Automatica 81 (2017) 253–260

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Sequential pursuit of multiple targets under external disturbances via Zermelo–Voronoi diagrams[☆]

Wei Sun, Panagiotis Tsiotras[1]

School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150, USA

ARTICLE INFO

Article history:
Received 25 November 2015
Received in revised form
17 February 2017
Accepted 18 February 2017

Keywords:
Pursuit-evasion
Zermelo–Voronoi diagram
Zermelo navigation problem
Pursuer–target assignment

ABSTRACT

We address the problem of pursuit between a collection of targets and a team of pursuers distributed in the plane subject to an environmental disturbance (e.g., wind, sea current). The objective of the pursuers is to intercept the moving targets which, however, are not affected by the presence of the flow field. We first solve the multiple-pursuers/single-target problem by assigning only one pursuer to chase the target at every instant of time, based on a Voronoi-like partition of the plane. During the pursuit, the pursuer assignment changes dynamically based on this partition. We then present an algorithm to efficiently update this Voronoi-like partition on-line. The results are then extended to the multiple-pursuers/multiple-targets case. Simulations are included to illustrate the theoretical results.

1. Introduction

Consider a scenario where a group of helicopters or small UAVs fly in the presence of a wind field and try to capture a group of vehicles moving on the ground, or a team of small marine or underwater vehicles attempting to intercept a ship which is large enough so that the sea currents do not significantly affect its motion. Given such a group of pursuers, we want to find a pursuit strategy to intercept the target(s) in minimum time. Problems of this nature fall under the general class of group pursuit. These are difficult problems to solve, in general (Blagodatskikh, 2008, 2009; Pittsyk & Chikrii, 1982). Their solution is also based on the information the pursuers and the targets/evaders have about each other, resulting in either cooperative or non-cooperative strategies (Bakolas & Tsiotras, 2012; Devillers & Golin, 1993; Krasovskiĭ & Krasovskiĭ, 1994; Pashkov & Terekhov, 1987; Vidal, Shakernia, Kim, Shim, & Sastry, 2002; Yeung & Petrosjan, 2006). In order to solve such problems, in this work we propose to use a *sequential* pursuit strategy (Bakolas & Tsiotras, 2012). By sequential (or relay) pursuit we mean that for each target, only one pursuer is assigned to chase this target at every instant of time. In addition to simplifying significantly the group pursuit problem, a relay pursuit strategy may be desirable in cases where the power or energy/fuel consumption of the agents is an important consideration, when the agents also play a dual role, both as pursuers as well as guardians protecting a certain area, or in order to account for possible deceptive strategies of an intelligent opponent.

For the multiple-pursuers/single-target problem, in contrast to most existing similar problem formulations in the literature (Bakolas & Tsiotras, 2012; Ibragimov, Salimi, & Amini, 2012; Jang & Tomlin, 2005; Khaidarov, 1984; Pshenichnyi, 1976), where the effect of the environment is not taken into consideration, in our problem setup (only) the pursuers are affected by known exogenous environmental conditions (e.g., the winds or currents). As with all pursuit games, the solution of this problem depends on the knowledge each pursuer has about the current and future position of the target. In this paper it will be assumed that each pursuer has a stroboscopic view of each target position. That is, each pursuer knows the current position of the target but it knows neither its velocity nor its future position. Our objective is to find the optimal assignment to determine which pursuer to go after which target at each instant of time so as to reduce, or minimize, the overall capture time. No assumptions about the target strategy are explicitly imposed a priori, other than the target moves with maximum speed, the value of which is known to all pursuers.

Our solution strategy is based on a dynamic assignment of the pursuers, by which the "best" pursuer to go after the target is selected according to a Voronoi-like partition of the plane, called

[☆] This work has been supported by NSF award 1160780 and AFOSR award FA9550-13-1-0029. The material in this paper was partially presented at the 52nd IEEE Conference on Decision and Control, December 10–13, 2013, Florence, Italy. This paper was recommended for publication in revised form by Associate Editor Andrey V. Savkin under the direction of Editor Ian R. Petersen.

E-mail addresses: wsun42@gatech.edu (W. Sun), tsiotras@gatech.edu (P. Tsiotras).

[1] Fax: +1 404 894 2760.

http://dx.doi.org/10.1016/j.automatica.2017.03.015
0005-1098/© 2017 Elsevier Ltd. All rights reserved.

the Zermelo–Voronoi partition, or the Zermelo–Voronoi Diagram (ZVD) (Bakolas & Tsiotras, 2010). The major difficulty in this setting arises from the fact that, owing to the presence of the wind field, a point in the plane can be close to a pursuer in terms of Euclidean distance, but it may not be close in terms of minimum-time to intercept. As a result, standard Voronoi partitions for this problem may lead to erroneous conclusions.

The ZVD is based on time-to-intercept as the relevant distance metric, instead of the Euclidean metric used in the standard Voronoi diagram. Such Voronoi-like diagrams have been previously introduced in Bakolas and Tsiotras (2010, 2012, 2013). For instance, in Bakolas (2014) the author constructs a Voronoi-like partition in a spatiotemporal flow field by taking the proximity metric as the time required for each vehicle to reach an arbitrary point using a line-of-sight (LOS) control law. Furthermore, as it is shown later on, the use of the so-called Zermelo Navigation Law (Eqs. (6)–(7) below) leads to a better solution (in terms of capture time) than one that uses a classical LOS control law.

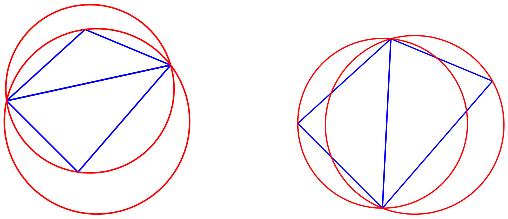
Voronoi-like diagrams have been used in the past to solve group pursuit problems in the plane (Bakolas & Tsiotras, 2010, 2012). In Bakolas and Tsiotras (2012), the authors proposed a relay pursuit problem where a group of pursuers aims at capturing a target that follows a specific evading strategy. The assignment problem is solved by dynamically assigning the pursuer whose Voronoi cell contains the target. Although our work follows closely the original work in Bakolas and Tsiotras (2012), where ZVDs were first employed to generate the pursuer assignments in an external wind field, our work differs from – and extends the results of – this work as follows: First, and contrary to Bakolas and Tsiotras (2012), we take into account the known disturbance of the environment. Furthermore, this disturbance affects only the pursuers, thus leading to asymmetric dynamics between the target and the pursuers. The asymmetry of the agents' dynamics does not allow us to use a common reference frame to solve the problem, as was done, for example, in Bakolas and Tsiotras (2010). Second, we use a somewhat different Voronoi-like partition than the one in Bakolas and Tsiotras (2012). Owing to the fact that the target and the pursuers obey different dynamics, in our case we cannot check the condition in Bakolas and Tsiotras (2012) to update the ZVD; instead, we need to update the ZVD continuously. In the process, we propose a numerically efficient algorithm to update the ZVD on-the-fly that may be of independent interest (see Section 5). Third, we assume minimal knowledge of the target state, namely, only its instantaneous position is known to the pursuers. In Bakolas and Tsiotras (2012), on the other hand, it was assumed that the target implements a specific evading strategy, which is known to all of the pursuers.

Contributions. The major contributions of the paper are summarized below:

(a) We provide a new formulation for the solution of an *asymmetric* group pursuit problem using the recently introduced concept of Zermelo–Voronoi diagram (ZVD). The formulation leads to a decentralized solution of the original group pursuit problem by decomposing it to a sequence of simpler pursuit problems that are much easier to solve.

(b) We provide conditions so that the well-known Zermelo Navigation Law (ZNL) achieves capture against a maneuvering target. This key result allows us to implement the ZNL in a sequential manner against arbitrarily maneuvering targets and in the presence of unknown exogenous drift fields.

(c) We propose a decentralized algorithm for updating efficiently the ZVD as the pursuit evolves based on the dual of the Voronoi diagram, that is, the Delaunay Triangulation (DT) and we provide a complexity analysis for updating the DT (and hence also the ZVD). The updated ZVD is then used to provide the best assignments of the active pursuer(s).



(a) A triangulation of 4 points. (b) DT of the same points.

Fig. 1. Flip-edge method for generating the Delaunay triangulation.

(d) One of the major benefits of the proposed approach is that it scales well with the number of pursuers and the targets involved. We propose two algorithms that address multiple-pursuer/multi-target problems that make use of this nice property of the proposed ZVD decomposition.

2. Preliminaries on plane tessellations

Given a finite number of distinct points in the Euclidean plane, called the *generators*, we associate their locations with a set of points in the plane, such that each point in this set is closer (with respect to a given distance metric) to its own generator than to any other generator. The result is a tessellation of the plane into a set of regions associated with the given generators (Okabe, Boots, Sugihara, & Chiu, 2009).

2.1. Delaunay triangulation

Given an (ordinary) Voronoi diagram of a point set in a generic configuration (that is, no three points are on the same line and no four points lie on the same circle), we may join all pairs of generators whose Voronoi cells share a common edge. We thus obtain a second tessellation consisting of only triangles, called the *Delaunay triangulation* (DT) of VD. The Delaunay triangulation is the dual graph of the Voronoi diagram.

A circle circumscribing any Delaunay triangle contains no generator in its interior (Okabe et al., 2009). This is the *Delaunay property*. Given any triangulation of a prescribed point set, we can construct the DT by flipping the edges until no triangle violates the Delaunay property. This method of generating the DT of a given point set is called the *flip-edge method* (Roos, 1993) (see Fig. 1).

2.2. Zermelo–Voronoi diagram

The problem of obtaining the proximity relations between a set of agents can often be recast as a set membership problem. For instance, the question of determining which of the agents is closest (in terms of arrival time) to a static target at a particular instant of time, reduces to a set membership problem, namely, one of forming the so-called Zermelo–Voronoi Diagram (ZVD) (Bakolas & Tsiotras, 2010), and then finding the cell in which the target resides at the given time instant.

Given a finite number of agents at some time t, the ZVD is a partition of the plane whose generalized distance is the minimum time for the corresponding Zermelo navigation problem (Bryson & Ho, 1975) from each agent's current position (the generator) to the agent's terminal configuration.

To this end, let the index set $I = \{1, 2, \dots, n\}$, and consider n agents starting from distinct initial positions $X^i_{P_0} \in \mathbb{R}^2$, $i \in I$, whose dynamics are given by

$$\dot{X}^i = u^i + w(X^i, t), \qquad X^i(0) = X^i_{P_0}, \quad i \in I, \tag{1}$$

where $X^i = [x^i, y^i]^\top \in \mathbb{R}^2$ denotes the position of the ith agent, $u^i \in \mathbb{R}^2$ is the control input of the ith agent and $w(X^i, t) \in \mathbb{R}^2$ represents the environmental disturbance (winds/currents). Given $X \in \mathbb{R}^2$, let $T^i_* = T^i_*(X^i_{P_0}, X)$ denote the minimum time such that $X^i(T^i_*) = X$, subject to the dynamics (1). As shown in Bryson and Ho (1975), the control law u^i_{ZN} that solves this minimum-time problem is the Zermelo navigation law (ZNL) given in Eqs. (6)–(7) below.

Definition 2.1 (*Zermelo–Voronoi Diagram Bakolas & Tsiotras, 2010*). The *Zermelo–Voronoi diagram (ZVD)*[2] (or *Zermelo–Voronoi partition*) at time $t \geq 0$, associated with the system (1), is a set partition of the plane $\mathcal{Z}^t = \{Z^t_1, Z^t_2, \ldots, Z^t_n\}$ such that

(i) $\mathbb{R}^2 = \bigcup_{i=1}^n Z^t_i$,
(ii) $\mathrm{cl}\, Z^t_i = Z^t_i$ for all $i \in I$.
(iii) $\mathrm{int}\, Z^t_i \cap \mathrm{int}\, Z^t_j = \varnothing$ for all $i, j \in I$ such that $i \neq j$.
(iv) For each $X \in \mathrm{int}\, Z^t_i$, it follows that $T^i_*(X^i(t), X) < T^j_*(X^j(t), X)$ for $i \neq j$.

The sets Z^t_i are the *Zermelo–Voronoi cells* at time t. The cell Z^t_i consists of all points that can be reached by the ith agent faster than by any other agent, assuming that all agents use the same control law u^i_{ZN}.

The following proposition characterizes a useful property of the ZVD that will be used later on.

Proposition 2.2 (Bakolas & Tsiotras, 2010). *Let $\mathcal{V} = \{V_i, i \in I\}$, be the partition of the ordinary Voronoi diagram with generators $P = \{P_i, i \in I\}$. Assume that the dynamics of each agent initially placed at the generator positions are given by (1), and assume that $w = w(t)$. Let the one-to-one, continuous function $F : \mathbb{R}^2 \to \mathbb{R}^2$ be defined by $F(X) = f_{P_i}(X)$, $X \in V_i$, $i \in I$, where $f_{P_i}(X) = X + \int_0^{|X-P_i|} w(\tau)\,\mathrm{d}\tau$, $i \in I$. Then $Z_i = F(V_i)$ and thus ZVD is the image of VD under the mapping F.*

In other words, when the external field is time-varying (but not spatially varying) there exists a homeomorphism between the ordinary Voronoi diagram and the Zermelo–Voronoi diagram with the same generators. This results in fast numerical algorithms for generating Zermelo–Voronoi partitions using well-established algorithms for generating ordinary Voronoi partitions from computational geometry (Okabe et al., 2009).

3. Problem setup

In this section, we formulate the dynamic pursuit problem with multiple pursuers and a single target. Section 6 provides an extension of the methodology to problems with multiple targets. To this end, consider a group of n pursuers in the plane indexed by the set I, and assume that at time $t = 0$ the pursuers are located at n distinct positions in the plane, designated by $P_0 = \{X^i_{P_0} \in \mathbb{R}^2, i \in I\}$. The kinematics of the pursuers are described by

$$\dot{X}^i_P = u^i_P + w(t), \qquad X^i_P(0) = X^i_{P_0}, \quad i \in I, \tag{2}$$

where $X^i_P := [x^i_P, y^i_P]^\top \in \mathbb{R}^2$ denotes the position of the ith pursuer, $u^i_P \in \mathbb{R}^2$ is the control input of the ith pursuer such that $u^i_P \in \mathcal{U}_P$, for all $i \in I$, and $w(t) \in \mathbb{R}^2$ represents the wind disturbance. The set $\mathcal{U}_P$ consists of all piecewise continuous functions whose range is included in the set $U_P = \{u \in \mathbb{R}^2, |u| \leqslant \bar{u}\}$. It is assumed, furthermore, that there exists $0 < \bar{w} < \bar{u}$ such that $|w(t)| \leqslant \bar{w}$, for all $t \geqslant 0$. The restriction on the magnitude of the wind disturbance is imposed in order to ensure complete pursuer controllability, namely, that the pursuers are able to reach any point on the plane in finite time. The absence of controllability leads to complicated behavior and requires a more detailed analysis (Bakolas & Tsiotras, 2010, 2013).

The objective of the pursuers is to intercept a target, whose kinematics is given by

$$\dot{X}_T = u_T, \qquad X_T(0) = X_{T_0}, \tag{3}$$

where $X_T = [x_T, y_T]^\top \in \mathbb{R}^2$ is the position of the target, and u_T is its control input such that $u_T \in \mathcal{U}_T$, which consists of all piecewise continuous functions whose range is included in the set $U_T = \{u \in \mathbb{R}^2, |u| \leqslant \bar{q}\}$. Note that the target is not affected by the wind field.

Remark 1. It is worth mentioning that the models (2) and (3) also capture cases when the pursuers and the target are subject to different disturbances. To see this, assume that the governing equations for the pursuers and the target are given, respectively, by

$$\dot{X}^i_P = u^i_P + w_1(t) \quad \text{and} \quad \dot{X}_T = u_T + w_2(t). \tag{4}$$

By letting the new states $Z^i_P(t) = X^i_P(t) - \int_0^t w_2(\tau)\,\mathrm{d}\tau$ and $Z_T(t) = X_T(t) - \int_0^t w_2(\tau)\,\mathrm{d}\tau$ one obtains the system of differential equations for Z^i_P and Z_T that are of the same form as in (2) and (3), respectively, with $w(t) = w_1(t) - w_2(t)$.

Throughout this work, we assume that the pursuers do not know how the target maneuvers a priori. Instead, they have accurate measurements of the current position of the target at every instant of time. One reasonable strategy for each pursuer is therefore to use the Zermelo navigation law (Bakolas & Tsiotras, 2012; Bryson & Ho, 1975) in order to intercept the target. As discussed in Bakolas and Tsiotras (2010), starting at time $t = 0$, the optimal time of arrival T^i_{ZN} of the ith pursuer from $X^i_{P_0}$ to X_{T_0} is given by

$$T^i_{\mathrm{ZN}} = \min\left\{T > 0 : \bar{u}T - |X_{T_0} - X^i_{P_0} - \int_0^T w(\tau)\,\mathrm{d}\tau| = 0\right\}. \tag{5}$$

Then Zermelo's navigation control can be obtained by

$$u^i_{\mathrm{ZN}} = \bar{u}(\cos\theta^*_i, \sin\theta^*_i)^\top, \tag{6}$$

where

$$\theta^*_i = \mathrm{Arg}\left(X_{T_0} - X^i_{P_0} - \int_0^{T^i_{\mathrm{ZN}}} w(\tau)\,\mathrm{d}\tau\right), \quad i \in I. \tag{7}$$

3.1. Pursuer assignment

We only consider sequential, or relay, pursuit strategies. That is, we assume that, at every instant of time, only a single pursuer is chasing the target, i.e., at every time $t \geqslant 0$, there exists a unique $i \in I$, such that $u^i_P(t) = u^i_{\mathrm{ZN}}(t)$, whereas $u^j_P(t) = 0$, for all $j \in I, j \neq i$. In such a case, we will call i the *active pursuer* at time t. Let $\mathcal{A}^t$ denote the set of active pursuers at time t. Clearly, for a sequential pursuit the set $\mathcal{A}^t$ is a singleton. Our goal is to find a sequence of active pursuers to capture the target in the shortest possible time. To this end, define a mapping $\sigma : [0, \infty) \mapsto I$, where σ belongs to the set of all the right continuous, piecewise constant functions, denoted as Σ, such that $\sigma(t) = i$ if and only if $i \in \mathcal{A}^t$. We call σ the *assignment function* of the target.

The pursuer–target assignment problem can then be stated as follows: Given a target and a set of pursuers in the plane, determine an assignment function $\sigma_{\min} \in \Sigma$ to minimize capture time under the assumption that each pursuer is using a stroboscopic strategy based on the target's current location.

[2] Note that in Bakolas and Tsiotras (2010) this is referred to as the dual Zermelo–Voronoi diagram, not to be confused with the dual graph of the Zermelo–Voronoi diagram.

4. Analysis and implementation of the pursuer–target assignment problem

Before proceeding with the solution of the optimal pursuer assignment problem, we first need to determine the conditions on the target's maneuverability such that there exists an assignment function leading to finite capture time. Below we provide a sufficient condition for the existence of capture time.

The *robust line-of-sight* navigation law (RLOS) steers a pursuer towards a target at every instant of time, while maximizing the speed along the ensuing path. This is the optimal strategy, among all control strategies that force the pursuer to move along the current line-of-sight (Bakolas & Tsiotras, 2012). The RLOS navigation law of the ith pursuer can be expressed as Bakolas and Tsiotras (2012)

$$u^i_{\mathrm{RLOS}}(t, Y^i) = \sqrt{\bar{u}^2 - \langle w(t), e^i_2(t)\rangle^2}\, e^i_1(t) - \langle w(t), e^i_2(t)\rangle e^i_2(t), \tag{8}$$

where $Y^i(t) = X_T(t) - X^i_P(t)$ is the vector from the pursuer to the target, $e^i_1(t) = Y^i(t)/|Y^i(t)|$ and $\langle e^i_2(t), e^i_1(t)\rangle = 0$ for all $i \in I$ and $t \geq 0$.

The following result is adapted from Bakolas and Tsiotras (2012).

Proposition 4.1. *Let $\epsilon > 0$, and assume that the dynamics of each pursuer is given by (2) and the dynamics of the target is given by (3). Then, for each pursuer i, and for all initial conditions $X^i_{P_0}$ and X_{T_0}, there exists a finite time $T^i_{\mathrm{RLOS}}(X^i_{P_0}, X_{T_0}) \geqslant 0$ such that the ith pursuer driven by the RLOS navigation law (8) enters the set $\{X \in \mathbb{R} : |X - X_T(T^i_{\mathrm{RLOS}})| \leqslant \epsilon\}$, provided that*

$$|\langle w(t) - u_T(t), e^i_1(t)\rangle| < \sqrt{\bar{u}^2 - \bar{w}^2}, \quad \textit{for all } t \geqslant 0. \tag{9}$$

The following corollary is immediate from Proposition 4.1.

Corollary 4.2. *Assume that (9) holds for all $i \in I$. Any sequential pursuit strategy in which each active pursuer employs the RLOS navigation law (8) leads to capture of the target by at least one pursuer.*

Proposition 4.1 implies that the minimum-time intercept problem using Zermelo's navigation law in (6)–(7) always has a solution, for all initial conditions for the pursuers and the target. Furthermore, applying Zermelo's navigation law instead of (8) results in a smaller intercept time, that is, $t^i_{\mathrm{ZN}} \leq t^i_{\mathrm{RLOS}}$ for all $i \in I$. This, in turn, suggests that a sequential strategy that uses Zermelo's navigation law for each pursuer will eventually lead to capture. By imposing an alternative condition we can actually prove the following result.

Proposition 4.3. *Let $\epsilon > 0$, and assume that the dynamics of each pursuer is given by (2) and the dynamics of the target is given by (3). Furthermore, assume that there exists $\tilde{w} > 0$ such that, for all $t \geq 0$ and $T > 0$,*

$$\frac{1}{T}\int_t^{t+T} w(\tau)\,\mathrm{d}\tau < \tilde{w}. \tag{10}$$

Then, for each pursuer i, and for all initial conditions $X^i_{P_0}$ and X_{T_0}, there exists a finite time $t^i_{\mathrm{ZN}}(X^i_{P_0}, X_{T_0}) \geqslant 0$ such that the ith pursuer driven by the robust Zermelo navigation law enters the set $\{X \in \mathbb{R} : |X - X_T(t^i_{\mathrm{ZN}})| \leqslant \epsilon\}$ provided that

$$\bar{q} < \bar{u} - \tilde{w}. \tag{11}$$

Proof. Let $Y^i(t) = X_T(t) - X^i_P(t)$ be the vector from the ith pursuer to the target. Assume that at time $t = t_k$ the ith pursuer and the target are located at positions $X^i_P(t_k)$ and $X_T(t_k)$ respectively. It follows from (5) that the time-to-intercept of a stationary target at $X_T(t_k)$ is given by $T^i_{\mathrm{ZN}_k} = \min\{T > 0 : \bar{u}T - |Y^i(t_k) - \int_{t_k}^{t_k+T} w(\tau)\,\mathrm{d}\tau| = 0\}$. In particular,

$$\left|Y^i(t_k) - \int_{t_k}^{t_k+T^i_{\mathrm{ZN}_k}} w(\tau)\,\mathrm{d}\tau\right| = \bar{u}T^i_{\mathrm{ZN}_k}, \tag{12}$$

and the corresponding optimal control at t_k is given by

$$u^i_{\mathrm{ZN}}(t_k) = \frac{1}{T^i_{\mathrm{ZN}_k}}\left(Y^i(t_k) - \int_{t_k}^{t_k+T^i_{\mathrm{ZN}_k}} w(\tau)\,\mathrm{d}\tau\right). \tag{13}$$

At the next time step $t = t_k + \delta t$, we get

$$\begin{aligned} X^i_P(t_k + \delta t) = X^i_P(t_k) &+ \int_{t_k}^{t_k+\delta t} w(\tau)\,\mathrm{d}\tau \\ &+ \frac{\delta t}{T^i_{\mathrm{ZN}_k}}\left(Y^i(t_k) - \int_{t_k}^{t_k+T^i_{\mathrm{ZN}_k}} w(\tau)\,\mathrm{d}\tau\right), \end{aligned}$$

and, similarly, $X_T(t_k + \delta t) = X_T(t_k) + u^k_T\delta t$, where $u^k_T = u_T(t_k)$. Thus,

$$\begin{aligned} Y^i(t_k + \delta t) = Y^i(t_k) + u^k_T\delta t &- \int_{t_k}^{t_k+\delta t} w(\tau)\,\mathrm{d}\tau \\ &- \frac{\delta t}{T^i_{\mathrm{ZN}_k}} Y^i(t_k) + \frac{\delta t}{T^i_{\mathrm{ZN}_k}} \int_{t_k}^{t_k+T^i_{\mathrm{ZN}_k}} w(\tau)\,\mathrm{d}\tau. \end{aligned} \tag{14}$$

The time-to-intercept at time step $t = t_k + \delta t$ is given by $T^i_{\mathrm{ZN}_{k+1}} = \min\{T > 0 : \bar{u}T - |Y^i(t_k + \delta t) - \int_{t_k+\delta t}^{t_k+\delta t+T} w(\tau)\,\mathrm{d}\tau| = 0\}$. In particular,

$$|Y^i(t_k + \delta t) - \int_{t_k+\delta t}^{t_k+\delta t+T^i_{\mathrm{ZN}_{k+1}}} w(\tau)\,\mathrm{d}\tau| = \bar{u}T^i_{\mathrm{ZN}_{k+1}}. \tag{15}$$

Pick $\delta t = \epsilon/(\bar{u} + \bar{w})$, where $\epsilon > 0$ and assume that there exists $k > 0$ such that $T^i_{\mathrm{ZN}_k} \leq \delta t$. Then from (13) one obtains $|Y^i(t_k)| \leq |u^i_{\mathrm{ZN}}(t_k)|T^i_{\mathrm{ZN}_k} + |\int_0^{T^i_{\mathrm{ZN}_k}} w(\tau)\mathrm{d}\tau| \leq (\bar{u} + \tilde{w})T^i_{\mathrm{ZN}_k} \leq (\bar{u} + \tilde{w})\delta t \leq \epsilon$. This shows that the pursuer is in the ϵ ball centered at the evader's position and capture has occurred.

Suppose now that $T^i_{\mathrm{ZN}_k} > \delta t$ for all $k > 0$. In this case, using (14), and after some algebraic manipulations, one obtains the following expression for the term on the left-hand-side of (15)

$$\begin{aligned} &Y^i(t_k + \delta t) - \int_{t_k+\delta t}^{t_k+\delta t+T^i_{\mathrm{ZN}_{k+1}}} w(\tau)\,\mathrm{d}\tau \\ &\quad = \left(1 - \frac{\delta t}{T^i_{\mathrm{ZN}_k}}\right)\left(Y^i(t_k) - \int_{t_k}^{t_k+T^i_{\mathrm{ZN}_k}} w(\tau)\,\mathrm{d}\tau\right) \\ &\quad\quad - \int_{t_k+T^i_{\mathrm{ZN}_k}}^{t_k+\delta t+T^i_{\mathrm{ZN}_{k+1}}} w(\tau)\,\mathrm{d}\tau + u^k_T\delta t. \end{aligned}$$

Letting $\Theta_k = Y^i(t_k) - \int_{t_k}^{t_k+T^i_{\mathrm{ZN}_k}} w(\tau)\,\mathrm{d}\tau$, and

$$H_k = -\int_{t_k+T^i_{\mathrm{ZN}_k}}^{t_k+\delta t+T^i_{\mathrm{ZN}_{k+1}}} w(\tau)\,\mathrm{d}\tau + u^k_T\delta t, \tag{16}$$

Eq. (15) can be written as $|\alpha\Theta_k + H_k| = \bar{u}T^i_{\mathrm{ZN}_{k+1}}$, where $\alpha = 1 - \delta t/T^i_{\mathrm{ZN}_k} > 0$, whereas (12) can be written $|\Theta_k| = \bar{u}T^i_{\mathrm{ZN}_k}$.

Subtracting the last two equations yields $-\bar{u}\Delta T_k = |\Theta_k| - |\alpha\Theta_k + H_k|$, where $\Delta T_k = T^i_{\mathrm{ZN}_{k+1}} - T^i_{\mathrm{ZN}_k}$. We claim that $\Delta T_k < 0$ for all $k > 0$ such that $T^i_{\mathrm{ZN}_k} > \delta t$. To this end, note that using the triangle inequality, yields $-\bar{u}\Delta T_k \geq |\Theta_k| - \alpha|\Theta_k| - |H_k| = \bar{u}\delta t - |H_k|$, where we have made use of the fact that $(1-\alpha)|\Theta_k| = \delta t/T^i_{\mathrm{ZN}_k}|\Theta_k| = \bar{u}\delta t$. Hence, $|H_k| \geq \bar{u}(\Delta T_k + \delta t)$. From (10) and (16) we also have $|H_k| = |u^k_T\delta t - \int_{t_k+T^i_{\mathrm{ZN}_k}}^{t_k+\delta t+T^i_{\mathrm{ZN}_{k+1}}} w(\tau)\,\mathrm{d}\tau| \leq \bar{q}\delta t + \tilde{w}|\Delta T_k + \delta t|$. Thus, we get

$$\bar{u}(\Delta T_k + \delta t) \leq |H_k| \leq \bar{q}\delta t + \tilde{w}|\Delta T_k + \delta t|. \tag{17}$$

If $\Delta T_k \geq 0$, for some $k > 0$ such that $T^i_{\mathrm{ZN}_k} > \delta t$, then it follows from the previous expression that $\bar{u}(\Delta T_k + \delta t) \leq \bar{q}\delta t + \tilde{w}(\Delta T_k + \delta t)$ or that $(\tilde{w} - \bar{u})\Delta T_k \geq (\bar{u} - \bar{q} - \tilde{w})\delta t$, which leads to a contradiction since the left-hand side of the previous inequality is non-positive and the right-hand side is positive. It follows that $T^i_{\mathrm{ZN}_{k+1}} - T^i_{\mathrm{ZN}_k} = \Delta T_k < 0$ as claimed. This inequality implies that the sequence $\{T^i_{\mathrm{ZN}_k}\}_{k=1}^{\infty}$ is strictly decreasing, and since it is also bounded from below, it converges. Hence, $\lim_{k\to\infty}\Delta T_k = 0$. Taking the limit as $k \to \infty$ in (17) yields $\bar{u}\delta t \leq (\bar{q} + \bar{w})\delta t$ or that $\bar{u} \leq \bar{q} + \tilde{w}$, contradicting (11). ■

Note that, in general, $\tilde{w} \leq \bar{w}$ and the two bounds are equal for the case of a constant external field. The condition (11) is rather restrictive since it implies that all pursuers dominate the target even in the presence of the external field. However, this condition is *necessary* in case the wind field is not known a priori and one needs to ensure capture against any exogenous field of (only) known upper bound. To see this, just consider the extreme case where a single pursuer goes after a single target under a constant wind drift $w = \bar{w}(X_{P_0} - X_{T_0})/|X_{P_0} - X_{T_0}|$.

The next corollary follows immediately from Proposition 4.3.

Corollary 4.4. *Assume that (11) holds for all $i \in I$. Any sequential pursuit strategy in which each active pursuer employs Zermelo's navigation law (6)–(7) leads to capture of the target by at least one pursuer.*

At this point, it is important to compare Corollaries 4.2 and 4.4. None of the conditions (9) and (11) imply the other, although condition (11) is, in general, less restrictive than (9). However, owing to the optimality of the ZNL it is clear that (9) (with $u_T = 0$) should be sufficient for capture even for the case of ZNL. Although capture is ensured in both cases using a sequential pursuit strategy, in light of the discussion immediately after Corollary 4.2, it should be clear that the use of the sequential ZNL strategy leads to a shorter capture time than the use of a sequential RLOS strategy. This is confirmed in Section 7 where numerical examples comparing the two alternative relay pursuit strategies (that is, the RLOS and the ZNL) are presented. The fact that for stationary target the ZNL leads to capture is trivial since the ZNL is the optimal strategy. Ensuring, however, that the repeated use of the ZNL against an unknown maneuvering target will lead to capture requires additional assumptions along with a more detailed analysis, which is provided as part of the proof of Proposition 4.3.

Armed with Corollary 4.4 and similarly to Bakolas and Tsiotras (2012), we propose a sequential (or relay) pursuit assignment strategy based on the target's membership in the Zermelo–Voronoi cell of one of the pursuers. Without loss of generality, we assume that $X_T(0) \in \operatorname{int} Z^0_i$ for some $i \in I$. Assuming that each active pursuer chooses the Zermelo navigation law, we may now propose the following algorithm to assign the active pursuer:

Case 1: Multiple-pursuers/single-target

(a) For each $t \geq 0$, set $\sigma(t) = i$ if $X_T(t) \in \operatorname{int} Z^t_i$.
(b) If $\sigma(t^-) = i$ and $X_T(t) \in \operatorname{int} Z^t_j$ with $i \neq j$, let $\sigma(t) = j$.
(c) If $|X^i_P - X_T(t)| > \epsilon$, go to step (b). Otherwise, terminate the procedure and return the assignment function σ.

In other words, the algorithm at each time $t \geq 0$, assigns the ith pursuer to be the active pursuer if the target resides in the interior of the corresponding Zermelo–Voronoi cell Z^t_i at time t.

5. Update algorithm for ZVD

In order to implement the previous algorithm we need, at every instant of time, to know the ZVD in order to determine which Zermelo–Voronoi cell the target resides in. We can either build a ZVD from scratch at each time step, or update the ZVD from the previous time step. Since at every time interval, only one generator moves relatively to the rest, it is expected that the latter option will be more efficient. Hereby, we present an algorithm that updates the ZVD from one time step to the next when only a single pursuer has moved.[3]

Utilizing Proposition 2.2, our strategy for updating the ZVD is to update the ordinary VD corresponding to the same generators first and then form the ZVD through this transformation. However, in order to update the ordinary VD we will, instead, update its dual graph, namely, its Delaunay Triangulation. We use a modification of the algorithm introduced in Guibas and Russel (2004) for updating DT since it is relatively efficient and it fits our problem.

In order to update the DT from the previous time step to the current time step, a straightforward way would be to put all the points in a queue and every time we push a point out from the queue, we remove this point from the original triangulation and then insert it back at the new location at the present time (Kao, 1991). Each deletion and insertion of the DT preserves the Delaunay property, so the procedure would yield a valid DT. However, the procedure is not very efficient since even if all the points remain static during the time interval, we still need to delete and insert all the points to complete the update. Moreover, removing a point from a DT is a fairly expensive process. Given the previous considerations, we propose an alternative approach to deal with moving generators, which most of the times leads to an efficient updating of the DT.

We want the update algorithm to take advantage of the fact that part of the DT structure has not changed from the previous time step. To this end, denote by DT$_k$, and DT$_{k+1}$ the Delaunay Triangulation at time steps t_k and t_{k+1} respectively. Assume that the corresponding generator sets are given by P_k and P_{k+1}. Our goal is to update DT$_k$ into DT$_{k+1}$ with as few deletions as possible. To this end, we first check whether we can generate DT$_{k+1}$ from DT$_k$ using only the flip-edge method. The flip-edge method can be applied when DT$_k$ is an embedding (Guibas & Russel, 2004). Recall that, given a point set, a triangulation is an embedding if the triangulation associated with this point set has no overlapping triangles. If a triangulation is not an embedding, we say that it is an unembedding. Fig. 2(a) shows a DT associated with a given point set, and Fig. 2(b) shows the DT associated with a new point set, where point 5 has changed its location. Some triangles overlap with each other in Fig. 2(b). Thus, the DT in Fig. 2(b) is an unembedding. Also notice that if an unembedding occurs, there exists at least one triangle that has changed its orientation. For example, the triangle with vertices 3, 4, and 5 in Fig. 2(a) has a clockwise orientation.

[3] This algorithm can also be applied to the case where more than one generator moves (see Section 6).

In Fig. 2(b), on the other hand, the orientation of the triangle with the same vertices has counter-clockwise orientation, i.e., its orientation has changed.

We introduce the *orientation certificate* to check whether DT_k is an embedding or not (Guibas & Russel, 2004). If the orientation certificate is passed, we can simply use the flip-edge method to update the DT. Otherwise, we need to remove the points that cause the unembedding and then check the orientation certificate until it is passed. After this iteration, we obtain a triangulation with no overlaps, and we can then use the flip-edge method to transform it into a DT. Finally, we insert the removed points to their current locations and form DT_{k+1}. The difference between this procedure and the method introduced in Guibas and Russel (2004) is that in the latter reference the authors remove the points randomly to get the triangulation candidate, whereas in our case we only remove the moving points since they are the only possible generators that may cause unembedding.

The algorithm for updating the Zermelo–Voronoi diagram from the previous time step to current time step is given in Algorithm 1.

Algorithm 1 Update Zermelo–Voronoi Diagram

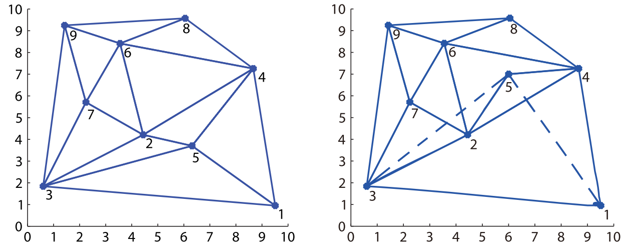
Input: Coordinates P_{k-1} of the generators at the previous time step and the corresponding Delaunay triangulation DT, coordinates P_k of point set at current time step.

Output: Updated Zermelo–Voronoi Diagram and Delaunay Triangulation at the current time step.

```
procedure UPDATE_DT(DT,$P_{k-1}$,$P_k$)
    while the triangulation DT is not embedded under current coordinates $P_k$ do
        Update DT by removing one of the points that cause the unembedding (in our case the active pursuer);
        store the current coordinates of removed points into the set $R$;
    end while
    if $R$ is not empty then
        flip the remaining triangulation into a Delaunay triangulation;
    end if
    for $i = 0$ to $length(R)$ do
        Update DT by inserting the $i$th point in $R$ into the triangulation;
    end for
    Transform DT into an ordinary Voronoi diagram VD;
    Transform VD into the ZVD at current time through the coordinate transformation of Proposition 2.2.
    return ZVD and DT.
end procedure
```

5.1. Complexity analysis

The complexity of the previous algorithm is dominated by the deletion and insertion of generators during the ZVD update. To remove a point from the standard Delaunay Triangulation, we use the deletion method introduced in Devillers (1999) and to insert a point into the DT, we choose the algorithm given in Kao (1991). The complexity of deleting a generator through the method in Devillers (1999) is $O(k \log k)$, where k is the degree of the deleted generator (that is, the number of edges associated with this generator). When the generator is randomly picked, the expected value of k is 6, without any assumption on the point distribution (Devillers, 1999). Given a base set of points, any sequence of insertions to the Delaunay triangulation can be performed in expected amortized time $O(\log n)$ for each insertion (Kao, 1991), where an algorithm is said to have an amortized time complexity of $f(n)$ if the total cost of any sequence of N operations divided by N is $O(f(n))$. As a result, the overall complexity of the algorithm is of order $O(\log n)$ when $n \gg k$ or $O(k \log k)$ when n is close to k. This is better than the complexity $O(n \log n)$ of constructing the ZVD from scratch at each instant of time (Kao, 1991). The flip-edge algorithm introduced in De Berg, Cheong, Van Kreveld, and Overmars (2008) can also be used in practice, instead of the deletion and insertion of the generator, although it has worst case complexity $O(n^2)$. However, this is only achieved if $O(n^2)$ edges need to be flipped, which is rarely the case. Owing to the continuous movement of the active pursuer(s), much fewer edge flips are required to transform a DT in the previous time step to the DT in the current time step. In all our numerical simulations it was observed that the flip edge method performed consistently well irrespective of the number of targets.



(a) Delaunay Triangulation of 9 generators.

(b) Generator 5 changed its location and caused unembedding.

Fig. 2. Unembedding caused by relocation of a generator.

6. Sequential pursuit of multiple targets

In this section, we extend the previous results to the case of a pursuit problem with multiple pursuers and targets. To this end, consider a group of n pursuers in the plane, denoted by $\{P_1, P_2, \ldots, P_n\}$, and m targets, denoted by $\{T_1, T_2, \ldots, T_m\}$. Let $J = \{1, \ldots, m\}$ denote the index set of the targets. The objective of the pursuers is to intercept the targets. It is assumed that after a pursuer intercepts a target, it is still capable of going after other targets. The kinematics of the ith pursuer, $i \in I$, are described by (2) and the kinematics of the jth target is given by

$$\dot{X}_T^j = u_T^j, \qquad X_T^j(0) = X_{T_0}^j, \tag{18}$$

where $X_T^j = [x_T^j, y_T^j]^\top \in \mathbb{R}^2$ is the position of the jth target, and u_T^j is its control input such that $u_T^j \in \mathcal{U}_T^j$, which consists of all piecewise continuous functions whose range is included in the set $U_T^j = \{v \in \mathbb{R}^2, |v| \leqslant \bar{q}_j\}$. Note that the targets are not affected by the wind field.

At this point, we are ready to extend the previous pursuer assignment algorithm of the multiple-pursuers/single-target case to the problem with multiple targets. In particular, we present two methods to assign the pursuers to the targets.

Without loss of generality, we assume that none of the targets lie on the boundary of the ZVD at time $t = 0$ and thus each target has been initially assigned to one of the pursuers. Let the assignment function for the multi-target case be denoted by $\sigma = (\sigma_1, \ldots, \sigma_m) \in \Sigma^m$ where each σ_j is the assignment function corresponding to target $j \in J$. The two algorithms proceed as follows.

Algorithm 2a: Multiple-pursuers/multiple-targets case

(a) For each $t \geq 0$, and for each $j \in J$, let $a_j \in I$ be such that $X_T^j(t) \in \operatorname{int} Z_{a_j}^t$. In this case, set $\sigma_j(t) = a_j$. If $a_{j_1} = \cdots = a_{j_s} = i$ for some $j_1, \ldots, j_s \in J$ we set $\sigma_{j_k}(t) = i$ where i is such that

$T^i_{\text{ZN}}(X^i_P, X^{j_k}_T) \leq T^i_{\text{ZN}}(X^i_P, X^{j_\ell}_T)$ for all $\ell = 1, \ldots, s$, and we also set $\sigma_{j_\ell}(t) = 0$, for all $j_\ell \neq j_k$. That is, we assign to the jth target the pursuer corresponding to the a_jth ZV cell and if there are multiple targets inside Z_{a_j} we assign to the a_jth pursuer the target that is closest (in terms of intercept time) to the pursuer, whereas all other targets remain unassigned.

(b) At every time step, after we update the ZVD and the locations of the pursuers and the targets, we repeat the procedure outlined in step (a) and we check the distance between each target and its assigned pursuer. Specifically, for each $j \in J$, if $|X^{a_j}_P - X^j_T(t)| \leq \epsilon$, let $J = J \setminus \{j\}$. Check if $J \neq \varnothing$ in which case return to step (a).

Otherwise, the procedure is terminated and the assignment function σ is returned.

In this algorithm, when one pursuer is paired to multiple targets, the pursuer always goes after the target whose current location can be reached by this pursuer in the shortest amount of time under Zermelo's control law. The targets whose distances from the assigned pursuer become smaller than ϵ are removed from the target set and the pursuit continues with the remaining targets. The procedure terminates when the distances between all pursuers and the corresponding targets become smaller than ϵ.

This method may result in a case where one pursuer is assigned to multiple targets. Consequently, some targets may not be actively pursued until other targets have been captured and the respective active pursuers become available. When the number of pursuers is larger than the number of targets, we can avoid this situation by applying the following alternative algorithm.

Algorithm 2b: Multiple-pursuers/multiple-targets case

(a) For each $t \geq 0$, and for each $j \in J$, let $a_j \in I$ be such that $X^j_T(t) \in \operatorname{int} Z^t_{a_j}$. In this case, set $\sigma_j(t) = a_j$. If $a_{j_1} = \cdots = a_{j_s} = i$ for some $j_1, \ldots, j_s \in J$ we set $\sigma_{j_k}(t) = i$ where i is such that $T^i_{\text{ZN}}(X^i_P, X^{j_k}_T) \leq T^i_{\text{ZN}}(X^i_P, X^{j_\ell}_T)$ for all $\ell = 1, \ldots, s$, and we also set $\sigma_{j_\ell}(t) = 0$, for all $j_\ell \neq j_k$. If there exists an unassigned target at the end of this process, then go to (b); otherwise, go to (c).

(b) Generate the new ZVD from the set of unassigned pursuers at the end of step (a). For each target $j \in J$ with $\sigma_j(t) = 0$, repeat the procedure in (a).

(c) Check the distance between each target and its assigned pursuer. That is, for each $j \in J$, if $|X^{a_j}_P - X^j_T(t)| \leq \epsilon$, let $J = J \setminus \{j\}$. If $J \neq \varnothing$ return to step (a). Otherwise, terminate and return the assignment function σ.

In the previous algorithm each pursuer is assigned to a single target. If more than one pursuers are assigned to multiple targets, each of these pursuers is assigned to the target whose current location can be reached by the pursuer in the shortest time. All other assignments between this pursuer and the rest of the targets are discarded, and a new ZVD is generated excluding those pursuers and targets that have already been assigned. As with the first algorithm, the targets whose distances from the assigned pursuer get smaller than ϵ are removed from the target set and the pursuit continues with the remaining targets. The procedure terminates if the distances between all active pursuers and the corresponding targets become smaller than ϵ.

Both of the previous two algorithms have their own advantages and disadvantages. The first algorithm does not have a restriction on the number of pursuers, but when the number of pursuers is relatively large, multiple targets may still be assigned to one pursuer and the unassigned pursuers are not utilized to reduce the time-to-capture of all the targets. In the second algorithm, each target is chased by a single pursuer at every instant of time. This tends to reduce the time-to-capture of all the targets under the assumption of sequential pursuit.

The following result gives a sufficient condition for the multiple-pursuers/multiple-targets assignment scheme to terminate in finite time.

Corollary 6.1. *Assume that* $\max_{j \in J} \bar{q}_j < \bar{u} - \tilde{w}$. *Then any sequential pursuit strategy in which each active pursuer employs the robust Zermelo's navigation law* (6)–(7) *leads to capture of all the targets by at least one pursuer.*

Proof. Since $\max_{j \in J} \bar{q}_j < \bar{u} - \tilde{w}$, it follows that $\bar{q}_j < \bar{u} - \tilde{w}$, for all $j \in J$. By Corollary 4.4, each evader can be captured by at least one pursuer in finite time. Therefore, for a finite number of evaders, the sequential pursuit scheme terminates in finite time. ■

7. Simulation results

We consider a scenario with 12 pursuers and 3 targets. The initial positions of the targets are given by $X^1_{T_0} = [3, 5]$, $X^2_{T_0} = [6, 7]$, $X^3_{T_0} = [4.5, 6]$. Without loss of generality, we assume that each target moves in a straight line with velocity $[0.4, 0.6]^\top$. The pursuers are initially located at distinct positions determined by P_0 and are shown in Fig. 3. The wind field that affects the pursuers is given by $w(t) = [-0.2 - 0.2\cos(t), 0.3]^\top$.

Fig. 3 depicts the trajectories of the pursuers and targets when the first algorithm in Section 6 is applied.[4] The pursuit process can be summarized as follows. At the start of the pursuit, Targets 1 and 3 are paired with Pursuer 2 and Target 2 is paired with Pursuer 4. Pursuer 2 goes after Target 1 in the beginning since the time it takes for Pursuer 2 to arrive at the location where Target 1 resides in is smaller than the time to arrive at the initial location of Target 3. Pursuer 4 chases Target 2 since it is the only target paired with this pursuer. At time $t_1 = 0.75$, Target 3 switches its assignment from Pursuer 2 to 5, because it enters the Zermelo–Voronoi cell of Pursuer 5 at this time step. At time $t_2 = 1.60$, the original Target 1 is captured by Pursuer 2 and removed from the target set. One of the two remaining targets is captured at time $t_3 = 1.80$. The last evading target is eventually captured at $t_4 = 4.65$ by Pursuer 5.

In order to compare the robust ZN law proposed in this paper with the naïve RLOS navigation law, we repeat the previous example where now the ZNL is replaced with the RLOS strategy. The time-to-capture under this condition is $t_4 = 5.0$, which is larger than the previous time-to-capture when the robust ZN law is applied ($t_4 = 4.65$).

The pursuit process under the second algorithm[5] is presented in Fig. 4. A summarized description of the pursuit process is given as follows. At $t = 0$ Pursuers 2, 4 and 5 are assigned to Targets 1, 2 and 3, respectively. Both Targets 1 and 3 are in the Zermelo–Voronoi cells of Pursuers 2 at $t = 0$, and Pursuer 2 is assigned to Target 1 since it can reach the location of Target 1 faster than that of Target 3 via the use of Zermelo's navigation law. Hence, Target 3 is assigned to one of the pursuers other than Purser 2 or 4. This leads to assigning Target 3 to Pursuer 5. Target 1 is captured by Pursuer 2 at $t_1 = 1.60$ and Target 2 is captured by Pursuer 4 at $t_2 = 1.80$. The last target (Target 3) is captured by Pursuer 5 at $t_3 = 3.55$.

For this problem a RLOS strategy leads to capture at $t_4 = 3.6$, which is, again, larger than the capture time resulting from the use of the sequential ZNL.

8. Conclusions

Under the assumption that at most one pursuer is actively chasing a moving target at every instant of time, we have proposed

[4] An animated version can be found at http://dcsl.gatech.edu/movies/MPMTalgo1.avi.

[5] An animated version can be found at http://dcsl.gatech.edu/movies/MPMTalgo2.avi.

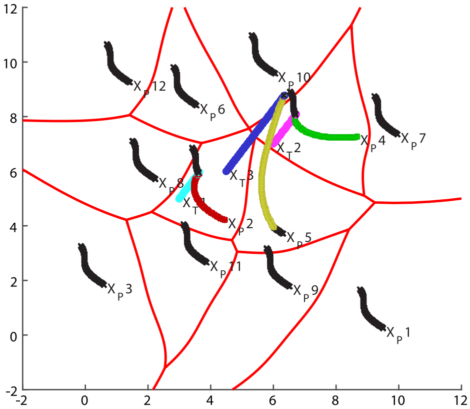


Fig. 3. Trajectories of pursuers and evaders by applying the first algorithm. Red curves in the background represents the ZVD at $t = 0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

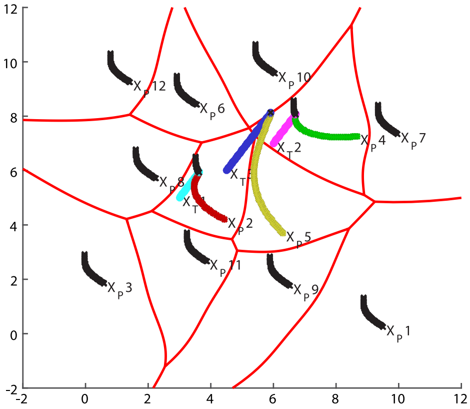


Fig. 4. Trajectories of pursuers and evaders after applying the second algorithm. Red curves in the background represents the ZVD at $t = 0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a target–pursuer assignment strategy to capture several moving targets by a set of pursuers in a wind field, when the only information about the targets known to the pursuers are their current locations at every instant of time. The targets are not affected by the wind field, resulting in asymmetric pursuer/target dynamics. We take advantage of the fact that the problem of assigning a pursuer to the moving target can be associated with a dynamically changing Zermelo–Voronoi partitioning problem. This partition assigns to each pursuer the points that can be intercepted faster than any other pursuer, using the minimum-time Zermelo's navigation law. We use the Zermelo–Voronoi diagram (ZVD) to dynamically assign the active pursuers at each instant of time.

Several extensions of this work are possible. One of the challenges is to remove the restriction that only one pursuer chases a target at every instant of time. This problem can be easily solved by constructing the ZVD having as its generators the targets (instead of the pursuers). Multiple pursuers may then be included in each target cell, in which case a natural question would be to consider cooperation among these pursuers in order to intercept the target faster. Another promising direction is to develop new ZVD update algorithms as well as consideration of more complicated flow fields, including pursuit-evasion scenarios with obstacles. For some initial results along this direction, see Sun, Tsiotras, Lolla, Subramani, and Lermusiaux (2017).

References

Bakolas, E. (2014). Decentralized spatial partitioning for multi-vehicle systems in spatiotemporal flow-field. *Automatica*, *50*(9), 2389–2396.

Bakolas, E., & Tsiotras, P. (2010). Minimum-time paths for a light aircraft in the presence of regionally-varying strong winds. In *AIAA Infotech@Aerospace*, (Atlanta, GA), AIAA Paper 2010-3380, April 20–22.

Bakolas, E., & Tsiotras, P. (2010). The Zermelo-Voronoi diagram: A dynamic partition problem. *Automatica*, *46*(12), 2059–2067.

Bakolas, E., & Tsiotras, P. (2012). Feedback navigation in an uncertain flow-field and connections with pursuit strategies. *AIAA Journal of Guidance, Control, and Dynamics*, *35*, 1268–1279. July-August.

Bakolas, E., & Tsiotras, P. (2012). Relay pursuit of a maneuvering target using dynamic Voronoi diagrams. *Automatica*, *48*, 2213–2220.

Bakolas, E., & Tsiotras, P. (2013). Optimal partitioning for spatiotemporal coverage in a drift field. *Automatica*, *49*(7), 2064–2073.

Blagodatskikh, A. I. (2008). Group pursuit in Pontryagin's nonstationary example. *Differential Equations*, *44*(1), 40–46.

Blagodatskikh, A. I. (2009). Simultaneous multiple capture in a simple pursuit problem. *Journal of Applied Mathematics and Mechanics*, *73*(1), 36–40.

Bryson, A. E., & Ho, Y.-C. (1975). *Applied optimal control: Optimization, estimation, and control*. Taylor & Francis.

De Berg, M., Cheong, O., Van Kreveld, M., & Overmars, M. (2008). *Computational geometry: Algorithms and applications*. Springer.

Devillers, O. (1999). On deletion in Delaunay triangulations. In *Proceedings of the fifteenth annual symposium on computational geometry*, (New York, USA), (pp. 181–188).

Devillers, O., & Golin, M. (1993). Dog bites postman: Point location in the moving Voronoi diagram and related problems. In *Algorithms–ESA'93*, (pp. 133–144).

Guibas, L., & Russel, D. (2004). An empirical comparison of techniques for updating Delaunay triangulations. In *Proceedings of the twentieth annual symposium on computational geometry*, (New York, USA), (pp. 170–179).

Ibragimov, G. I., Salimi, M., & Amini, M. (2012). Evasion from many pursuers in simple motion differential game with integral constraints. *European Journal of Operational Research*, *218*(2), 505–511.

Jang, J. S., & Tomlin, C. J. (2005). *Control strategies in multi-player pursuit and evasion game, vol. 6239* (pp. 15–18). AIAA.

Kao, T. (1991). Dynamic maintenance of Delaunay triangulations. *Work*, *9*, 32.

Khaidarov, B. K. (1984). Positional i-capture in the game of a single evader and several pursuers. *Journal of Applied Mathematics and Mechanics*, *48*(4), 406–409.

Krasovskiĭ, A. N., & Krasovskiĭ, N. N. (1994). *Control under lack of information*. Springer Science & Business Media.

Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. (2009). *Spatial tessellations: Concepts and applications of Voronoi diagrams, Vol. 501*. Wiley.

Pashkov, A. G., & Terekhov, S. D. (1987). A differential game of approach with two pursuers and one evader. *Journal of Optimization Theory and Applications*, *55*(2), 303–311.

Pittsyk, M., & Chikrii, A. A. (1982). On a group pursuit problem. *Journal of Applied Mathematics and Mechanics*, *46*(5), 584–589.

Pshenichnyi, B. N. (1976). Simple pursuit by several objects. *Cybernetics and Systems Analysis*, *12*(3), 484–485.

Roos, T. (1993). Voronoi diagrams over dynamic scenes. *Discrete Applied Mathematics*, *43*(3), 243–259.

Sun, W., Tsiotras, P., Lolla, T., Subramani, D., & Lermusiaux, P. (2017). Multiple-pursuer-one-evader pursuit evasion game in dynamic flow fields. *AIAA Journal of Guidance, Control, and Dynamics*,.

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.

Yeung, D. W. K., & Petrosjan, L. A. (2006). *Cooperative stochastic differential games*. Springer Science & Business Media.

Wei Sun received his B.S. degree in Mathematics from the Peking University, China, in 2010 and his M.S. degrees in both Aerospace Engineering and Mathematics from the Georgia Institute of Technology, Atlanta, in 2015. He is currently a Ph.D. candidate at the Georgia Institute of Technology. His current research interests include optimal control, differential games, reinforcement learning and trajectory optimization with applications in motion planning, and pursuit evasion games.

Panagiotis Tsiotras is the Dean's Professor in the School of Aerospace Engineering at the Georgia Institute of Technology. He received his Engineering Diploma in Mechanical Engineering from the National Technical University of Athens, in Greece (1986) and his Ph.D. degree in Aeronautics and Astronautics from Purdue (1993). His main research interests are in optimal and nonlinear control, and vehicle autonomy. In 1996 he received the NSF CAREER award. He has served in the Editorial Boards of the *Transactions on Automatic Control*, the *IEEE Control Systems Magazine*, the *Journal of Guidance, Control and Dynamics* and the journal *Dynamics and Control*. He is a Fellow of AIAA and a Senior Member of IEEE.