A Sequential Pursuer-Target Assignment Problem Under External Disturbances

Wei Sun[1] and Panagiotis Tsiotras[2]

***Abstract*— In this paper we deal with the problem of a team of pursuers distributed in the plane subject to an environmental disturbance (e.g., wind). The objective of the pursuers is to intercept a moving target which is not affected by the presence of the disturbance. We solve this 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 present an algorithm to efficiently update this Voronoi-like partition on-line. Simulations are included to illustrate the theoretical results.**

I. Introduction

Consider a scenario where a group of helicopters or small UAVs in a wind field are trying to capture a vehicle moving on the ground, or a team of small marine or underwater vehicles attempting to reach 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 in minimum time. Problems of this nature fall under the general class of group pursuit problems [1], [2]. These are difficult problems to solve, in general. Their solution is also based on the information the pursuers and the target have about each other, resulting in either cooperative or non-cooperative strategies [1]–[9]. In this work, in order to solve this problem, we propose a *sequential* pursuit strategy. By sequential (or relay) pursuit we mean that only one pursuer is assigned to chase the 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 consumption of the agents is an important factor, when the agents also play a dual role as guardians protecting a certain area, or in order to account for possible deceptive strategies of an opponent.

In contrast to most standard pursuit-evasion problem formulations [10], where the effect of the environment is not taken into consideration, in our problem setup (only) the pursuers will be affected by known exogenous environmental conditions (e.g., winds or sea currents). It will also be assumed that each pursuer has a stroboscopic view of the target position [10]. That is, each pursuer knows the current position of the target but not its future position nor its velocity. Our objective is to find which pursuer will go after the target at each instant of time so as to reduce or minimize the capture time.

[1]W. Sun is a Ph.D. candidate at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta. GA 30332-0150. USA. Email: wsun42@gatech.edu

[2]P. Tsiotras is a Professor at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta. GA 30332-0150. USA. Email: tsiotras@gatech.edu

Our strategy to solve this problem will be based on the dynamic assignment of the best pursuer to go after the target based on a Voronoi-like partition of the plane called the Zermelo-Voronoi partition, or the Zermelo-Voronoi Diagram (ZVD) [11]. Such Voronoi-like diagrams have been previously introduced in [5], [7], [11] and use time-to-intercept as the relevant distance metric. Essentially, a ZVD allows one to succinctly encode the "isocost" surfaces of the associated minimum-time to intercept problems emanating from the pursuer locations. The difficulty in our problem arises from the fact that, owing to the presence of a wind field, a point in the plane maybe close to a pursuer in terms of Euclidean distance, but 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.

Zermelo-Voronoi diagrams have been used in the past to solve group pursuit problems in the plane [5], [7], [11]. Although our work follows closely the original work in [7], where ZVDs were first employed in order to generate the pursuer assignments, and of [5], where the concept of relay pursuit was first introduced, our work differs from those works as follows: a) in this paper the wind disturbance affects only the pursuers, thus leading to asymmetric dynamics between the target and the pursuers; b) we assume minimal knowledge of the target state, namely, only its instantaneous position is known to the pursuers. On the other hand, in [5] it was assumed that the target was implementing an evading strategy that it was known to all the pursuers; c) finally, we also propose a numerically efficient algorithm to update the ZVD on-the-fly to increase the computational efficiency of the proposed assignment algorithm.

II. Preliminaries

A. Voronoi Diagrams and Delaunay Triangulation

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 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. If we use a Euclidean distance metric, this tessellation results in the *ordinary Voronoi diagram* (VD) generated by the given point set. The corresponding regions are called the *Voronoi cells* of the tessellation [12].

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 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 [12]. This is the *Delaunay property*. Given any triangulation of a given 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* [13].

B. The Zermelo-Voronoi Diagram

When we deal with pursuer-target problems, in many cases we want to know the proximity relation between a set of agents, acting as pursuers, and a target on the plane. The problem of obtaining this proximity relation 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) [11], and then finding the cell in which the target resides at the given time instant.

We state the precise definition of the ZVD below.

Definition 2.1 (Zermelo-Voronoi Diagram [11]): Given a set of n agents starting from distinct initial positions, whose dynamics are given by

$$\dot{X}_P^i = u_P^i + w(X_P^i, t), \qquad X_P^i(0) = X_{P_0}^i, \tag{1}$$

where $X_P^i := [x_P^i, y_P^i]^\mathsf{T} \in \mathbb{R}^2$ denotes the position of the i^{th} agent, $u_P^i \in \mathbb{R}^2$ is the control input of the i^{th} agent, and $w(X_P^i, t)^T \in \mathbb{R}^2$ represents the wind disturbance, the *Zermelo-Voronoi diagram* (ZVD)[1] (or Zermelo-Voronoi partition) is a set partition of the plane $\mathcal{Z} = \{Z_1, Z_2, ..., Z_n\}$ such that

$i)$ $\mathbb{R}^2 = \bigcup_{i=1}^n Z_i$,
$ii)$ For any point in Z_i, the agent i will reach this point faster than any other agent.

The sets Z_i are the *Zermelo-Voronoi cells* for the partition.

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

Proposition 2.2 ([11]): Let $\mathcal{V} = \{V_i, i \in I\}$, where $I = \{1, 2, ..., n\}$, 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 $w(X_P^i, t) = w(t)$ for all $i \in I$. Let the one-to-one, continuous function $F : \mathbb{R}^2 \to \mathbb{R}^2$ be defined by

$$F(X) = f_{P_i}(X), \quad X \in V_i, \quad i \in I, \tag{2}$$

where

$$f_{P_i}(X) = X + \int_0^{|X-P_i|} w(\tau)\,\mathrm{d}\tau, \quad i \in I. \tag{3}$$

Then $Z_i = F(V_i)$ and thus ZVD is the image of VD under the mapping F.

In other words, for this case there exists a homeomorphism between the ordinary Voronoi diagram and the Zermelo-Voronoi diagram with the same generators.

[1]Note that in [11] this is referred to as the dual Zermelo-Voronoi diagram, not to be confused with the dual graph of the Zermelo-Voronoi diagram.

III. Problem Setup

Consider a group of n pursuers in the plane, denoted by the index set $I = \{1, 2, ..., n\}$, and assume that at time $t = 0$ the pursuers are located at n distinct positions in the plane, designated by $\mathcal{P}_0 = \{X_{P_0}^i \in \mathbb{R}^2, i \in I\}$. The kinematics of the i^{th} pursuer, $i \in I$, are described by (1), where it is assumed that $u_P^i \in \mathcal{U}_P$, where 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

$$|w(t)| \leqslant \bar{w}, \tag{4}$$

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 [14].

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

$$\dot{X}_T = u_T, \qquad X_T(0) = X_{T_0}, \tag{5}$$

where $X_T = [x_T, y_T]^\mathsf{T} \in \mathbb{R}^2$ is the position of the target, and $u_T \in \mathbb{R}^2$ is its control input such that $u_T \in \mathcal{U}_T$. The set $\mathcal{U}_T$ 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.

We assume that the pursuers have accurate measurements only of the current position of the target at every instant of time. One reasonable strategy for every pursuer is therefore to use the Zermelo navigation law [5], [15] in order to intercept the target, that is, at every instant of time, the pursuer approaches the target with the control law obtained by the solution of the corresponding Zermelo navigation problem. This control law is optimal at $t = 0$ if the target remains stationary for all $t \geqslant 0$ [15]. As discussed in [11], starting at time $t = 0$, the optimal time of arrival T_{ZN}^i of the i^{th} pursuer from $X_{P_0}^i$ to X_{T_0} is given by

$$T_{\mathrm{ZN}}^i = \min\{T > 0 : \bar{u}T - |X_{T_0} - X_{P_0}^i - \int_0^T w(\tau)\,\mathrm{d}\tau| = 0\} \tag{6}$$

Then the Zermelo's navigation control can be obtained by $u_{\mathrm{ZN}}^i = \bar{u}(\cos\theta_i^*, \sin\theta_i^*)^\mathsf{T}$, where $\theta_i^* = \mathrm{Arg}\big(X_{T_0} - X_{P_0}^i - \int_0^{T_{\mathrm{ZN}}^i} w(\tau)\,\mathrm{d}\tau\big)$, for $i \in I$.

IV. Analysis and implementation of the pursuer-target assignment problem

Before proceeding with the solution of the optimal pursuer assignment problem, we 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 finite capture time.

The *robust optimal line-of-sight* navigation law (ROLS) 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 [16]. We can use this strategy to ensure capture as follows. First, let $Y^i(t) = X_T(t) - X_P^i(t)$ be the vector from the pursuer to the target. The ROLS navigation law of the i^{th} pursuer can be expressed as [16]

$$u^i_{\text{ROLS}}(t, Y^i) = \sqrt{\bar{u}^2 - \langle w(t), e^i_2(t)\rangle^2} e^i_1(t) - \langle w(t), e^i_2(t)\rangle e^i_2(t), \tag{7}$$

where $e^i_1(t) = Y^i(t)/|Y^i(t)|$ and $e^i_2(t) = Se^i_1(t)$ for all $i \in I$, where S is the matrix $\left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$.

The following result is adapted from [16].

Proposition 4.1: Let $\epsilon > 0$, and assume that the dynamics of each pursuer is given by (1) and the dynamics of the target is given by (5). Then, for each pursuer i, and for all initial conditions $X^i_{P_0}$ and X_{T_0}, there exists a finite time $T^i_{\text{ROLS}}(X^i_{P_0}, X_{T_0}) \geqslant 0$ such that the ith pursuer driven by the ROLS navigation law (7) enters the set $\{X \in \mathbb{R}^2 : |X - X_T(T^i_{\text{ROLS}})| \leqslant \epsilon\}$, provided that there exists $c > 0$ such that

$$\sqrt{\bar{u}^2 - \langle w(t), e^i_2(t)\rangle^2} + \langle w(t) - u_T(t), e^i_1(t)\rangle > c, \tag{8}$$

for all $t \geqslant 0$.

Proof: From equation (4) we have that

$$\langle u^i_{\text{ROLS}}(t, Y), e^i_1(t)\rangle = \sqrt{\bar{u}^2 - \langle w(t), e^i_2(t)\rangle^2}. \tag{9}$$

Using the control (7) in the i^{th} pursuer dynamics, and subtracting it from the target dynamics (5), we get

$$\dot{Y}^i(t) = -u^i_{\text{ROLS}}(t, Y^i) - w(t) + u_T(t). \tag{10}$$

Since the i^{th} pursuer moves along the line-of-sight, using (10) along with (8), it follows that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}|Y^i| &= \langle \dot{Y}^i, e^i_1(t)\rangle \\ &= -\langle u^i_{\text{ROLS}}(t, Y), e^i_1(t)\rangle - \langle w(t) - u_T(t), e^i_1(t)\rangle < -c.\end{aligned}$$

Thus, the ROLS navigation law will drive the ith pursuer to within an ϵ-ball of the target in finite time. ∎

The following corollary is immediate from Proposition 4.1.

Corollary 4.2: Assume that (8) holds for all $i \in I$ and all $t \geq 0$. Any sequential pursuit strategy in which each pursuer employs the ROLS navigation law (7) leads to capture of the target by a pursuer.

Since the time to capture using the ROLS control law is always larger than or equal to the time of capture using the Zermelo navigation law, Proposition 4.1 implies that the minimum-time intercept problem using Zermelo's navigation law always has a solution (if the target is stationary) for all initial conditions for the pursuers and the target. Since by applying the Zermelo's navigation law instead of (7) results in a smaller intercept time, that is, $T^i_{\text{ZN}} \leq T^i_{\text{ROLS}}$ for all $i \in I$, this, in turn, implies that a sequential strategy that uses Zermelo's navigation law for each pursuer should lead to capture. By imposing a somewhat stronger 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 (1) and the dynamics of the target is given by (5). Furthermore, assume that

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

Then, for each pursuer i, and for all initial conditions $X^i_{P_0}$ and X_{T_0}, there exists a finite time $T^i_{\text{ZN}}(X^i_{P_0}, X_{T_0}) \geqslant 0$ such that the ith pursuer driven by the Zermelo navigation law enters the set $\{X \in \mathbb{R}^2 : |X - X_T(T^i_{\text{ZN}})| \leqslant \epsilon\}$.

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 (6) that the time-to-intercept a stationary target at $X_T(t_k)$ is given by $T^i_{\text{ZN}_k} = \min\{T > 0 : \bar{u}T - |Y^i(t_k) - \int_{t_k}^{t_k+T} w(\tau)\,\mathrm{d}\tau| = 0\}$. In particular,

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

and the corresponding optimal control law at t_k for the ith pursuer is given by

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

At the next time step, $t = t_k + \delta t$, we can easily compute that

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

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,

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

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

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

At this point, pick $\delta t = \epsilon/(\bar{u} + \bar{w})$, and assume that there exist $k > 0$ such that $T^i_{\text{ZN}_k} \leq \delta t$. Then from (13) one obtains

$$\begin{aligned}|Y_i(t_k)| &\leq |u^i_{\text{ZN}}(t_k)T^i_{\text{ZN}_k} + \int_{t_k}^{t_k+T^i_{\text{ZN}_k}} w(\tau)\,\mathrm{d}\tau| \\ &\leq \bar{u}T^i_{\text{ZN}_k} + \bar{w}T^i_{\text{ZN}_k} = (\bar{u} + \bar{w})T^i_{\text{ZN}_k} \\ &\leq (\bar{u} + \bar{w})\delta t \leq (\bar{u} + \bar{w})\frac{\epsilon}{\bar{u} + \bar{w}} = \epsilon.\end{aligned}$$

This shows that the pursuer is in the ϵ ball centered at the evader's position and capture has occurred.

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

$$Y^i(t_k+\delta t) - \int_{t_k+\delta t}^{t_k+\delta t+T^i_{\mathrm{ZN}_{k+1}}} w(\tau)\,\mathrm{d}\tau = \left(1-\frac{\delta t}{T^i_{\mathrm{ZN}_k}}\right)\Big(Y^i(t_k) - \int_{t_k}^{t_k+T^i_{\mathrm{ZN}_k}} w(\tau)\,\mathrm{d}\tau\Big) - \int_{t_k+T^i_{\mathrm{ZN}_k}}^{t_k+\delta t+T^i_{\mathrm{ZN}_{k+1}}} w(\tau)\,\mathrm{d}\tau + u_T^k\delta t.$$

Let $\Theta_k = Y^i(t_k) - \int_{t_k}^{t_k+T^i_{\mathrm{ZN}_k}} w(\tau)\,\mathrm{d}\tau$, and $H_k = -\int_{t_k+T^i_{\mathrm{ZN}_k}}^{t_k+\delta t+T^i_{\mathrm{ZN}_{k+1}}} w(\tau)\,\mathrm{d}\tau + u_T^k\delta t$. Then (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 as $|\Theta_k| = \bar{u}T^i_{\mathrm{ZN}_k}$. Subtracting the last two expressions yields

$$-\bar{u}\Delta T_k = |\Theta_k| - |\alpha\Theta_k + H_k|, \tag{16}$$

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, (16) 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 $(1-\alpha)|\Theta_k| = \delta t/T^i_{\mathrm{ZN}_k}|\Theta_k| = \bar{u}\delta t$. We also have $|H_k| \leq |u_T^k|\delta t + \left|\int_{t_k+T^i_{\mathrm{ZN}_k}}^{t_k+\delta t+T^i_{\mathrm{ZN}_{k+1}}} w(\tau)\,\mathrm{d}\tau\right| \leq \bar{q}\delta t + \bar{w}|\Delta T_k + \delta t|$. Thus, we get

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

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 $|H_k| \leq \bar{q}\delta t + \bar{w}(\Delta T_k + \delta t)$ and hence $-\bar{u}\Delta T_k \geq \bar{u}\delta t - \bar{q}\delta t - \bar{w}(\Delta T_k + \delta t)$ or that

$$(\bar{w} - \bar{u})\Delta T_k \geq (\bar{u} - \bar{q} - \bar{w})\delta t, \tag{18}$$

which leads to a contradiction since the left-hand side of inequality (18) is non-positive and the right-hand side is positive. It follows that $T^i_{\mathrm{ZN}_{k+1}} - T^i_{\mathrm{ZN}_k} = \Delta T_k < 0$ for all $k > 0$. This 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 $k\to\infty$ of (17) yields $\bar{u}\delta t \leq (\bar{q}+\bar{w})\delta t$ or that $\bar{u} \leq \bar{q}+\bar{w}$, contradicting (11). ■

The next corollary follows immediately from the previous proposition.

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

We may now propose the following algorithm to assign the active pursuers:

Dynamic Assignment of Active Pursuer

a) Construct the ZVD and assign the i^{th} pursuer to be the active pursuer if the target resides in the corresponding Zermelo-Voronoi cell Z_i.
b) At every time step, generate the ZVD and assign the j^{th} pursuer to be the active pursuer if the target resides in the corresponding Zermelo-Voronoi cell Z_j.
c) Check the distance between the target and the active pursuer and repeat step b) if the distance is bigger than ϵ. Otherwise, terminate the procedure and return the sequence of active pursuers.

V. Update Algorithm to Dynamically Generate the ZVD

Hereby, we present an algorithm that generates the ZVD by updating the ZVD from one time step to the next when a single pursuer has moved[2].

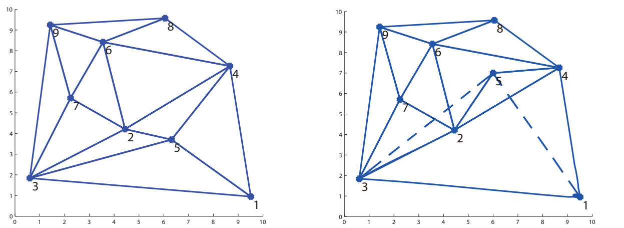
From (2), we know that there exists an invertible, continuous transformation between the ordinary VD and the ZVD. Thus, our strategy for updating the ZVD is to update the ordinary VD corresponding to the same generators first, and then form the ZVD through this transformation. In order to update the ordinary VD we will, instead, update its dual graph, namely, its Delaunay Triangulation. There exist several algorithms in the literature for updating a DT [17]–[20]. We will use a modification of the algorithm introduced in [18] 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 of the queue, we remove this point from the original triangulation and then insert it back at the new location at the present time [21]. 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. As shown in [22] the complexity of generating a new DT by removing a point is of complexity $O(k\log k)$, where k is the degree (number of neighbors) of the removed point.

Given the previous considerations, we propose an alternative approach to deal with moving generators. 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 Triangulations at time steps t_k and t_{k+1} respectively. Assume that the corresponding generator sets are given by $\mathcal{P}_k$ and $\mathcal{P}_{k+1}$. Our goal is to update DT_k to DT_{k+1} with as few deletions as possible. To this end, we want to check first if 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 [18]. 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*. Figure 1(a) shows a DT associated with a given point set, and Figure 1(b) shows the DT associated with a new point set, where point 5 has changed its location. Some triangles overlap with each other in Figure 1(b). Thus, the DT in Figure 1(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

[2]This algorithm can also be applied to the case where more than one generator is moving.

with vertices 3, 4, and 5 in Figure 1(a) has a clockwise orientation. In Figure 1(b), on the other hand, the orientation of the triangle with the same vertices has counter-clockwise orientation, i.e., its orientation has changed.



(a) Delaunay Triangulation of nine generators.

(b) Generator no. 5 changed its location and caused an unembedding.

Fig. 1. Unembedding caused by relocation of a generator.

We introduce the *orientation certificate* to check whether DT_k is an embedding or not [18], [23]. 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 algorithm for updating the Zermelo-Voronoi diagram from the previous time step to current time step is given in Algorithm 1.

To remove a point from the standard Delaunay Triangulation, we use the deletion method introduced in [22] and to insert a point into the DT, we choose the algorithm given in [21]. Both these algorithms have complexity $O(k \log k)$, where k is the number of neighbors of the removed point. The flip-edge algorithm introduced in [24] has worst case complexity $O(n^2)$, but in practice it is much faster.

Algorithm 1 Update Zermelo Voronoi Diagram

Input: Coordinates $\mathcal{P}_{k-1}$ of the generators at the previous time step and the corresponding Delaunay triangulation DT, coordinates $\mathcal{P}_k$ 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 $\mathcal{P}_k$ **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 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 (2).
  return ZVD and DT.
end procedure

VI. Simulation Results

In this section, we consider a scenario where the target is moving in a straight line according to equation (5), where $u_T(t) = [-0.4, -0.5]^\mathsf{T}$. Assume that there exist 12 pursuers, each having maximum unit speed ($\bar{u} = 1$), which are initially located at distinct positions determined by $\mathcal{P}_0$. The wind field that affects the pursuers is given by

$$w(t) = \begin{bmatrix} -0.2 - 0.2\cos(t) \\ 0.3 \end{bmatrix}.$$

Figures 2-4 illustrate the trajectories of the pursuers in the wind and the moving target. Specifically, Figure 2 shows the ZVD formed by the pursuers at $t = 0$. As seen in this figure, $i = 4$ is the active pursuer since the target falls in the Zermelo-Voronoi cell of X_P^4. Figure 3 illustrates the trajectories of the target and the pursuers in the time interval $[0, \tau_1]$, where $\tau_1 = 2.6$ is the switching time. The Zermelo-Voronoi Diagram at $t = \tau_1$ is also presented to show that the target is about to leave the Zermelo-Voronoi cell of X_P^4 and enter another cell. Figure 4 shows the trajectories of the target and the pursuers from $t = \tau_1$ to capture time $T_c = 5.0$, as well as the Zermelo-Voronoi Diagram at $t = T_c$. In the last time interval the target is assigned to $i = 5$.

For comparison, note that when only one pursuer tries to capture the target, the shortest possible time is $T_c = 7.5$. In that case there is a single active pursuer, namely, X_P^4.

VII. Conclusions

Under the assumption that only one pursuer is actively chasing a moving target at every instant of time, we have proposed a target-pursuer assignment strategy to capture a moving target by a set of pursuers in a wind field, when the only information about the target known to the pursuers is the current target location at every instant of time. The target is 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, by utilizing the minimum-time Zermelo's navigation law. We use the Zermelo-Voronoi diagram (ZVD) to dynamically assign the active pursuer at each instant of time.

Several extensions of this work are possible. An obvious one is to consider a maneuvering target whose strategy is given in a feedback form, and investigate its impact on the optimal assignment strategy. We could also assume that each

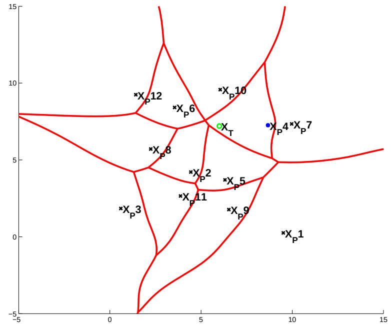


Fig. 2. Zermelo-Voronoi Diagram formed by pursuers at $t = 0$, X_P^4 is the active pursuer

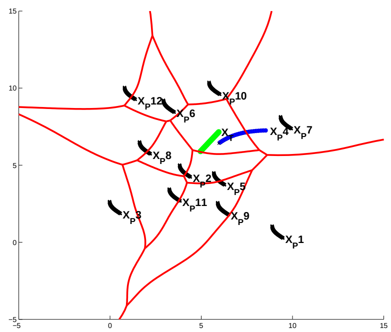


Fig. 3. Zermelo-Voronoi Diagram formed by pursuers at the first switch time $t = 2.6$, and trajectories of pursuers and target for $t \in [0, 2.6)$.

agent (target or pursuer) obeys a turning constraint, leading to the solution of an input constrained Zermelo minimum-time problem for each pursuer/tagret pair. This can be easily done using standard techniques from optimal control theory. More challenging would be to remove the restriction that only one pursuer chases the target at every instant of time and consider problems where cooperation among pursuers is possible (or even necessary) in order to intercept the target. Finally, the case of multiple targets is also a problem that naturally fits the ZVD partitioning framework since the complexity of the ZVD construction is independent of the number of targets.

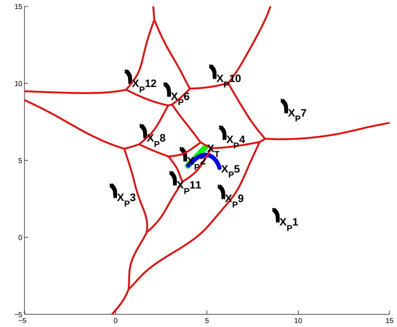


Fig. 4. Zermelo-Voronoi Diagram formed by pursuers at the capture time $T_c = 5.0$, and trajectories of pursuers and target for $t \in [2.6, 5.0]$, X_P^5 is the active pursuer.

References

[1] M. Pittsyk and A. Chikrii, “On a group pursuit problem,” *Journal of Applied Mathematics and Mechanics*, vol. 46, no. 5, pp. 584–589, 1982.

[2] A. Blagodatskikh, “Simultaneous multiple capture in a simple pursuit problem,” *Journal of Applied Mathematics and Mechanics*, vol. 73, no. 1, pp. 36–40, 2009.

[3] O. Devillers and M. Golin, “Dog bites postman: Point location in the moving Voronoi diagram and related problems,” *Algorithms–ESA’93*, pp. 133–144, 1993.

[4] O. Devillers, M. Golin, K. Kedem, and S. Schirra, “Queries on Voronoi diagrams of moving points,” *Computational Geometry*, vol. 6, no. 5, pp. 315–327, 1996.

[5] E. Bakolas and P. Tsiotras, “Relay pursuit of a maneuvering target using dynamic Voronoi diagrams,” *Automatica*, vol. 48, pp. 2213–2220, Aug. 2012.

[6] A. Blagodatskikh, “Group pursuit in Pontryagin’s nonstationary example,” *Differential Equations*, vol. 44, no. 1, pp. 40–46, 2008.

[7] E. Bakolas and P. Tsiotras, “Optimal pursuer and moving target assignment using dynamic Voronoi diagrams,” in *American Control Conference*, (San Francisco, California, USA), pp. 5444–5449, 2011.

[8] J. P. Hespanha, H. J. Kim, and S. Sastry, “Multiple-agent probabilistic pursuit-evasion games,” in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 3, pp. 2432–2437, IEEE, 1999.

[9] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, “Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation,” *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 662–669, 2002.

[10] O. Hajek, *Pursuit Games: An Introduction to the Theory and Applications of Differential Games of Pursuit and Evasion*. Mineola, New York: Dover Publications, second ed., 2008. Chap. 1, pp. 1–7.

[11] E. Bakolas and P. Tsiotras, “The Zermelo-Voronoi diagram: A dynamic partition problem,” *Automatica*, vol. 46, no. 12, pp. 2059–2067, 2010.

[12] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, vol. 501. Wiley, 2009.

[13] T. Roos, “Voronoi diagrams over dynamic scenes,” *Discrete Applied Mathematics*, vol. 43, no. 3, pp. 243–259, 1993.

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

[15] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*. Taylor & Francis, 1975.

[16] E. Bakolas and P. Tsiotras, “Feedback navigation in an uncertain flow-field and connections with pursuit strategies,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 35, pp. 1268–1279, July-August 2012.

[17] I. Lee and M. Gahegan, “Interactive analysis using voronoi diagrams: Algorithms to support dynamic update from a generic triangle-based data structure,” *Transactions in GIS ((this is the name of the journal, GIS stands for geographic information system))*, vol. 6, no. 2, pp. 89–114, 2002.

[18] L. Guibas and D. Russel, “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, Association for Computing Machinery, 2004.

[19] P. M. Manhães de Castro, J. Tournois, P. Alliez, and O. Devillers, “Filtering relocations on a Delaunay triangulation,” in *Computer Graphics Forum*, vol. 28, pp. 1465–1474, 2009.

[20] O. Devillers, “The Delaunay hierarchy,” *International Journal of Foundations of Computer Science*, vol. 13, no. 2, pp. 163–180, 2002.

[21] T. Kao, “Dynamic maintenance of Delaunay triangulations,” *Work*, vol. 9, p. 32, 1991.

[22] O. Devillers, “On deletion in Delaunay triangulations,” in *Proceedings of the Fifteenth Annual Symposium on Computational Geometry*, (New York, USA), pp. 181–188, Association for Computing Machinery, 1999.

[23] L. J. Guibas, J. S. B. Mitchell, and T. Roos, “Voronoi diagrams of moving points in the plane,” in *Graph-Theoretic Concepts in Computer Science*, pp. 113–125, Springer, 1992.

[24] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer, 2008.