Robotics and Autonomous Systems 125 (2020) 103370

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Min–max time efficient inspection of ground vehicles by a UAV team

Alexey A. Munishkin [a], Dejan Milutinović [a,*], David W. Casbeer [b]

[a] Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, USA
[b] Control Science Center of Excellence, Air Force Research Laboratory, Wright-Patterson AFB, Dayton, OH 45433, USA

ARTICLE INFO

Article history:
Available online 16 December 2019

ABSTRACT

We present a control design for N unmanned aerial vehicles (UAVs) tasked with an inspection of M ground moving vehicles. The location of each ground vehicle is known to each UAV, but the navigation and intent of each ground vehicle are unknown, therefore, this uncertainty has to be anticipated in each UAV's navigation. We use the minimum time stochastic optimal control to navigate each UAV towards the inspection of each ground vehicle. Based on this control, we formulate assignments of ground vehicles to be inspected by UAVs as an optimization problem to inspect all ground vehicles in the minimum expected time. Accounting for ground vehicle uncertain trajectories, we update the optimal assignment by a Markov inequality rule. The rule prevents the possibility of indefinite updating of assignments without finishing the inspection of all vehicles. On the other hand, it updates an assignment if it leads to a statistically significant improvement of the inspection expected time. The presented approach is illustrated with numerical examples.

1. Introduction

This paper is the result of an exploration to formulate the control of stochastic multi-agent systems by an integration of stochastic optimal control strategies that are designed for a pair of agents. In our case, the pair is composed of one UAV and one ground vehicle. To study such an integration of the control strategies, we consider a scenario with a team of N unmanned air vehicles (UAVs) that are tasked to inspect efficiently with respect to time a group of M ground vehicles. Locations and headings of the ground vehicles are known to the UAVs, but navigation strategies of ground vehicles are unknown. To anticipate this uncertainty, the headings of ground vehicles are described as stochastic processes, and as a result, the UAV navigation has to be a solution of a feedback stochastic control problem.

A solution of the feedback stochastic optimal control problem for the N UAVs and M ground vehicles has to depend on the relative positions between any pair of UAVs and ground vehicles. Moreover, the solution should account for the number of assignment sequences in which N UAVs can inspect M ground vehicles. We can potentially formulate the minimum time stochastic optimal control using the Hamilton–Jacobi–Bellman (HJB) equation, but the number of variables, i.e., state variables with which we have to deal to keep track of all relative positions and assignment sequences, would quickly increase. This combinatorial increase in the number of state–space variables leads to the well-known *curse-of-dimensionality* problem [1], which refers to situations in which the computational complexity needed to solve the optimal solution goes beyond the computational power of modern computers. Consequently, we propose an approach in which we first solve for the stochastic optimal control of one UAV inspecting one ground vehicle (*one-on-one*) and use this to formulate a solution for the problem of N UAVs inspecting M ground vehicles.

The minimum time stochastic optimal control of a single UAV entering the tail sector of another vehicle while safely navigating around it is presented in [2]. The computational method for this type of *one-on-one* agent problem has been improved and used with a scalable value approximation [3] in a complex scenario of a single UAV safely intercepting a group of vehicles [4]. Here we consider a scenario of having multiple UAVs inspecting multiple ground vehicles and we assume that the UAVs are able to avoid each other, or more realistically, they are able to fly at different altitudes. Consequently, collision avoidance is not considered and the main problem lies in how to assign [5,6] each UAV to ground vehicles, so that the inspection time for all ground vehicles is time efficient.

Related assignment problems have been previously considered in [7,8], but in terms of path-planning [9]. Non path-planning, i.e., feedback-based navigations are frequently found in a line of work called sense and avoid [10], and game theoretic approaches to safe navigation [11]. They also appear in the context of pursuit-evasion games [12], including two car-like vehicles investigated in [13]. Scalable algorithms for multiple-pursuers/single-evader games have been considered in [14,15]. Game theoretic problems for multi-pursuit and multi-evasion strategies using a hierarchical

* Corresponding author.
E-mail address: dmilutin@ucsc.edu (D. Milutinović).

https://doi.org/10.1016/j.robot.2019.103370
0921-8890/

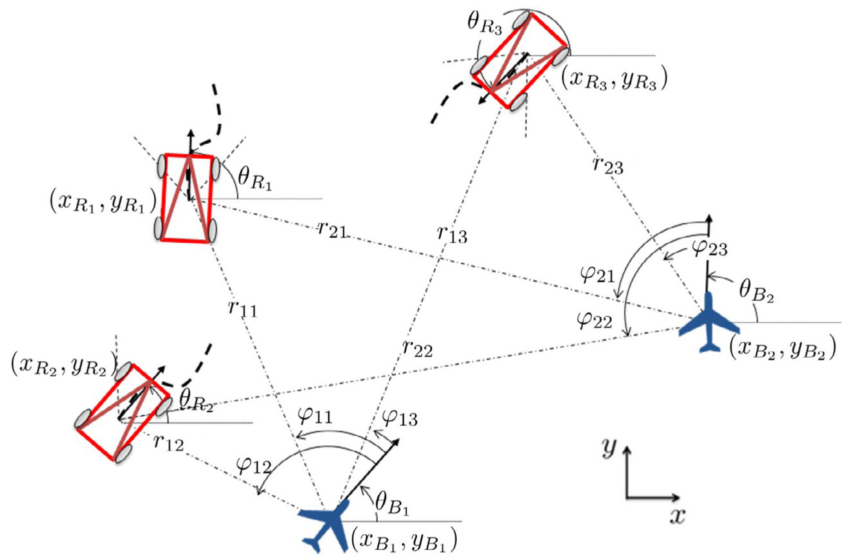


Fig. 1. Geometry of the multi-agent/multi-target problem: θ_{B_i} and θ_{R_j} are the heading angles of the ith blue and jth red, respectfully. $\alpha_{ji} = \theta_{R_i} - \theta_{B_j}$ is the difference between the headings, and φ_{ji} is the bearing angle of the jth blue to ith red. r_{ji} is the distance between jth blue to ith red.

or decomposition algorithmic approach [16,17] are more similar to the multi-agent problem in this paper.

In this work, we approach the problem of N UAVs inspecting M ground vehicles as a Markov inequality-based switching problem which is inspired by the result [18] for one UAV inspecting multiple ground vehicles. Dealing here with N UAVs, the switching is among possible inspection assignments of UAVs to the ground vehicles to be inspected. No knowledge of the ground vehicles navigation strategy or intent is known by UAVs, therefore, each of the ground vehicles heading angles is anticipated to be a Brownian random walk. The contribution of this paper is that it solves the stochastic multi-agent problem using *one-on-one* stochastic optimal control solutions and their composition with the promise of time-efficient navigation. The presented approach scales well with the number of agents and allows for real-time computations of control actions. The preliminary version of this work was presented in [19].

The paper is organized as follows. Section 2 discusses the problem, which stating succinctly is controlling a team of UAVs to inspect all ground vehicles in the minimum time. The stochastic optimal control of a single UAV to inspect a single ground vehicle in the minimum time is presented in Section 3. The time optimal assignment and dynamic re-assignment are discussed in Sections 4 and 5, respectively. Section 6 shows results of our numerical simulations and Section 7 provides conclusions.

2. Problem formulation

Let us consider a scenario with five agents depicted in Fig. 1. Three of the agents, labeled with R_i, $i = 1, 2, 3$, are ground vehicles with equal speeds $v_{R_i} = v_R$, and we refer to them as red agents. The other two agents, B_j, $j = 1, 2$, are fixed-wing aerial vehicles flying at different altitudes at constant speeds v_{B_j}, and we refer to them as blue agents. The constant speed assumption approximates that the UAVs cannot stop and that without energy constraints, the UAVs will fly at maximum speeds to cover larger areas of interest. Without the loss of generality, we will assume that the speeds of blue agents are equal, $v_{B_j} = v_B$. We will also assume that the speed of blue (aerial) agents is larger than the speed of red (ground) agents, $v_B > v_R$. The kinematics of the jth blue agent is described using the deterministic kinematics of a Dubins vehicle given as

$$dx_{B_j} = v_B \cos(\theta_{B_j})dt \tag{1}$$

$$dy_{B_j} = v_B \sin(\theta_{B_j})dt \tag{2}$$

$$d\theta_{B_j} = u_{B_j}dt \tag{3}$$

where the couple (x_{B_j}, y_{B_j}) and θ_{B_j} describe the 2D blue agent's position and heading angle, respectively. The control input for each blue is a bounded turning rate $u_{B_j} \in [-u_{max}, u_{max}]$.

We assume that each blue agent knows its relative position and heading relative to each red agent. These relative positions are uniquely defined based on the relative coordinates

$$r_{ji} = \sqrt{(x_{R_i} - x_{B_j})^2 + (y_{R_i} - y_{B_j})^2}, \tag{4}$$

$$\varphi_{ji} = \arctan\left(\frac{y_{R_i} - y_{B_j}}{x_{R_i} - x_{B_j}}\right) - \theta_{B_j}, \varphi \in [-\pi, \pi) \tag{5}$$

$$\alpha_{ji} = \theta_{R_i} - \theta_{B_j}, \alpha \in [-\pi, \pi) \tag{6}$$

where r_{ji} is the distance between B_j and R_i, φ_{ij} is the bearing angle from B_j to R_i, and α_{ji} is the difference between the R_i and B_j heading angles. The definitions of these coordinates are illustrated in Fig. 1. Based on these relative coordinates, we define the tail sector $\mathcal{T}_i(t)$ for each ith red agent as

$$\mathcal{T}_i(t) = \left\{ (r, \varphi, \alpha) : \begin{matrix} r \le r_{min} \\ |\varphi| \le \varphi_m \\ |\alpha| \le \alpha_m \end{matrix} \right\} \tag{7}$$

where φ_m and α_m are the angles defining the tail sector width and alignments of the blue agents' heading angles, and r_{min} is the tail sector length (see Fig. 2). A successful inspection is defined using (7) as $(r_{ji}, \varphi_{ji}, \alpha_{ji}) \in \mathcal{T}_i$, i.e., it occurs when B_j is in this sector of R_i and only if its heading is aligned with that of R_i.

The control problem in this paper is to find the control for the blue agent team providing that the time of inspection of all reds is minimal, i.e., that the blue team visit all tail sectors of red agents in minimum time. However, blue agents *have no knowledge of the red ground vehicles' navigation strategy, i.e., the motion of the corresponding tail sectors*. To anticipate that uncertainty, the kinematics of each red R_i, $i = 1, \ldots, M$ agent is modeled by the stochastic kinematics

$$dx_{R_i} = v_R \cos(\theta_{R_i})dt \tag{8}$$

$$dy_{R_I} = v_R \sin(\theta_{R_i})dt \tag{9}$$

$$d\theta_{R_i} = \sigma_R dw_i \tag{10}$$

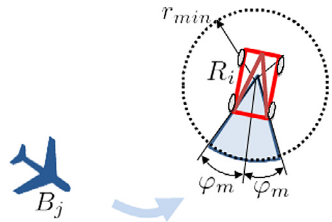


Fig. 2. Tail sector of red agent R_i: r_{min} is the length of the sector and the φ_m angle defines the width of the tail sector. Blue agent B_j is in the tail sector of R_i when its position is within the sector and its heading is aligned with the heading of R_i within the range of α_m, i.e., $|\theta_{R_i} - \theta_{B_j}| \leq \alpha_m$. The heading angles θ_{R_i} and θ_{B_j} are depicted in Fig. 1.

where the position of R_i is given by x_{R_i}, y_{R_i} and the heading angle is $\theta_{R_i} = \int_0^t \sigma_R dw_i$. The latter describes that the heading angles of red agents are random walks since dw_i denotes the Wiener process increments. The scaling parameter σ_R is identical for all red agents.

A solution of the minimum time feedback optimal control problem for N blue and M red agents depends on the number of variables that increases quickly due to the number of combinations in which N blue agents can visit M tail sectors of red agents. This number should also account for solutions allowing that a single blue agent may need to visit multiple red agent tail sectors. While in principle we can formulate the stochastic optimal control using the Hamilton–Jacobi–Bellman (HJB) equation and all necessary coordinates describing relative positions of blues and reds, the computational complexity of such a solution goes quickly beyond the computational power of modern computers because of the number of relative coordinates, as well as due to the so-called *curse-of-dimensionality* [1].

We approach the problem by dividing it into (1) the minimum time stochastic optimal control problem of one blue inspecting one red agent (*one-on-one*), and (2) the problem of computing optimal inspection assignments for the N blue agents to time efficiently inspect M reds. This approach allows us to formulate the solution for the navigation strategy of N blues that scales well with the number of agents and guarantees that all M red agents are inspected efficiently with respect to time. This guaranteed property and time efficiency are achieved by the optimality of the *one-on-one* problem and assignment updates when the assignment leads to a probabilistic chance for the decrease of inspection time for all red agents by the blues.

3. Minimum time stochastic optimal control

In this section, we deal with the scenario of a single blue B_j that enters the tail sector $\mathcal{T}_i$ of a single red R_i in the minimum expected time, thus $i = j = 1$. To simplify the notation, in this section we will drop the subscripts i and j from B_j, R_i and $\mathcal{T}_i$.

Using Itô calculus, the kinematic model of B (1)–(3), the kinematic model of R (8)–(10) and the definitions of relative coordinates between B and R (4)–(6), we can derive the following stochastic differential equations describing the evolutions of relative positions between B and R

$$dr = (v_R \cos(\varphi - \alpha) - v_B \cos(\varphi))dt = b_r dt \tag{11}$$

$$d\varphi = \left(-u_B + \frac{-v_R \sin(\varphi - \alpha) + v_B \sin(\varphi)}{r}\right) dt = b_\varphi dt \tag{12}$$

$$d\alpha = -u_B dt + \sigma_R dw = b_\alpha dt + \sigma_R dw \tag{13}$$

The minimum expected time control u_B for B to reach the target set $\mathcal{T}(t)$ of R is the control that minimizes the cost

$$\mathcal{J}(u_B) = E\left\{g(\tau) + \int_0^\tau 1dt\right\} \tag{14}$$

where $g(\tau) = g(r(\tau), \varphi(\tau), \alpha(\tau))$ is the terminal cost defined as

$$g(\tau) = \begin{cases} 0 & \text{if } (r(\tau), \varphi(\tau), \alpha(\tau)) \in \mathcal{T} \\ M & \text{if } (r(\tau), \varphi(\tau), \alpha(\tau)) \in \mathcal{P} \end{cases} \tag{15}$$

and the set $\mathcal{P}$ is defined as

$$\mathcal{P}(t) = \left\{(r, \varphi, \alpha) : \begin{matrix} r \leq r_{min} \\ |\varphi| > \varphi_m \\ |\alpha| > \alpha_m \end{matrix}\right\} \tag{16}$$

The cost function $\mathcal{J}$ is constructed to yield the optimal control u_B that minimizes the time for B to reach the target set $\mathcal{T}$ and avoids configurations in which B is in the proximity of R, but not in the target set $\mathcal{T}$. This is expressed by the definition of terminal cost $g(\tau)$ in which there is a large positive penalty $M \gg 0$ for the set $\mathcal{P}$ and no penalty for reaching the set $\mathcal{T}$.

The cost function $\mathcal{J}(u_B)$ gives rise to the stochastic HJB equation defining the evolution of the cost-to-go function $U = U(r, \varphi, \alpha)$ for the optimal control

$$0 = \min_{u_B}\left\{b_r \frac{\partial U}{\partial r} + b_\varphi \frac{\partial U}{\partial \varphi} + b_\alpha \frac{\partial U}{\partial \alpha} + \sigma^2 \frac{\partial^2 U}{\partial \alpha^2} + 1\right\} \tag{17}$$

with the two boundary conditions $U = 0$ for all $(r, \varphi, \alpha) \in \mathcal{T}$ and $U = M$ for all $(r, \varphi, \alpha) \in \mathcal{P}$. The solution of the HJB equation yields the cost-to-go function U and the corresponding optimal state feedback control $u_B = u_B(r, \varphi, \alpha)$. Once the optimal control is computed, it can be used to compute the expected time $V = V(r, \varphi, \alpha)$ to reach the target under the optimal control by computing the steady-state solution of the backward Kolmogorov equation

$$0 = b_r \frac{\partial V}{\partial r} + b_\varphi \frac{\partial V}{\partial \varphi} + b_\alpha \frac{\partial V}{\partial \alpha} + \sigma^2 \frac{\partial^2 V}{\partial \alpha^2} + 1 \tag{18}$$

with the boundary condition $V = 0$ for all $(r, \varphi, \alpha) \in \mathcal{T}$ and reflective boundary condition elsewhere on the boundary of the solution domain.

The solution of the optimal control problem used in this paper is based on the so-called locally consistent Markov chain discretization of the HJB equation [20]. The discretization yields a Markov chain with control u_B-dependent transition probabilities while the problem of solving the HJB equation is converted into a discrete state space dynamic programming problem that can be solved using value iterations [21]. Further details about the numerical method can be found in [4] and [2]. In both of these papers, the controllers have been implemented and tested with ground robots as a part of different control problems. Similar controllers have been used for the navigation of a small UAV in the presence of stochastic winds [22]. A general explanation of the method for the control of nonholonomic vehicles is given in [23]. The method has also been used for target tracking problems [24] and flight tested with real UAVs [25].

Once the control is computed, we can use it to compute the solution of the Kolmogorov equation (18) to obtain expected time V, i.e. its discrete space numerical representation. Due to the similarity of the Kolmogorov equation (18) with (17), the computations are based on the same discretization scheme and value iterations [21], except that instead of *min* operator, the value iterations are based on already computed optimal control.

The units in (4)–(6), which we used to compute the numerical optimal control in the example of this paper, are normalized so that all the angles are in radians and the velocities are $v_B = 0.1$ and $v_R = 0.05$. The noise scaling parameter $\sigma_R = 10\pi/180$ and the maximum turning rate of each blue $u_{max} = 0.5$. The tail sector (7) to be reached by the blue is defined by $r_{min} = 0.05$, $\varphi_m = 10\pi/180$, and $\alpha_m = 20\pi/180$. The computational domain is

$$\mathcal{K} = \{[r_{min}, R_{max}] \times [-\pi, \pi - \Delta\varphi] \times [-\pi, \pi - \Delta\alpha]\} \tag{19}$$

with the smallest distance equal to r_{min} and the largest distance $R_{max} = 2.04$. The domain (19) is discretized with the steps $\Delta r = (R_{max} - r_{min})/99 \approx 0.0201$ and $\Delta\varphi = \Delta\alpha = 5\pi/180$ in the direction of r, φ, α state space variables to obtain points $(r^h, \alpha^h, \varphi^h) \in \mathcal{K}$ of the discretized computational domain. Since the angles φ and α in our problem formulation have full ranges, in the discretized computational domain the pairs of points $(r^h, -\pi, \alpha^h)$ and $(r^h, \pi - \Delta\varphi, \alpha^h)$, as well as $(r^h, \varphi^h, -\pi)$ and $(r^h, \varphi^h, \pi - \Delta\alpha)$ are next to each other, i.e., we use periodic boundary conditions along the φ and α state space variables. Other boundary conditions involved in locally consistent Markov chain approximation method based computations of U^h and V^h, which are discrete approximations of U and V in (17) and (18), respectively, take into account (15), which are the boundary conditions for the tail sector (7) and (16). Since in our computational domain r_{min} is both the length of the tail sector, as well as, the smallest distance of the computational domain, the target set $\mathcal{T}$ and the set $\mathcal{P}$ are in our discrete space computations represented as

$$\begin{aligned} \mathcal{T}^h &= \left\{ (r^h, \varphi^h, \alpha^h) : \begin{matrix} r^h = r_{min} \\ |\varphi^h| \le \varphi_m \\ |\alpha^h| \le \alpha_m \end{matrix} \right\}, \\ \mathcal{P}^h &= \left\{ (r^h, \varphi, \alpha) : \begin{matrix} r^h = r_{min} \\ |\varphi^h| > \varphi_m \\ |\alpha^h| > \alpha_m \end{matrix} \right\} \end{aligned} \tag{20}$$

Our computations result in a discrete approximation V^h of expected times V, therefore we should note that the number of possible values that can be taken by V^h cannot be larger than the number of discrete points $(r^h, \varphi^h, \alpha^h) \in \mathcal{K}$ in our discretized computational domain. Due to that, we can define an expected time threshold $\underline{C} > 0$ as

$$\underline{C} = \min_{\{(r^h,\varphi^h,\alpha^h)\in\mathcal{K}\}\setminus\mathcal{T}^h} V^h(r^h, \varphi^h, \alpha^h) \tag{21}$$

which is the smallest expected time over all discrete points from the discretized computational domain that are outside of the target set. After we introduce the time optimal inspection assignment in the next section, in Section 5 we discuss a dynamical update of assignments and use $\underline{C}$ as a threshold value to conclude that any further assignment updates are unnecessary.

4. Time optimal inspection assignment

A general formulation of the assignment problem in which B_j, $j = 1, \ldots, N$ agent should inspect R_i, $i = 1, \ldots, M$, should take into account that a single B_j may be assigned to a sequence of R_i agents and that all R_i agents are assigned to at least one B_j agent. This type of problem can be formulated as an optimization problem on a graph. In the most general case, the graph nodes would be blue and red agents, and the edges between them would have an associated time to travel from any blue agent to any red agent. The solution would be a sequence of red agents that should be visited by each blue. However, in this paper we assume that the red agents are moving stochastically, which creates a major challenge for an optimal assignment solution. On the other hand, we can rely on the results obtained from the stochastic optimal control one-on-one solution presented in the previous section for easily obtaining the edge costs associated with the graph assignment problem.

To address the assignment problem, we consider the optimization on a bipartite graph. The graph is depicted in Fig. 3 and its edges are associated with the expected times $V_{j,i}(t) = V(r_{ji}(t), \varphi_{ji}(t), \alpha_{ji}(t))$ of B_j reaching $\mathcal{T}_i$, i.e., jth blue inspecting the ith red agent. Given these expected times $V_{j,i}$, we propose an assignment that at $t = 0$ minimizes the expected time of inspection to the last red, which corresponds to the longest expected time $V_{j,i}$. This can be formulated as

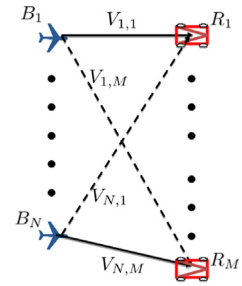


Fig. 3. Assignment graph problem, where $V_{i,j}$ denotes the expected time of B_j inspecting R_i. The dashed lines denote a possible assignment and the solid lines denote an assigned blue–red pair.

$$C^o = \min_{\mathcal{A}} \left\{ \max_{j\in\mathcal{N}_0} \{V_{j,i}(0) z_{j,i}(0)\} \right\}, \quad z_{j,i} \in \{0, 1\} \tag{22}$$

$$j \in \mathcal{N}_0 \subseteq \{1, 2, \ldots, N\}, i \in \mathcal{M}_0 \subseteq \{1, 2, \ldots, M\} \tag{23}$$

$$\text{subject to} \begin{cases} \sum_{i\in\mathcal{M}_0} z_{j,i}(0) = 1, & \text{for all } j \\ \sum_{j\in\mathcal{N}_0} z_{j,i}(0) \ge 1, & \text{for all } i, \text{ if } |\mathcal{N}_0| \ge |\mathcal{M}_0| \\ \sum_{j\in\mathcal{N}_0} z_{j,i}(0) \le 1, & \text{for all } i, \text{ if } |\mathcal{N}_0| < |\mathcal{M}_0| \end{cases} \tag{24}$$

where C^o is the initial optimal cost corresponding to the initial optimal assignment $A^o \in \mathcal{A}$ and the minimization is over a finite set of all possible assignments $\mathcal{A} = \{A_1, A_2, \ldots\}$. Each assignment A_k is uniquely defined by the assignment variables $z_{j,i}$ that can be depicted by the graph in Fig. 3. The assignment variable $z_{j,i} = 1$ if B_j is assigned to inspect R_i, otherwise, $z_{j,i} = 0$. The sets $\mathcal{N}_0$ and $\mathcal{M}_0$ contain indexes of all blue and red agents, respectively, that are included in the assignment, and $|\mathcal{N}_0|$, $|\mathcal{M}_0|$ denote cardinal numbers of these sets.

The first constraint in (24) states that the assignment has to have each B_j assigned to one R_i. The second one states that if the number of blue agents is greater than or equal to the number of red agents $|\mathcal{N}_0| \ge |\mathcal{M}_0|$, then each R_i must be assigned to at least one blue agent. Finally, the third constraint covers the case when the number of red agents is greater than the number of blue agents $|\mathcal{N}_0| < |\mathcal{M}_0|$, which states that each red agent must be assigned to at most one blue agent, hence there will be red agents that are left unassigned.

By introducing a new decision variable μ, we can rewrite the optimization problem (23)–(24) as a mixed integer linear programming (MILP) problem

$$C^o = \min \mu \tag{25}$$

$$z_{j,i} \in \{0, 1\}, \; j \in \mathcal{N}_0 \subseteq \{1, 2, \ldots, N\}, i \in \mathcal{M}_0 \subseteq \{1, 2, \ldots, M\} \tag{26}$$

$$\text{subject to} \begin{cases} V_{j,i}(0) z_{j,i} \le \mu, \text{ for all } i, j \\ \mu \ge 0, \\ \sum_{i\in\mathcal{M}_0} z_{j,i}(0) = 1, & \text{for all } j \\ \sum_{j\in\mathcal{N}_0} z_{j,i}(0) \ge 1, & \text{for all } i, \text{ if } |\mathcal{N}_0| \ge |\mathcal{M}_0| \\ \sum_{j\in\mathcal{N}_0} z_{j,i}(0) \le 1, & \text{for all } i, \text{ if } |\mathcal{N}_0| < |\mathcal{M}_0| \end{cases} \tag{27}$$

The MILP minimizes the variable μ, and $\max\{\cdot\}$ in the objective function (22) is replaced by the inequality constraints $V_{j,i}(0) z_{j,i} \le \mu$ for all $j \in N_0$ and $i \in M_0$. The second constraint $\mu \ge 0$ is not necessary, but it clarifies that μ has to be positive since it is a time variable.

5. Dynamic switching for re-assignment

To discuss the switching among assignments, we introduce the cost $C_k(t)$ of the assignment A_k evaluated at time t as

$$C_k(t) = \max_{j \in \mathcal{N}_0}\{V_{j,i}(t) z_{j,i}(t)\}, \ \ i \in \mathcal{M}_0 \tag{28}$$

where $C_k(t)$ is the current cost of assignment A_k and the initial cost in (25) is $C^o = \min_k\{C_k(0)\}$. The current assignment A_k is uniquely defined by its corresponding $z_{j,i} \in \{0, 1\}$ values, therefore, the cost of the assignment is defined as the maximum over the blue agents, i.e., maximum of their times to inspect assigned red agents.

Since red agents are moving stochastically, it is obvious that the initial configuration with cost C^o defined by the solution of (25) may be inferior to any other configuration A_k for which $C_k(t) < C^o$. To pursue the idea of optimality of assignment at every time instant t, one may think about solving the optimization (25) at every time instant (greedy approach). However, note that the navigation strategy is based on the minimization of expected times to the target sets $\mathcal{T}_i$, therefore, there is a non-zero probability for the increase of times to target sets. Because of that, although there may exist multiple assignments that result in the inspection of all red agents, once we start switching among them using the greedy approach, it can result in an infinite sequence of assignment switchings without ever inspecting any red agent. This is a well-known characteristic of the so-called *hybrid systems*.

To resolve the problem of an infinite sequence of switchings, here we propose to use the switching rule that was presented and analyzed in [18]. The rule is that if at time t

$$\frac{\min\{C_k(t), C_k(\tau^s)\} - C_m(t)}{\min\{C_k(t), C_k(\tau^s)\}} \geq p \text{ and } C_k(\tau^s) > \underline{C}, \quad p \in (0, 1) \tag{29}$$

where $\underline{C}$ is defined by (21), then the switching from the assignment $A_k \in \mathcal{A}$ to $A_m \in \mathcal{A}$ takes place, $k \neq m$. In this rule, k is the index of the current assignment and m is the index of the next assignment, which is not necessarily $k+1$, $\tau^s < t$ is the time at which the assignment becomes A_k, and $s \geq 0$ is the number of total switchings from $t = 0$. Hence after the switching at time $\tau^{s+1} = t$, s is incremented by one.

If τ_F defines the time to inspect all red agents, then the above rule guarantees

$$Pr\left\{\tau_F < \min\{C_k(t), C_k(\tau^s)\}\right\} \geq p \tag{30}$$

which means that the switching happens only if it provides that the time τ_F is with a probability p shorter than the shortest of the current expected time to inspect all red agents and the expected time to inspect all red agents immediately after τ^s, i.e., after the last switching of the assignment. The rule does not explicitly shorten the time to capture any specific red agent, but the last red agent, i.e., shorten the expected time to inspect them all. As an example, a certain blue agent can come to a relative position in which it can inspect a particular red agent in a short time interval. The rule will re-assign that blue agent to another red agent if the re-assignment reduces the expected time to inspect the last red agent. In the light of this, the condition $C_k(\tau^s) > \underline{C}$ is the mild one and prevents further assignment updates only if the time for inspecting the whole team of red agents is shorter than $\underline{C}$, which is the shortest non-zero expected time in our computational domain as defined by (21).

Following from [18], the switching rule (29) guarantees that after every switch, the maximum expected time to inspect the next red agent decreases and that its inspection will happen in a finite expected time. After the inspection of the red agent, we can set $t = 0$, exclude the inspected agent from $\mathcal{M}_0$ and potentially compute a new assignment based on the updated set $\mathcal{M}_0$ of red

Initial sets $\mathcal{N}_0$ and $\mathcal{M}_0$
FirstAssignment $= True$
While $|\mathcal{M}_0| \neq 0$
If ($|\mathcal{N}_0| < |\mathcal{M}_0|$) or (FirstAssignment $== True$) ;then

$$C_0 = \min \mu, \ z_{j,i} \in \{0,1\}$$
$$j \in \mathcal{N}_0 \subseteq \{1, 2, \dots N\}, i \in \mathcal{M}_0 \subseteq \{1, 2, \dots M\}$$
$$\text{subject to} \begin{cases} V_{j,i}(0) z_{j,i}(0) \leq \mu, \text{for all } i, j \\ \mu \geq 0 \\ \sum_{i \in \mathcal{M}_0} z_{j,i}(0) = 1, \quad \text{for all } j \\ \sum_{j \in \mathcal{N}_0} z_{j,i}(0) \geq 1, \quad \text{for all } i, \text{ if } |\mathcal{N}_0| \geq |\mathcal{M}_0| \\ \sum_{j \in \mathcal{N}_0} z_{j,i}(0) \leq 1, \quad \text{for all } i, \text{ if } |\mathcal{N}_0| < |\mathcal{M}_0| \end{cases}$$

FirstAssignment $= False$
end
While none of $R_i \in \mathcal{M}_0$ are inspected
If $\frac{\min\{C_k(t), C_k(\tau^s)\} - C_m(t)}{\min\{C_k(t), C_k(\tau^s)\}} \geq p$, switch A_k to A_m
end
Update $\mathcal{M}_0$
end

Fig. 4. Dynamic re-assignment algorithm: Sets $\mathcal{N}_0$ and $\mathcal{M}_0$ of available blue and red agents, respectively. The number of agents in each set is $|\mathcal{N}_0|$ and $|\mathcal{M}_0|$; C_k is the cost of the assignment A_k and is defined by (28).

agents to be inspected. However, in this case the corresponding optimal cost can be larger than that computed for the initial problem before the inspection. Therefore, in the case of $|\mathcal{N}_0| \geq |\mathcal{M}_0|$, after the inspection of a red agent by a blue agent, we set $t = 0$, assign the blue agent to the closest red agent from the updated set $\mathcal{M}_0$ of red agents to be inspected, evaluate the cost of the current assignment $C_k(0)$ and continue using the switching rule (29).

In the case when the number of blue agents is smaller than the number of red agents $|\mathcal{N}_0| < |\mathcal{M}_0|$, the initial assignment does not assign some of the red agents to blue agents. Therefore, in this case after a red agent is inspected by a blue agent, we set $t = 0$ and compute a new assignment for the set $\mathcal{M}_0$, which includes only red agents that have not been previously inspected or assigned. The final time we do it is for the case $|\mathcal{N}_0| = |\mathcal{M}_0|$.

In summary, initially or after every inspection of a red agent by a blue agent when $|\mathcal{N}_0| \leq |\mathcal{M}_0|$, we compute the optimal assignment for the current positions of all agents (greedy approach). Otherwise, for $|\mathcal{N}_0| > |\mathcal{M}_0|$ and between inspections, we use the rule (29) for switching among different assignments. The overall algorithm is depicted in Fig. 4. Since the rule guarantees a finite expected time for the inspection of the next red agent [18], the overall algorithm guarantees that the expected time for the inspection of the whole team is also finite.

6. Results

In this section, we first present an illustrative example with 2 blue and 3 red agents. The example deals with a smaller number of agents and allows us to describe the time progress of the presented scenario. In stochastic simulation runs with the same number of agents and the same initial conditions, there are simulation runs in which the assignment switching does not happen, therefore, the example is illustrative since it involves the assignment switching. To discuss the performance of the proposed approach, we run a larger number of numerical simulations with 3 blue and 3 red agents, and discuss the distribution of times for the inspection of the whole team. As a part of that discussion, we also report briefly average times for the inspection in a scenario with 3 blue and 6 red agents.

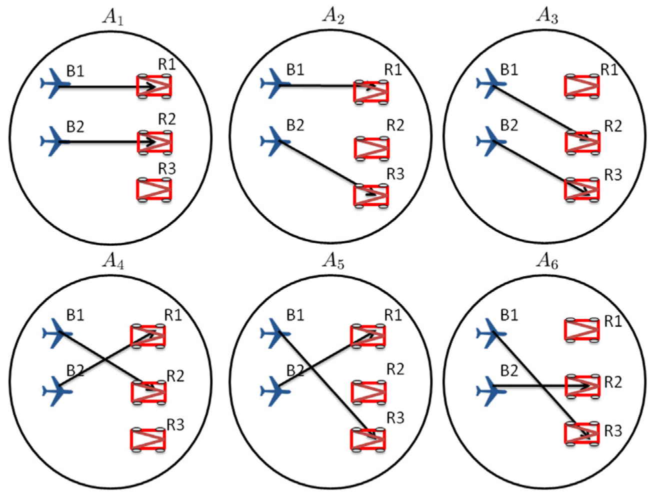


Fig. 5. Possible assignments: each assignment is depicted by the lines connecting the UAVs ($B1$, $B2$) with the ground vehicles ($R1$, $R2$, $R3$). Beyond these assignments, those in which a single ground vehicle is assigned to both UAVs are labeled by the single label A_0.

6.1. Illustrative example

To illustrate our results, we use an example with two UAVs (blue agents) and three ground vehicles (red agents). The velocity of UAVs is $v_B = 0.1$ and the velocity of ground vehicles is $v_R = 0.05$. These and all other problem parameters for the example are provided in Section 3, before and after (19), defining the computational domain of the *one-on-one* stochastic optimal control solution. Once the numerical solution and the expected time have been computed, we search over the discrete space of the solution to find the minimal non-zero expected time for a UAV to inspect a ground vehicle. This value defines the threshold $\underline{C} = 1.649$ s. Once the threshold is reached, we stop the switching assignments using the rule (29). The switching assignment rule parameter in the example is $p = 0.05$ and the rule defines switching among possible assignments.

In this example, we can identify six possible assignments that are labeled as A_i $i = 1, 2\ldots6$ and depicted in Fig. 5. Beyond these assignments, the only other possible assignments are those in which a single ground vehicle is assigned to both UAVs, and we label all of them with the single label A_0. Given the small number of possible assignments, the optimization (22) can be performed by the evaluation of each assignment and the selection of the one with the smallest cost. However, the identification of all possible assignments and their enumeration would be a very complex process in a situation with more blue and red agents, and to avoid it we truly benefit from the MILP formulation of the optimal assignment.

For example, in the case of 2 blue and 3 red agents, we define a vector $\mathbf{z}$ as

$$\mathbf{z} = [z_{1,1}\ z_{1,2}\ z_{1,3}\ z_{2,1}\ z_{2,2}\ z_{2,3}\ \mu]^T,\ \ \mathbf{z} \in \{0, 1\}^6 \times \mathbf{R} \tag{31}$$

which is composed of 6 binary and one real number components, and introduce the vector $\mathbf{f} = [0\ 0\ 0\ 0\ 0\ 0\ 1]^T$ to formulate the MILP from the previous section as

$$C^o = \min \mathbf{f}^T\mathbf{z} \tag{32}$$

$$\mathbf{A}\mathbf{z} \le \mathbf{b} \tag{33}$$

$$\mathbf{B}\mathbf{z} = \mathbf{1}_{2\times 1} \tag{34}$$

where

$$\mathbf{A} = \begin{bmatrix} V_{1,1} & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & V_{1,2} & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & V_{1,3} & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & V_{2,1} & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & V_{2,2} & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & V_{2,3} & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix},\ \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{35}$$

While the presented formulation is specific to the case of 2 blue and 3 red agents, the matrices **A**, **B** and vectors on the right side of inequality and equality constraints can be generated and updated automatically in the numerical simulations to address any possible number of blue and red agents. An efficient numerical implementation of the MILP solution computations is beyond the scope of this paper, however, as an example, if we use the MATLAB function *intlinprog*, the solution for 2 blue and 3 red agents is computed in less than a second on an Intel Core i5 processor with 8 GB of RAM.

The simulation in Fig. 6A starts with the two blue UAVs behind the three red ground vehicles, where B_1 is assigned to inspect R_1 and B_2 to inspect R_2, i.e., the current assignment is A_1 as shown in Fig. 5. In Fig. 6B, B_1 switches its assignment to R_3 and after this, B_2 switches to R_1 (see Fig. 6C). This assignment lasts until R_3 has been inspected by B_1 at $t = 14.6$ s (see Fig. 6D) and is no longer of interest to B_1 or B_2.

As depicted in Fig. 7, at the beginning of time interval D a new optimal assignment is computed for B_1 and B_2. Because of that, we can observe a positive jump in the assignment cost, which is the longest expected time of the assignment. The new assignment is that B_1 inspects R_1 with the expected time to inspection of 31.7 s, and B_2 inspects R_2 with the expected time to inspection of 28.1 s; therefore, the cost of the assignment is 31.7 s This is the optimal assignment since the alternative assignment has the cost of 37.3 s, which results from the expected times of 37.3 s for B_1 inspecting R_2 and 11 s for B_2 inspecting R_1. Following the optimal assignment, R_2 has been inspected by B_2 at $t = 23.7$ (see Fig. 6E) without any switchings. Following this, the only assignment is A_0 in which both B_1 and B_2 inspect the remaining ground vehicle R_1. Fig. 7*a* shows that R_1 has been inspected by B_1 at $t = 32.8$ s at the end of the time interval F.

6.2. Numerical simulation with three blue and three red agents

For a scenario with 3 blue and 3 red agents, we use an example in which the agents are initially positioned as depicted in Fig. 8a. We generate 1000 possible stochastic trajectories for each red agent. Then we use those trajectories to run numerical simulations with three blue agents for 1000 times and two types of re-assignments. The first type of the re-assignments is the one in which the assignment is computed at the beginning of the simulation and after the inspection of each red agent. We refer to that type of the re-assignments as *sequential* since we do not allow for dynamic re-assignments using the rule (29) between the inspections of two red agents. The second type of the re-assignments is a *dynamic* re-assignment as described in Fig. 4.

From the 1000 simulation runs for the *sequential* and *dynamic* re-assignments, we record times for the team of blue agents to

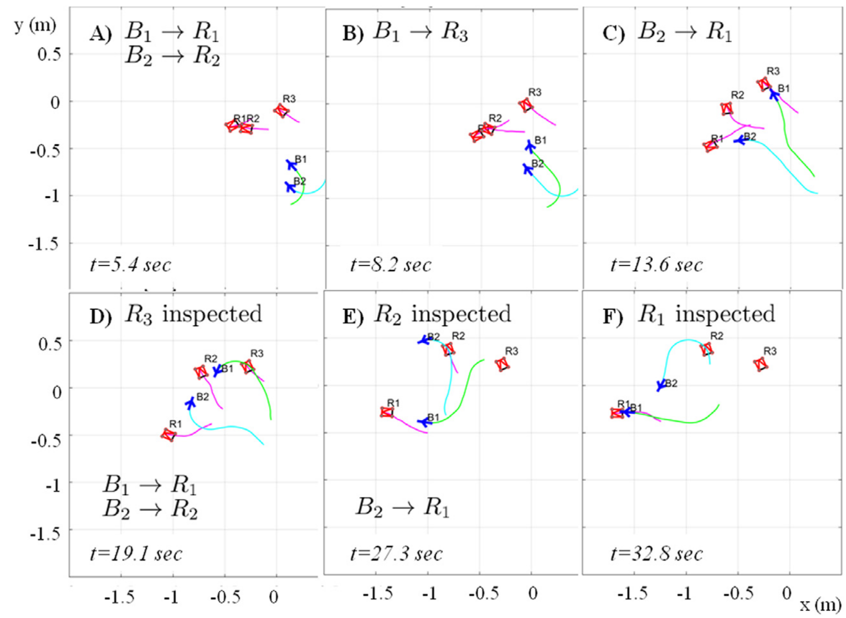


Fig. 6. Simulation in which the two UAVs switch inspect the three ground vehicles. The progress of time is from A to F. The ground vehicles R_3, R_2, R_1 have been inspected at $t = 14.6$, $t = 23.7$, $t = 32.8$, respectively.

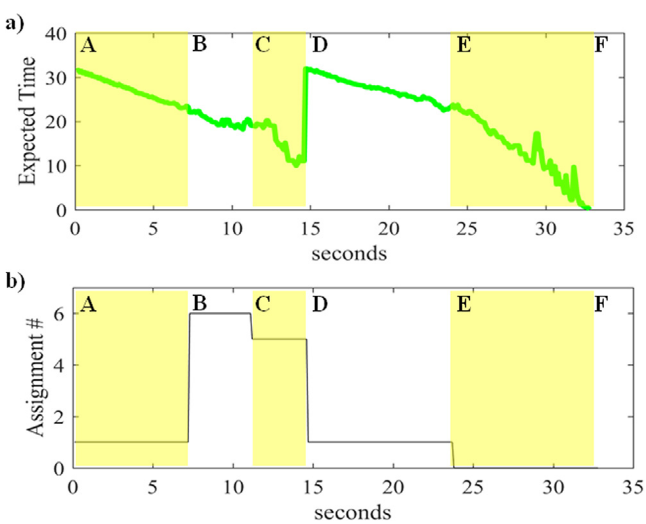


Fig. 7. (a) The cost of the current assignment A_k (see Fig. 4), which is the longest expected time of the assignment; (b) the current assignment.

inspect all three red agents. The average time for the *sequential* re-assignments is 98.0303 s (std = 63.1493 s) and for the *dynamic* ones is 92.8141 s (std = 15.9089 s). Therefore, the straightforward conclusion is that the *dynamic* re-assignments improve the time for the inspection of the whole team. Once we plot the distribution of the inspection time (see Fig. 9), we find that the difference in averages is mostly due to a small number of inspected times that are extremely long. When we exclude these data from our computations for the *sequential* re-assignment average, we obtain 92.5556 s (std = 16.3641 s). Because of that, we can conclude that the *dynamic* re-assignments improve the time to inspect the whole team by preventing very long inspection times. This is not unexpected given that the rule for *dynamic* switching re-assignments deals with shortening the time to inspect the last red agent.

The significance of the scenario with $|\mathcal{N}_0| = |\mathcal{M}_0|$ is that it includes the larger number of red agents, in our case 3, for which the MILP optimization applies only initially and the re-assignments are always performed based on the switching rule. This is not the case once we consider a scenario in which the number of red agents is larger than the number of blue agents $|\mathcal{N}_0| < |\mathcal{M}_0|$. As an example, we considered the scenario in which 3 blue and 6 red agents were initially positioned as depicted in Fig. 8b. After the 1000 simulations, we found no difference in the performance with or without *dynamic* re-assignments. We conclude that this is because the switching rule does not apply all the time, but only after the number of both blue and red agents to be inspected is 3.

For an illustration of the simulations of 3 blue vs. 3 red agents and 3 blue vs. 6 red agents, we provide two examples with the dynamic re-assignments. (Please follow the links to the videos in the supplementary material: link1, link2).

7. Conclusions

In this paper we presented the control design for N UAVs tasked to perform the time efficient inspection of M ground moving vehicles. The navigation and intent of each ground vehicle are unknown, therefore, the uncertainty of its navigation has to be anticipated in the navigation of each UAV. The controller for each UAV to inspect each ground vehicle is based on the minimum time stochastic optimal control. This *one-on-one* vehicle optimal control solution is used to compute the expected time of the inspection. We further use that expected time to formulate the assignment problem of deciding what ground vehicle each UAV should inspect. We formulate it as the optimization problem of minimizing the expected time to inspect all ground vehicles. Since the ground vehicles have uncertain trajectories, the optimal assignment may need to be recomputed. However, the recomputing may result in an indefinite sequence of assignment updates without the UAVs ever inspecting all ground vehicles. To address that, we update assignments with the Markov inequality rule. While the rule prevents the possibility of indefinite changes

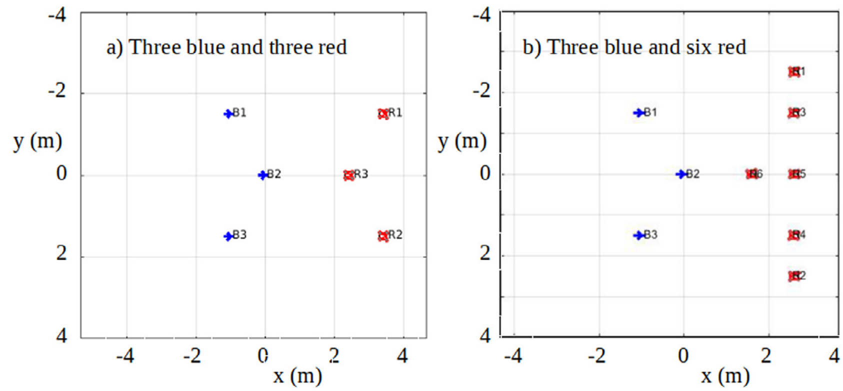


Fig. 8. The initial configurations of blue and red agents: the blue agents are presented with arrow-like symbols depicting aircraft and the red agents with rectangular symbols depicting ground vehicles, i.e., cars. (a) 3 blue and 3 red agents; (b) 3 blue and 6 red agents.

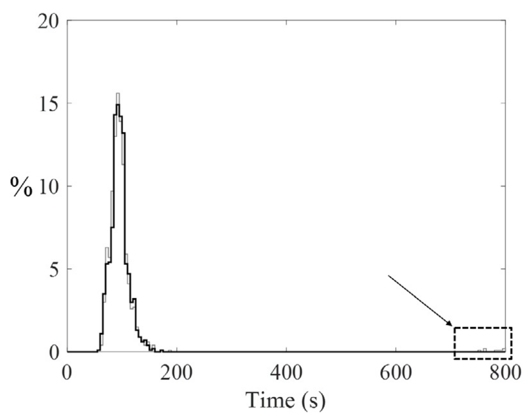


Fig. 9. Results of 1000 simulations: the histograms of the times to the inspection of the last red. The thick line corresponds to the *dynamic* re-assignment algorithm and the thin line to the *sequential* one. The dashed line around 800 s highlights the portion of *sequential* runs that took longer than 700 s.

of assignments, it also updates an assignment if it leads to a statistically significant improvement of the expected time of the inspection. The proposed approach was illustrated by the numerical example with 2 UAVs and 3 ground vehicles. Furthermore, we ran a large number of numerical simulations for 3 UAVs and 3 ground vehicles, and another for 3 UAVs and 6 ground vehicles. The results show that if the number of UAVs is greater than or equal to the number of ground vehicles, then the dynamic algorithm does better than the sequential one by preventing very long inspection times. This is due to the Markov inequality rule updates that apply to all re-assignments and shorten the time of inspection for the group of red agents to be inspected [19].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Funding was provided by U.S. Department of Defense (FA8650-17-F-2220).

Appendix A. Supplementary data

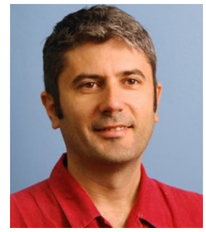
Supplementary material related to this article can be found online at https://doi.org/10.1016/j.robot.2019.103370.

References

[1] W.B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality, in: Wiley Series in Probability and Statistics, 2007.
[2] A.A. Munishkin, D. Milutinović, D.W. Casbeer, Stochastic optimal control navigation with the avoidance of unsafe configurations, in: Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), 2016.
[3] A. Hashemi, D.W. Casbeer, D. Milutinović, Scalable value approximation for multiple target tail-chase with collision avoidance, in: Proceedings of the 55th IEEE Conference of Decision and Control (CDC), 2016.
[4] A.A. Munishkin, A. Hashemi, D.W. Casbeer, D. Milutinović, Scalable Markov chain approximation for a safe intercept navigation in the presence of multiple vehicles, Auton. Robots 43 (3) (2019) 575–588.
[5] R. Burkard, M. Dell'Amico, S. Martello, Assignment Problems: Revised Reprint, SIAM: Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009, chapter 4: Linear Assignment Problem.
[6] M.Z. Spivey, W.B. Powell, Dynamic assignment problem, Transp. Sci. 38 (4) (2004) 399–419.
[7] R.W. Beard, T.W. McLain, M.A. Goodrich, E.P. Anderson, Coordinated target assignment and intercept for unmanned air vehicles, IEEE Trans. Robot. Autom. 18 (6) (2002) 911–922.
[8] J.S. Bellingham, M. Tillerson, M. Alighanbari, J.P. How, Cooperative path planning for multiple UAVs in dynamic and uncertain environments, in: Proceedings of the 41st IEEE Conference on Decision and Control, Vol. 3, 2002, pp. 2816–2822.
[9] S.M. LaValle, Planning Algorithms, Cambridge University Press, 2006.
[10] P. Angelov, Sense and Avoid in UAS: Research and Applications, John Wiley & Sons, 2012.
[11] I.M. Mitchell, A.M. Bayen, C.J. Tomlin, A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games, IEEE Trans. Automat. Control 50 (7) (2005) 947–957.
[12] R. Isaacs, Games of Pursuit, RAND Corporation, Santa Monica, CA, 1951.
[13] W.M. Getz, M. Pachter, Capturability in a two-target game of two cars, J. Guid. Control 4 (1) (1981) 15–22.
[14] A. Festa, R.B. Vinter, Decomposition of differential games with multiple targets, J. Optim. Theory Appl. (2016) 849–875.
[15] A. Von Moll, D.W. Casbeer, E. Garcia, D. Milutinović, Pursuit-evasion of an evader by multiple pursuers, in: Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), 2018.
[16] J. Ge, L. Tang, J. Reimann, G. Vachtsevanos, Hierarchical decomposition approach for pursuit- evasion differential game with multiple players, in: IEEE Aerospace Conference, 2006.
[17] D.J.B.C. Li Jr., G. Chen, C. Kwan, M.H. Chang, A hierarchical approach to multi-player pursuit-evasion differential games, in: Proceedings of the 44th IEEE Conference of Decision and Control (CDC), Vol. 44 (5), 2005, pp. 5674–5679.
[18] D. Milutinović, D.W. Casbeer, M. Pachter, Markov inequality rule for switching among time optimal controllers in a multiple vehicle intercept problem, Automatica 87 (2018) 274–280.

[19] A.A. Munishkin, D. Milutinović, D.W. Casbeer, Time efficient inspection of ground vehicles by a uav team using a markov inequality based rule, in: Proceedings of the 14th International Symposium on Distributed Autonomous Robotic Systems (DARS 2018), 2019, pp. 95–108.
[20] W.H. Fleming, H.M. Soner, Controlled Markov Processes and Viscosity Solutions, Springer Science+Business Media, 2006.
[21] H.J. Kushner, P. Dupuis, Numerical Methods for Stochastic Control Problems in Continuous Time, second ed., Springer-Verlag, New York, NY, USA, 2001.
[22] R.P. Anderson, E. Bakolas, D. Milutinović, P. Tsiotras, Optimal feedback guidance of a small aerial vehicle in a stochastic wind, J. Guid. Control Dyn. 36 (4) (2013) 975–985.
[23] J. Choi, D. Milutinović, Tips on stochastic optimal feedback control and bayesian spatio-temporal models: Application to robotics, ASME J. Dyn. Syst. Meas. Control 3 (137) (2015).
[24] R. Anderson, D. Milutinović, A stochastic approach to dubins vehicle tracking problems, IEEE Trans. Automat. Control 10 (59) (2014) 2801–2806.
[25] D. Milutinović, D.W. Casbeer, S. Rasmussen, D. Kingston, A 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 (CCTA), 2017.

Alexey Munishkin's 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 the 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.



Dr. Dejan Milutinović is a professor in the Department of Electrical and Computer engineering, University of California Santa Cruz. His research interests are in the area of modeling and control of stochastic dynamical systems applied to robotics. He 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. In the past he served as an associate editor for the ASME Journal of Dynamic Systems, and Control and IEEE Robotics and Automation Letters. Currently, he is a senior editor at-large of the Journal of Intelligent and Robotic Systems.

Dr. David Casbeer is the team lead for the UAV Cooperative and Intelligent Control Team within the Control Science Center of Excellence in the Air Force Research Laboratory's Aerospace Systems Directorate. The UAV team focuses on decision making, planning, and coordination for multiple autonomous UAVs acting and reacting in uncertain and adversarial environments. This team was awarded the AFOSR Star Team award for 2018–2020 for excellence in basic research. Dr. Casbeer received the B.S. and Ph.D. degrees from Brigham Young University in 2003 and 2009, respectively, where he advanced theory describing the statistics of decentralized estimation techniques. In 2016, Dr. Casbeer was awarded AFRL's Early Career Award for distinguished foundational research in multi-agent control. He served as the chair for the AIAA Intelligent Systems Technical Committee, and currently serves as a Senior Editor for the Journal of Intelligent and Robotic Systems and an editor for the AIAA Journal of Aerospace Information Systems.