are time intervals covered by the shaded segments, E and H. In both of these segments $u_{B_1} = u_{B_2} = u_{B_3}$; therefore, independently of the value u_{B_4}, the value u_B is defined by the previous values. Note that in the segment H, all individual u_{B_i}, $i = 1, 2, 3, 4$, are equal and it would be very strange to have u_B with a different value.

In the shaded segment F, we find $u_{B_1} = u_{B_2}$ and $u_{B_3} = u_{B_4}$. Since there is no majority of control values that are the same, the value u_B switches between these two values, or takes some value in between as may be expected by a *majority rule*. A clear example that our control does not yield such a rule is the control in the time interval covered by the shaded segment G. In the segment, $u_{B_1} = u_{B_2} = u_{B_4}$ and the control u_B still switches and takes some value in between the u_{B_1} and u_{B_3} values.

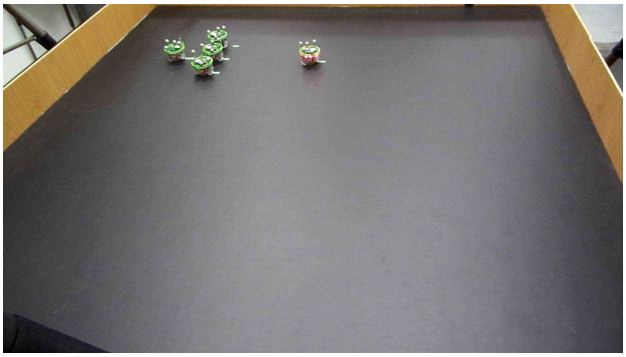

Fig. 9 Robot experiment setup with five e-puck robots. Each robot has a unique configuration of infrared reflecting markers (green rings with silver spheres) tracked by a motion capturing system with four Bonita 10 Vicon cameras. The robots are at their initial position (see the supplementary video) (Color figure online)

6 Robot experiment results

The experimental setup included five e-puck robots with infrared markers which were tracked by a motion capturing system composed of four Bonita 10 Vicon cameras. For each e-puck robot, we implemented proportional–integral (PI) controllers for the velocity and turning rate of the robot. These low-level control loops, software and hardware architecture supporting the experiment have been presented in Munishkin et al. (2016).

Out of five e-puck robots, one had the role of B and four had the roles of R_1–R_4. The velocity of B was set to a constant 10 cm/s. The control for B used motion capturing system measurements to compute relative positions between B and each R_i, $i = 1, 2, \dots 4$. The relative positions were used to read lookup tables for the expected time, hazard and optimal control u_{B_i}, and compute the turning rate u_B. This turning rate was sent to the low-level PI controller that controlled the motion of B.

The four R_1–R_4 robots were controlled to follow trajectories that were similar to those from the simulation. However, in dealing with the experimental setup, we had to adjust the trajectories to avoid collisions among R_1–R_4 due to their size and fit them within the confined space of the experiment, see Fig. 9. As a result, the velocities of R_i, $i = 1, 2, \dots 4$, were set to $v_R = 3.25$ cm/s. Although, this velocity was smaller then the one used in our stochastic control design ($v_R = 5$ m/s), we did not update the design with this value. Our experience in the experiments (Munishkin et al. 2016) and flight tests (Milutinović et al. 2017) is that the type of stochastic optimal control we use does not require a perfect matching of parameters with the reality.

The initial configuration of B and R_i robots, $i = 1, 2, \dots 4$, in the experiment was similar to the one used in the simulation and encircled in Fig. 10. The figure shows the complete trajectories until B enters the tail of R_3. This outcome is different from the simulation and can be explained by a higher hazard with respect to R_2 (green) than to R_3 (magenta) towards the end of the experiment around the time point 40 s. The trajectories that are confined to the dimensions of the experimental setup also result into smaller expected times than in the simulation, specially for R_4 (gray) in Fig. 11.

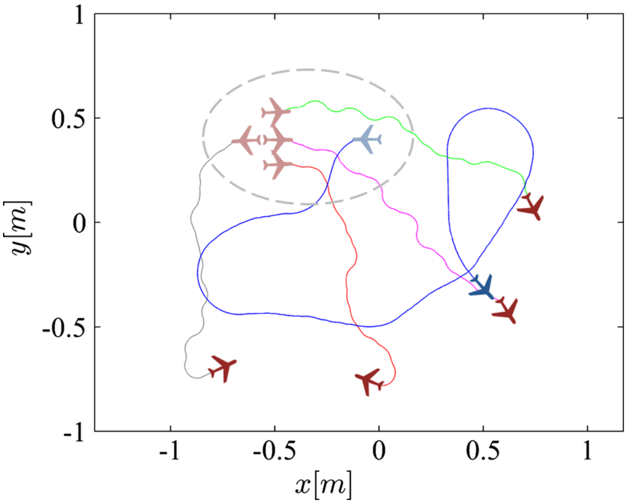

Fig. 10 Robot experiment trajectories. The initial condition and the trajectories of R_1–R_4 are similar to those in the simulation. The B robot trajectory is colored blue and the R_1–R_4 trajectories are colored red, green, magenta and gray, respectively (Color figure online)

Figure 12 depicts the optimal control u_i computed with respect to each R_i, $i = 1, 2, \dots 4$, as well as the control u_B which results from the one-step look-ahead cost approximation (48). This control, just like in the simulation, accounts both for the hazard and expected times to the tail of R_i, $i = 1, 2 \dots 4$, which are defined by the one-on-one solution and plotted in Fig. 11. Any time u_{B_i}, $i = 1, 2 \dots 4$, are the same for all i, the control $u_B = u_{B_i}$. For the time intervals in which at least one of u_{B_i} is different from the others, we can observe that u_B alternates its value. Since the vehicle's heading angle is the integral of u_B, the impact of fast alterations of u_B to the heading angle is smoothed out. The PI controller that controls the turning rate of the vehicle additionally contributes to the smoothing. Towards the end of the experiment, $u_B = u_{B_3}$ since B is close to the tail of R_3 and has to navigate optimally to it.

7 Conclusions

In this paper, we presented an approach to the safe navigation of the nonholonomic fixed wing unmanned aerial vehicle surrounded by multiple other vehicles. The approach is based on the stochastic optimal control which better addresses the uncertainty of the other vehicles' trajectories without assum-

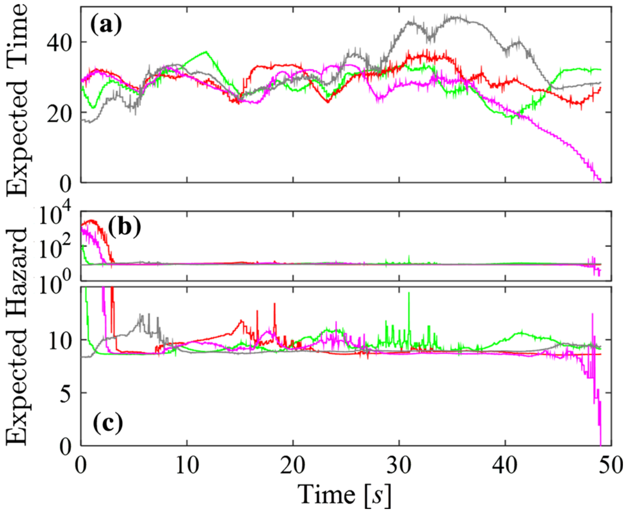

Fig. 11 Robot experiment expected times and hazards. **a** Expected time, **b** hazard values on the log scale and **c** hazard values on the linear scale. The diagrams corresponding to R_1–R_4 are colored red, green, magenta and gray, respectively (Color figure online)

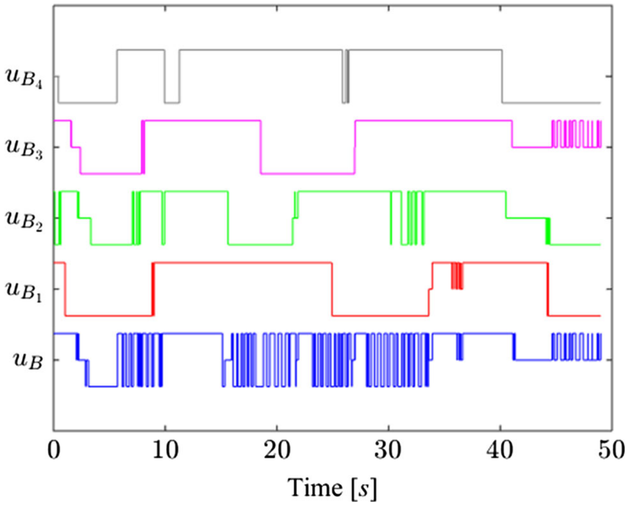

Fig. 12 Robot experiment control. The control u_B and control variables u_{B_i}, $i = 1, 2, 3, 4$, resulting from the solution of the one-on-one problem and relative positions between B and R_i at every time point of the simulation. All values are in the range $[-0.5, 0.5]$ (see the supplementary video)

ing a worst case scenario, which is classically considered by the game theory approaches to similar problems. The scenario that is used in our paper was selected to illustrate that safety is not only about collision avoidance.

The problem of the safe navigation in the presence of multiple vehicles is addressed based on the solution of the one-on-one vehicle problem. Therefore, we first explained the stochastic optimal control for that case and introduced the iterative algorithm for computing the avoidance set. The distinguishing property of our approach is that the avoidance set is computed taking into account the vehicle kinematics.

For the multiple vehicles, we derived the result based on the locally consistent Markov chain approximation. This result serves well to illustrate the complexity of the problem and as the basis for proposing the scalable method for computing the control. The scalable method is based on the expected times and hazards resulting from the one-on-one vehicle stochastic optimal control solution. Our scalable approach to the navigation in the multiple vehicle scenario was illustrated by the numerical simulation. We compared the trajectories of the vehicles with the expected times and hazards from our simulation. In addition, we compared the control action of computed solutions with the control actions resulting from the one-on-one solution.

The presented approach is suitable for real-time implementations, therefore, we presented the results from the experiment with small-scale laboratory e-puck robots. The navigation relies on the measurements of relative positions between B and red vehicles ($R_1 - R_4$) to be intercepted. While the measurements were corrupted by the noise and robot motion model could be more complex, we did not face any significant problem in the implementation.

The approach addresses the uncertainty in the relative positions through the kinematic uncertainty in the motion of red vehicles. The presence of any additional uncertainty would likely require the introduction of an additional stochastic process in the problem formulation. Unless the additional uncertainty in the problem cannot be addressed by an increased intensity of stochasticity in the kinematic model of red vehicles, the problem may be significantly different from the one presented in this paper.

Acknowledgements Funding was provided by U.S. Department of Defense (Grant No. FA8650-15-D-2516).

References

Aigner, M., & Fromme, M. (1984). A game of cops and robbers. *Discrete Applied Mathematics*, *8*(1), 1–12.

Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., & Siegwart, R. (2013). *Optimal reciprocal collision avoidance for multiple non-holonomic robots* (pp. 203–216). Berlin, Heidelberg: Springer.

Anderson, R., & Milutinović, D. (2011). A stochastic approach to dubins feedback control for target tracking. In 2011 IEEE/RSJ international conference on intelligent robots and systems (pp. 3917–3922). https://doi.org/10.1109/IROS.2011.6094760.

Anderson, R. P., & Milutinović, D. (2014). A stochastic approach to dubins vehicle tracking problems. *IEEE Transactions on Automatic Control*, *59*(10), 2801–2806. https://doi.org/10.1109/TAC.2014.2314224.

Ardema, M. D., Heymann, M., & Rajan, N. (1985). Combat games. *Journal of Optimization Theory and Applications*, *46*(4), 391–398.

Eklund, J., Sprinkle, J., Kim, H., & Sastry, S. (2005). Implementing and testing a nonlinear model predictive tracking controller for aerial pursuit/evasion games on a fixed wing aircraft. In 2005 American control conference (ACC) (Vol. 3, pp. 1509–1514).

Festa, A., & Vinter, R. B. (2016). Decomposition of differential games with multiple targets. *Journal of Optimization Theory and Applications*, *169*, 849–875.

Fleming, W. H., & Rishel, R. W. (1975). *Deterministic and stochastic optimal control*. New York: Springer.

Gardiner, C. (2009). *Stochastic methods: A handbook for the natural and social sciences*. Berlin, Heidelberg: Springer.

Getz, W. M., & Leitmann, G. (1979). Qualitative differential games with two targets. *Journal of Mathematical Analysis and Applications*, *68*, 421–430.

Getz, W. M., & Pachter, M. (1981). Capturability in a two-target "game of two cars". *Journal of Guidance and Control*, *4*(1), 15–22.

Grimm, W., & Well, K. H. (1991). Modelling air combat as differential game recent approaches and future requirements. In R. P. Hämäläinen, & H. K. Ehtamo (Eds.), *Differential games—Developments in modelling and computation*. Lecture notes in control and information sciences (Vol. 156). Berlin, Heidelberg: Springer.

Hashemi, A., Casbeer, D. W., & Milutinović, D. (2016). Scalable value approximation for multiple target tail-chase with collision avoidance. In 2016 IEEE 55th conference on decision and control (CDC) (pp. 2543–2548). https://doi.org/10.1109/CDC.2016.7798645.

Hoy, M., Matveev, A., & Savkin, A. (2015). Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey. *Robotica*, *33*(3), 463–497.

Huang, H., Ding, J., Zhang, W., & Tomlin, C. J. (2015). Automation-assisted capture-the-flag: A differential game approach. *IEEE Transactions on Control Systems Technology*, *23*(3), 1014–1028.

Isaacs, R. (1965). *Differential games*. New York, NY: Wiley.

Israelsen, B. W., Ahmed, N., Center, K., Green, R., & Bennett Jr., W. (2017). Adaptive simulation-based training of ai decision-makers using bayesian optimization. arxiv:1703.09310.

Kushner, H. J., & Dupuis, P. (2001). *Numerical methods for stochastic control problems in continuous time, stochastic modelling and applied probability* (Vol. 24). New York, NY: Springer.

Li, D., Cruz, J. B., & Schumacher, C. J. (2008). Stochastic multi-player pursuit-evasion differential games. *International Journal of Robust and Nonlinear Control*, *18*(6), 218–247.

McGrew, J. S., How, J. P., Williams, B., & Roy, N. (2010). Air-combat strategy using approximate dynamic programming. *Journal of Guidance, Control, and Dynamics*, *33*(5), 1509–1514.

Milutinović, D., Casbeer, D. W., Kingston, D., & Rasmussen, S. A. (2017). Stochastic approach to small uav feedback control for target tracking and blind spot avoidance. In Proceedings of the 1st IEEE conference on control technology and applications.

Munishkin, A. A., Milutinović, D., & Casbeer, D. W. (2016). Stochastic optimal control navigation with the avoidance of unsafe configurations. In 2016 international conference on unmanned aircraft systems (ICUAS) (pp. 211–218). https://doi.org/10.1109/ICUAS.2016.7502568.

Panagou, D., Stipanović, D. M., & Voulgaris, P. G. (2016). Distributed coordination control for multi-robot networks using Lyapunov-like barrier functions. *IEEE Transactions on Automatic Control*, *61*(3), 617–632.

Powell, W. B. (2009). What you should know about approximate dynamic programming. *Naval Research Logistics (NRL)*, *56*(3), 239–249.

Song, Q., & Yin, G. G. (2010). Convergence rates of Markov chain approximation methods for controlled diffusions with stopping. *Journal of Systems Science and Complexity*, *23*(3), 600–621.

Vidal, R., Shakernia, O., Kim, H. J., Shim, D. H., & Sastry, S. (2002). Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation*, *18*(5), 662–669.

Vieira, M. A. M., Govindan, R., & Sukhatme, G. S. (2009). Scalable and practical pursuit-evasion with networked robots. *Intelligent Service Robotics*, *2*(4), 247.

Virtanen, K., Karelahti, J., & Raivio, T. (2006). Modeling air combat by a moving horizon influence diagram game. *Journal of Guidance, Control, and Dynamics*, *29*(5), 1509–1514.

Wang, L., Ames, A. D., & Egerstedt, M. (2017). Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, *33*(3), 661–674. https://doi.org/10.1109/TRO.2017.2659727.

Yavin, Y. (1988). Stochastic two-target pursuit-evasion differential games in the plane. *Journal of Optimization Theory and Applications*, *56*(3), 325–343.

Yavin, Y., & Villers, R. D. (1988). Stochastic pursuit-evasion differential games in 3D. *Journal of Optimization Theory and Applications*, *56*(3), 345–357.

Alexey A. Munishkin research interests lie in optimization, multi-agent systems, and stochastic control. He received his B.S. degree in Computer Engineering with an emphasis in Robotics and Control from University of California, Santa Cruz in 2016. He is currently a graduate student pursuing a Ph.D. degree in Computer Engineering at the University of California, Santa Cruz.

Araz Hashemi research interests lie in estimation, decision making, and control in stochastic systems. He received his B.S. and Ph.D. degrees in Mathematics from Wayne State University in 2007 and 2014 respectively. He went on to a postdoctoral research position at the University of Delaware, and a National Research Council fellowship at the Air Force Research Labs. He is currently a Data Scientist for the Graham Media Group.

David W. Casbeer is the Technical Area Lead over Cooperative and Intelligent UAV Control with the Control Science Center of Excellence, Aerospace Systems Directorate, Air Force Research Laboratory, where he carries out and leads basic research involving the control of autonomous UAVs with a particular emphasis on high-level decision making and planning under uncertainty. He received B.S. and Ph.D. degrees in Electrical Engineering from Brigham Young University in 2003 and 2009, respectively. He currently serves as the chair for the

AIAA Intelligent Systems Technical Committee and a Senior Editor for the Journal of Intelligent and Robotic Systems.

Dejan Milutinović earned Dipl.-Ing (1995) and Magister's (1999) degrees in Electrical Engineering from the University of Belgrade, Serbia and a doctoral degree in Electrical and Computer Engineering (2004) from Instituto Superior Técnico, Lisbon, Portugal. From 1995 to 2000 he worked as a research engineer in the Automation and Control Division of Mihajlo Pupin Institute, Belgrade, Serbia. His doctoral thesis was the first runner-up for the best Ph.D. thesis of European Robotics in 2004 by EURON. He won the NRC award of the US Academies in 2008 and Hellman Fellowship in 2012. He is currently a Professor in the Department of Computer Engineering, UC Santa Cruz. His research interests are in the area of modeling and control of stochastic dynamical systems applied to robotics. He currently serves as an Associate Editor for the IEEE Robotics and Automation Letters (RA-L) and as an Editor for the Journal of Intelligent and Robotic Systems, Springer.