2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Active Target Tracking with Self-Triggered Communications

Lifeng Zhou and Pratap Tokekar

Abstract— **We study the problem of reducing the amount of communication in a distributed target tracking problem. We focus on the scenario where a team of robots are allowed to move on the boundary of the environment. Their goal is to seek a formation so as to best track a target moving in the interior of the environment. The robots are capable of measuring distances to the target. Decentralized control strategies have been proposed in the past that guarantee that the robots asymptotically converge to the optimal formation. However, existing methods require that the robots exchange information with their neighbors at all time steps. Instead, we focus on reducing the amount of communication among robots.**

We propose a self-triggered communication strategy that decides when a particular robot should seek up-to-date information from its neighbors and when it is safe to operate with possibly outdated information from the neighbor. We prove that this strategy converges to an optimal formation. We compare the two approaches (constant communication and self-triggered communication) through simulations of tracking stationary and mobile targets.

I. Introduction

The target tracking problem has been one of the most well-studied problems in the robotics community [1] and finds many applications such as surveillance [2]–[4], crowd monitoring [5], [6], and wildlife monitoring [7], [8]. In this paper, we study *active* target tracking with a team of robots where the focus is on actively controlling the state of the robot. Robots can exchange information with each other and decide how to move, so as to best track the target. It is typically assumed that exchanging information is beneficial. It is typically assumed that robots will exchange information at every timestep irrespective of whether that information is worth exchanging. We investigate the problem of deciding when is it worthwhile for the robots to exchange information and when is it okay to use possibly outdated information.

The motivation for our work stems from the observation that communication can be costly. For example, for smaller robots radio communication can be a significant source of power consumption. Robots can extend their lifetimes by reducing the time spent communicating. Our objective is to determine a strategy that communicates only when required without significantly affecting the tracking performance.

We study this problem in a simple target tracking scenario first introduced by Martinez and Bullo [9]. Here, the robots are restricted to move on the boundary of a convex environment. They can obtain distance measurements towards a target moving in the interior. The goal of the robots is to position themselves so as to maximize the information gained. The authors proposed a decentralized strategy where the robots communicate at all time steps with their neighbors and showed that it converges to the optimal (uniform) configuration. Instead, we apply a self-triggered coordination algorithm (following recent works [10], [11]) where each robot decides when to trigger communications with its neighbors. We apply this strategy to the aforementioned target tracking scenario and compare its performance with the baseline algorithm.

Simulation results validate the theoretical analysis showing that the self-triggered strategy converges to the optimal, uniform configuration. In some cases, the self-triggered strategy converges faster than the constant communication strategy. We investigate possible causes. In particular, we observe that the robots travel a longer distance in the wrong direction in the constant strategy as opposed to the self-triggered strategy when there are many robots.

The rest of the paper is organized as follows. We start by formalizing the problem in Section II. The self-triggered tracking strategy is presented in Section III, assuming that the target's position is known and is fixed. We relax these assumptions and present a practical extension in Section IV. The simulation results are presented in Section V. We conclude with a discussion of future work in Section VI.

II. Problem Formulation

Consider a group of N robots moving on the boundary of a convex polygon $\mathcal{Q} \subset \mathbb{R}^2$. Let $\partial\mathcal{Q}$ denote the boundary of $\mathcal{Q}$. The robots are tasked with tracking a target o located in the interior of $\mathcal{Q}$. Let $p_1, ..., p_N$ denote the positions of the robots. We can map any point on $\partial\mathcal{Q}$ to a unit circle $\mathbb{T}$ using the transformation $\varphi_o : \partial\mathcal{Q} \to \mathbb{T}$ given by

$$\varphi_o(p) = \frac{p - o}{\|p - o\|} \tag{1}$$

as shown in Figure 1. We identify every robot's position with the corresponding point on the unit circle. That is, $p_i \in \partial\mathcal{Q} \subset \mathbb{R}^2$ is identified with $\theta_i = \varphi_o(p_i) \in \mathbb{T}$, indicating the location on the circle $\mathbb{T}$ of robot i. Let $\theta = (\theta_1, ..., \theta_N) \in \mathbb{T}^N$ denote the vector of locations of all robots.

We assume that all robots follow simple first-order continuous-time motion model. Let $\omega_{\max}$ denote the common maximum angular speed[1] for all robots on the unit circle. Our results can be extended to the situation where each robot has its own maximum angular speed. Each robot i knows its own

The authors are with the Department of Electrical and Computer Engineering, Virginia Tech USA. {lfzhou, tokekar}@vt.edu.

This material is based upon work supported by the National Science Foundation under Grant Nos. 1566247 and 1637915.

[1]Strictly speaking, each robot has a maximum speed with which it can move on $\partial\mathcal{Q}$. In the appendix, we show how the maximum speed on $\partial\mathcal{Q}$ can be used to determine $\omega_{\max}$.

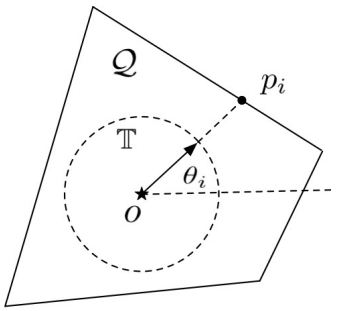


Fig. 1. The mapping from convex boundary $\partial\mathcal{Q}$ to unit circle $\mathbb{T}$.

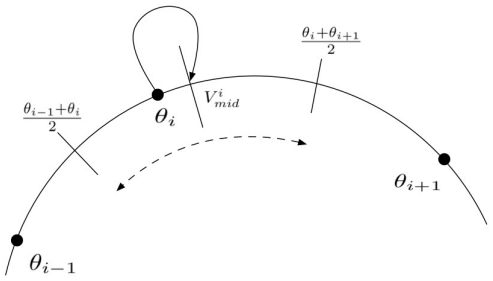


Fig. 2. Robot i goes towards the midpoint of its Voronoi segment via exact information from its neighbors.

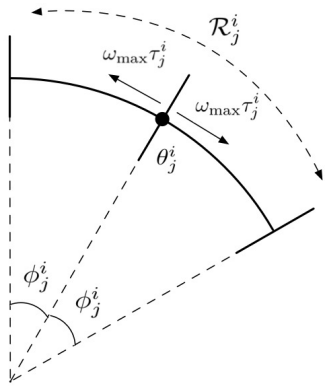


Fig. 3. Motion prediction set, $\mathcal{R}^i_j$, that each robot i maintains for its neighbors j. θ^i_j is the last known position of robot j and τ^i_j is the time elapsed since this last known position.

position exactly at all times. When two robots communicate they can exchange their respective positions.

Martinez and Bullo [9] showed that the optimal configuration for the robots that can obtain distance measurements towards the target is a uniform one, where all robots are equally spaced around the target. That is, $\theta_{i+1} - \theta_i = 2\pi/N, \forall i \in \{1, ..., N\}$. Optimality is defined with respect to maximizing the determinant of the Fisher Information Matrix (FIM). FIM is a commonly used measure for active target tracking (*e.g.*, [12]). Martinez and Bullo [9] presented a decentralized control law that is guaranteed to (asymptotically) converge to a uniform configuration when a robot is allowed to communicate with two of its immediate neighbors. That is, a robot i can communicate with only $i-1$ and $i+1$. The analysis requires that all robots know the position of the target o exactly and that the target remains stationary.

The control law in [9] assumes that neighboring robots communicate at every time step. We call this as the *constant communication strategy*. Our objective in this work is to reduce the number of communications between the robots while preserving the convergence properties. We present a *self-triggered strategy* where the control law for each robot not only decides how a robot should move, but also when should it communicate with its neighbors and seek new information. We show that the proposed self-triggered strategy is also guaranteed to converge to a uniform configuration, under the assumptions described next.

III. Self-Triggered Tracking Algorithm

In this section, we present the self-triggered tracking algorithm for achieving a uniform distribution along the unit circle. This requires knowing the center of the circle (*i.e.*, the target's position) and assuming that this center does not change. These assumptions are required for the convergence analysis to hold. We later relax these assumptions and present a practical version in the following section.

Our algorithm builds on the `self-triggered centroid algorithm` [10] which is a decentralized control law that achieves optimal deployment (*i.e.*, uniform Voronoi partitions) in a convex environment. We suitably modify this algorithm for the case where the robots are restricted to move only on the boundary, $\partial\mathcal{Q}$, and can communicate with only two neighbors as described in the previous section. We first present the control law and then an update policy to decide when a robot should seek new information from its neighbors.

A. Control Law

The constant control law in reference [9] drives every robot towards the midpoint of its Voronoi segment. Voronoi segment of the robot i is the part of the unit circle extending from $\frac{(\theta_{i-1}+\theta_i)}{2}$ to $\frac{(\theta_i+\theta_{i+1})}{2}$. The constant control law steers robot i towards the midpoint of its Voronoi segment V^i_{mid} by using real-time, up-to-date information from its neighbors, θ_{i-1} and θ_{i+1} (Figure 2). We refer to the book [13] for a comprehensive treatment on Voronoi segments.

In a self-triggered strategy, exact positions of the neighbor will not be always available in real-time. Consequently, the algorithm must be able to operate on inexact information. The information that each robot i holds about its neighbor j is the last known position of j, denoted by θ^i_j, and the time elapsed since the position of robot j was collected, denoted by τ^i_j. Based on this, robot i can compute the furthest distance that j could have moved in τ^i_j time as $\pm\phi^i_j$ where,

$$\phi^i_j = \omega_{\max}\tau^i_j. \tag{2}$$

Thus, robot i can build a prediction motion set $\mathcal{R}^i_j(\theta^i_j, \phi^i_j)$ that contains all the possible locations where robot j could have moved to in τ^i_j time (Figure 3).

Since robot i only communicates with its neighbors $i-1$ and $i+1$, the prediction motion range that robot i stores is given as $\mathcal{R}^i := \{\mathcal{R}^i_{i-1}(\theta^i_{i-1}, \phi^i_{i-1}), \mathcal{R}^i_{i+1}(\theta^i_{i+1}, \phi^i_{i+1})\}$. The

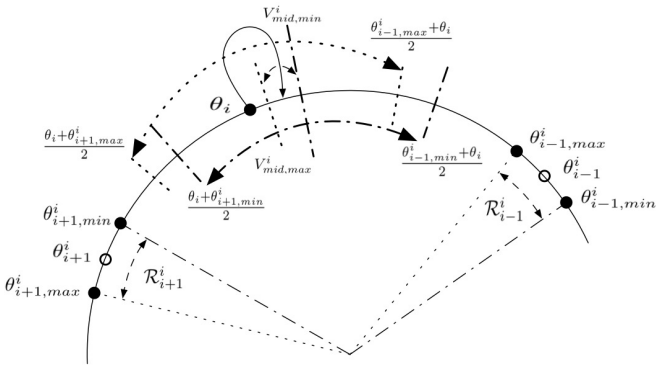


Fig. 4. Robot i goes towards the midpoint of its Voronoi segment via inexact motion prediction about its neighbors.

proposed self-triggered strategy uses these motion prediction ranges $\mathcal{R}^i$ for defining the control law of robot i.

Since the robot has inexact information of its neighbors, the midpoint of its Voronoi segment is a set instead of a point (Figure 4). We define,

$$\theta^i_{i-1,\min} = \left(\theta^i_{i-1} - \phi^i_{i-1}\right) \quad \theta^i_{i-1,\max} = \left(\theta^i_{i-1} + \phi^i_{i-1}\right)$$
$$\theta^i_{i+1,\min} = \left(\theta^i_{i+1} - \phi^i_{i+1}\right) \quad \theta^i_{i+1,\max} = \left(\theta^i_{i+1} + \phi^i_{i+1}\right).$$

Thus, we have:

$$\mathcal{R}^i_{i-1}(\theta^i_{i-1}, \phi^i_{i-1}) = \{\beta \in \mathbb{T} | \theta^i_{i-1,\min} \leq \beta \leq \theta^i_{i-1,\max}\},$$
$$\mathcal{R}^i_{i+1}(\theta^i_{i+1}, \phi^i_{i+1}) = \{\beta \in \mathbb{T} | \theta^i_{i+1,\min} \leq \beta \leq \theta^i_{i+1,\max}\}.$$

The minimum and maximum midpoints of robot i's Voronoi segment can be computed as,

$$V^i_{\text{mid},\min} = \frac{(\theta^i_{i-1,\min} + \theta_i)/2 + (\theta_i + \theta^i_{i+1,\min})/2}{2}, \quad (3)$$

$$V^i_{\text{mid},\max} = \frac{(\theta^i_{i-1,\max} + \theta_i)/2 + (\theta_i + \theta^i_{i+1,\max})/2}{2}. \quad (4)$$

For the midpoint of its Voronoi segment V^i_{mid}, we have:

$$V^i_{\text{mid},\min} \leq V^i_{\text{mid}} \leq V^i_{\text{mid},\max}. \quad (5)$$

Substitute Equations 3 and 4 into Equation 5 yields,

$$\frac{\theta^i_{i+1} + 2\theta_i + \theta^i_{i-1} - 2\omega_{\max}\tau^i}{4} \leq V^i_{\text{mid}}$$

with $\tau^i = \tau^i_{i-1} = \tau^i_{i+1}$ and

$$V^i_{\text{mid}} \leq \frac{\theta^i_{i+1} + 2\theta_i + \theta^i_{i-1} + 2\omega_{\max}\tau^i}{4}.$$

Therefore,

$$-\frac{\omega_{\max}\tau^i}{2} \leq V^i_{\text{mid}} - \frac{\theta^i_{i+1} + 2\theta_i + \theta^i_{i-1}}{4} \leq \frac{\omega_{\max}\tau^i}{2},$$

which yields,

$$\left| V^i_{\text{mid}} - \frac{\theta^i_{i+1} + 2\theta_i + \theta^i_{i-1}}{4} \right| \leq \frac{\omega_{\max}\tau^i}{2}. \quad (6)$$

Thus, the angular distance between V^i_{mid} and $\frac{\theta^i_{i+1}+2\theta_i+\theta^i_{i-1}}{4}$ is bounded by $\frac{\omega_{\max}\tau^i}{2}$.

In fact, the point $\frac{\theta^i_{i+1}+2\theta_i+\theta^i_{i-1}}{4}$ indicates the midpoint of i's guaranteed Voronoi segment gVs_i, defined as,

$$gVs_i = \left\{ \beta \in \mathbb{T} \left| \max_{\theta_i \in T_i} |\beta - \theta_i| \leq \min_{\theta_j \in T_j} |\beta - \theta_j|, \forall j \neq i \right. \right\}$$

where $T_1, \ldots, T_n \subset \mathbb{T}$ are a set of connected segments in $\mathbb{T}$. We refer to the report [14] for more details on the guaranteed Voronoi segment. Thus, the guaranteed Voronoi segment of robot i can be computed as,

$$gVs_i = \left\{ \beta \left| \frac{\theta_i + \theta^i_{i+1,\min}}{2} \leq \beta \leq \frac{\theta^i_{i-1,\max} + \theta_i}{2} \right. \right\}. \quad (7)$$

Robot i does not know the exact midpoint of its Voronoi segment V^i_{mid}. Instead, it can move towards the midpoint of its guaranteed Voronoi segment, gV^i_{mid}, given by:

$$\begin{aligned} gV^i_{\text{mid}} &= \frac{(\theta_i + \theta^i_{i+1,\min})/2 + (\theta^i_{i-1,\max} + \theta_i)/2}{2}, \\ &= \frac{\theta^i_{i+1} + 2\theta_i + \theta^i_{i-1}}{4}. \end{aligned} \quad (8)$$

In general, moving towards gV^i_{mid} does not guarantee that the robot moves closer to the midpoint of its Voronoi segment. However, the statement holds under the condition described next.

Lemma 1: Suppose robot i moves from θ_i towards gV^i_{mid}. Let θ'_i and V'_{mid} be its position and the midpoint of its Voronoi segment, respectively, after one time step. If $|\theta'_i - gV^i_{\text{mid}}| \geq |V^i_{\text{mid}} - gV^i_{\text{mid}}|$, then $|\theta'_i - V'_{\text{mid}}| \leq |\theta_i - V^i_{\text{mid}}|$.

The proof for this lemma follows directly from the proof for Lemma 5.1 in reference [10]. Consequently, as long as the robot can ensure that its new position θ'_i satisfies $|\theta'_i - gV^i_{\text{mid}}| \geq |V^i_{\text{mid}} - gV^i_{\text{mid}}|$, then it is assured to not increase its distance from the actual (unknown) midpoint of the Voronoi segment. However, the right hand side of this condition also is not known exactly since robot i does not know V^i_{mid}. Instead, we can upper bound this term using Equation 6. We denote this upper bound by $\text{ubd}_i := \frac{\omega_{\max}\tau^i}{2}$.

Corollary 1: Suppose robot i moves from θ_i towards gV^i_{mid}. Let θ'_i be its position after one time step. If

$$|\theta'_i - gV^i_{\text{mid}}| > \text{ubd}_i, \quad (9)$$

then $|\theta'_i - V'_{\text{mid}}| \leq |\theta_i - V^i_{\text{mid}}|$.

Next, we present a motion control law that steers the robots towards a uniform configuration on the circle. Intuitively, robot i computes its guaranteed Voronoi segment (7) using the last known positions of its neighbors and the real-time position of itself. It then computes the midpoint of its guaranteed Voronoi segment (Equation 8) and moves towards the midpoint until it is within distance ubd_i of it.

The control, $u_i(t_k)$, for robot i at time t_k is given by:

$$u_i(t_k) = \omega_i \ \text{unit}(gV^i_{\text{mid}} - \theta_i), \quad (10)$$

where, $\text{unit}(x) = \frac{x}{|x|}$ and

$$\omega_i = \begin{cases} \omega_{\max}, & \left|gV^i_{\text{mid}} - \theta_i\right| \geq \text{ubd}_i + \omega_{\max}\Delta t, \\ 0, & \left|gV^i_{\text{mid}} - \theta_i\right| \leq \text{ubd}_i, \\ \frac{\left|gV^i_{\text{mid}} - \theta_i\right| - \text{ubd}_i}{\Delta t}, & \text{otherwise.} \end{cases}$$

B. Triggering Policy

As time elapses, without new information the upper bound ubd_i grows larger until the condition in Equation 9 is not met. This triggers the robot to collect the updated information from its neighbors. There are two causes that may lead to the condition in Equation 9 being violated. The upper bound on the right hand side, ubd_i, might grow large because of the time elapsed since the last communication occurred. Or, robot i might move close to gV^i_{mid} which would require ubd_i to become small by acquiring new information. The second scenario might lead to frequent triggering when the robots are close to convergence. We introduce a user-defined tolerance parameter, $\sigma \geq 0$, to relax the triggering condition. Whenever the following condition is violated, the robot is required to trigger new communication:

$$\mathrm{ubd}_i < \max\{\|\theta' - gV^i_{\text{mid}}\|, \sigma\} \tag{11}$$

The motion control law is designed under the assumption that the robot i and its two neighbors are located in the counterclockwise order (*i.e.*, $\theta_{i+1} > \theta_i > \theta_{i-1}$). Since the robots are identical, there is no advantage gained by changing the order of robots along the boundary. In a self-triggered strategy, we only have a motion prediction set of the neighbors. If there is a possibility that this order may be violated, the robots must communicate and avoid it. We achieve this by requiring the robot to maintain the following:

$$\theta^i_{i+1} - \omega_{\max}\tau^i_{i+1} > \theta_i > \theta^i_{i-1} + \omega_{\max}\tau^i_{i-1} \tag{12}$$

This ensures that even in the worst case, the robots have not swapped their positions. Whenever there is a possibility of this condition being violated, the robot i triggers a new communication.

The complete self-triggered algorithm is given below.

Algorithm 1: SELF-TRIGGERED MIDPOINT

while all robots have not converged:
  for each robot $i \in \{1, ..., N\}$ perform:
    increment τ^i_{i-1} and τ^i_{i+1} by Δt
    compute $\mathcal{R}^i, gVs_i, gV^i_{\text{mid}}$, and ubd_i
    if Equation 11 OR Equation 12 is violated:
      trigger communication with $i+1$ and $i-1$
      reset τ^i_{i+1} and τ^i_{i-1} to zero
      recompute $\mathcal{R}^i, gVs_i, gV^i_{\text{mid}}$, and ubd_i
    end if
    compute and apply u_i as defined in Equation 10
  end for
end while

C. Convergence Analysis

Algorithm 1 is guaranteed to converge asymptotically to a uniform configuration along the circumference of the circle, irrespective of the initial configuration, assuming that no two robots are co-located initially. The proof for the algorithm follows almost verbatim from the proof of Proposition 6.1 in [10] with suitable modifications. We sketch the modifications required here.

In [10] the robots are allowed to move anywhere in the interior of $Q \subset \mathbb{R}^2$ whereas in our case the robots are restricted to move on ∂Q, equivalent to moving on the unit circle $\mathbb{T}$. Therefore, all the L_2 distances in the proof in [10] change to L_1 distances. Instead of moving to the midpoint of the guaranteed Voronoi segment, the robots in [10] move to the centroid of a guaranteed Voronoi region. Instead of communicating with the two clockwise and counter-clockwise neighbors, the robots in [10] communicate with all possible Voronoi neighbors. None of these changes affect the correctness of the proof. We add an extra condition that triggers communication to prevent robots from changing their order along $\mathbb{T}$. Since this condition only results in extra triggers, it can only help convergence. Finally, since there is a one-to-one and onto mapping between ∂Q and $\mathbb{T}$, convergence along $\mathbb{T}$ implies convergence along ∂Q.

IV. PRACTICAL EXTENSION: TRACKING OF MOVING TARGET WITH NOISY MEASUREMENTS

If the true position of the target, o^*, is known, then we can draw a unit circle centered at the target and use the strategy in Algorithm 1 to converge to a uniform distribution along the circle. According to the result in [9] this configuration maximizes the determinant of the FIM. In practice, however, we do not know the true position of the target. In fact, the goal is to use the noisy measurements from the robots to estimate the position of the target. Furthermore, the target may be mobile. This implies that the (unknown) center of the circle is also moving, further complicating the control strategy for the robots.

We use an Extended Kalman Filter (EKF) that estimates the position of the target (*i.e.*, center) and predicts its motion at every time step. The prediction and the estimate of the target from an EKF is a 2D Gaussian distribution parameterized by its mean, $\hat{o}_k$, and covariance. At each time step, we use the mean of the latest estimate as the center of the circle to compute the θ_i values using the transformation in Equation 1. We evaluate two regimes for the EKF. (1) *Centralized EKF:* A common fusion center obtains the measurements from all the robots and computes a single target estimate, $\hat{o}_k$, at every time step; and (2) *Decentralized EKF:* Each robot runs its own EKF estimator and has its own target estimate, $\hat{o}^i_k$, based on only its own measurements of the target. If at any time step, a robot communicates with its neighbors, then it also shares its current measurement with its neighbors. At these triggered instances, the robot updates its own estimate using its own measurement and measurements from its neighbors.

The rest of the process is similar to that in Algorithm 1. The centralized EKF scheme is a baseline which we compare against for the more realistic, decentralized strategy. The results are presented in the simulation section that follows.

V. SIMULATION RESULTS

We evaluate the proposed algorithm by comparing the time required to achieve a uniform configuration for the self-

triggered strategy and the constant communication strategy. We also compare the two algorithms for tracking moving targets with noisy measurements. Our implementation is available online.[2]

A. Stationary Target Case

We first compare the performance of the self-triggered and constant strategies in terms of their convergence speeds and the number of communication messages to achieve a uniform configuration on the boundary of a convex environment. Here, we focus on the base case of known, stationary target position. All results are for 30 trials with the initial positions of robots drawn uniformly at random on the boundary.

Figure 5 shows snapshots of the active tracking process under the proposed self-triggered strategy starting with the initial configuration at time step $k = 1$ (Figure 5-(a)) and ending in a uniform configuration around the target at $k = 760$ (Figure 5-(c)). The robots know the position of the stationary target. The initial positions of the robots are chosen uniformly at random on ∂Q. At each time step, we use the map φ_o to find θ_i on the unit circle (Equation 1), compute the control law as per Algorithm 1, and apply the inverse map φ^{-1} to compute the new positions of the robots on ∂Q. We set $\Delta t = 0.1\,s$ and assume that each robot has the same maximum angular velocity $\omega_{\max} = \frac{\pi}{180}\frac{rad}{s}$. In general, one can use the procedure given in the appendix to compute $\omega_{\max}$ for a given environment.

We first compare the convergence time of the two strategies (Figure 6-(a)). The convergence time, Ctime, is specified as the timestep k when the convergence error, Cerr, drops below a threshold. We use $0.1N$ as the threshold, where N is the number of robots. The convergence error term, Cerr, is defined as:

$$\mathrm{Cerr} = \sum_{i=1}^{N} \left|\theta_i - V^i_{\text{mid}}\right| \tag{13}$$

in the constant communication case, and

$$\mathrm{Cerr} = \sum_{i=1}^{N} \left|\theta_i - gV^i_{\text{mid}}\right| \tag{14}$$

in the self-triggered case.

The average number of communication messages is:

$$\overline{\mathrm{Com}} = \frac{\sum_{i=1}^{N} \mathrm{com}(i, \mathrm{Ctime})}{N \times \mathrm{Ctime}}$$

where $\mathrm{com}(i, \mathrm{Ctime})$ gives the total number of communications of a robot with its neighbors i at the end of Ctime. Figure 6-(b) shows the $\overline{\mathrm{Com}}$ in the self-triggered case. The number of communication messages in the constant communication case is a constant.

Figure 6-(a) shows that the self-triggered mechanism has faster convergence than the constant strategy when the number of robots is large. Intuitively, robots communicating constantly with its neighbors should converge faster. In order to investigate this counter-intuitive finding, we computed, $\overline{\mathrm{Wrd}}$, which is the average motion in the "wrong" direction, defined as

$$\overline{\mathrm{Wrd}} = \frac{\sum_{i=1}^{N} \mathrm{wrd}(i, \mathrm{Ctime})}{N \times \mathrm{Ctime}}$$

Here, $\mathrm{wrd}(i, \mathrm{Ctime})$ gives the total amount of motion in the "wrong" directions for each robot i until the convergence time. For robot i, we use the sign of the difference between its initial orientation and the orientation at the convergence time, i.e., $\theta^0_i - \theta_i(\mathrm{Ctime})$ to find which motion is in the "wrong" direction.

When the number of robots increases, the average distance traveled in the wrong direction becomes larger for the constant communication strategy. Recall that both strategies are guaranteed to converge only asymptotically. The rate of convergence is not known. We conjecture that frequent communication with more robots, especially initially, leads to frequent switching in directions before the robots move towards their final configuration. On the other hand, in a self-triggered strategy the robots commit to a direction and move until the next triggered instance, thereby possibly leading to fewer switches in wrong direction.

B. Moving target with centralized and decentralized EKF

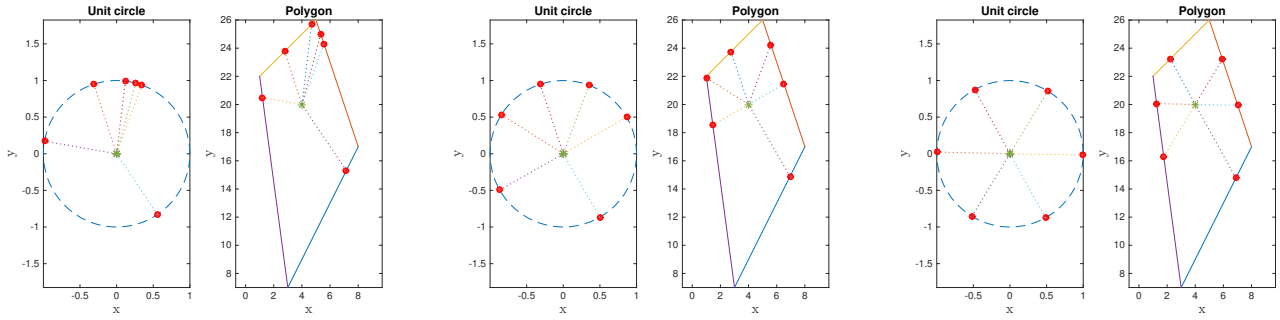
In this section, we present results from simulating the self-triggered and constant strategies for the realistic case of mobile, uncertain targets (Section IV). For both strategies, we consider the centralized EKF and decentralized EKF estimators. We use Cerr to measure the error between robots' deployment and the uniform distribution (Figure 7).

The constant communication strategy with centralized EKF performs the best (Figure 7). Between the two decentralized settings, the constant communication strategy (Figure 8-(a)) has an advantage over the self-triggered one since at each time step every robot obtains three measurements (one from self, and one each from the neighbors) to form a better estimate of the target's position. If all robots have similar (and consistent) estimates, then the strategy is likely to perform better. For self-triggered strategies the robots estimate the target with only their own measurements, except at trigger instances when they get measurements from their neighbors. Figure 8 shows one such instance of the self-triggered mechanism with decentralized EKF where the estimates of robots are uncertain and inconsistent. Our current work is focused on extending the self-triggered strategy taking into account the tracking performance.

VI. Discussion and Conclusion

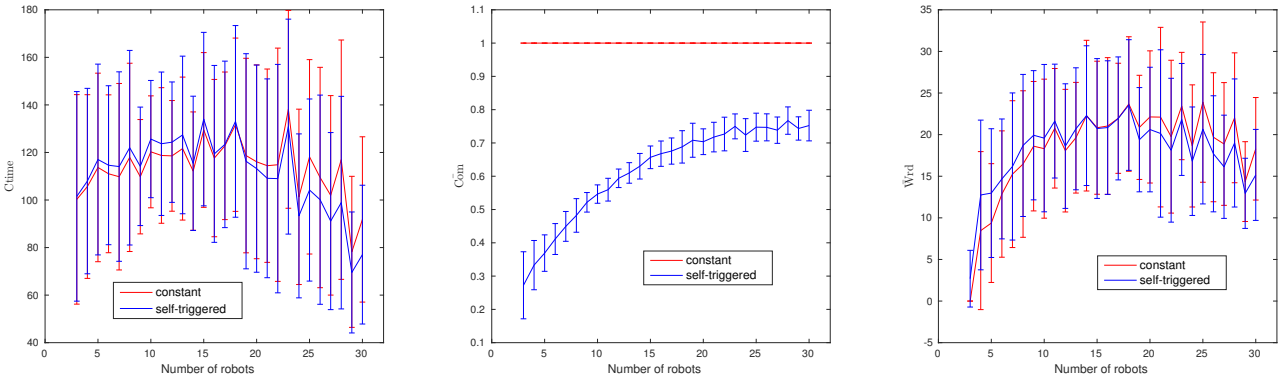
In this paper, we investigate the problem of active target tracking where each robot controls not only its own positions but also decides when to communicate and exchange information with its neighbors. We focused on a simpler target tracking scenario, first studied in reference [9]. We applied a self-triggered coordination strategy that asymptotically converges to a uniform configuration around the target while reducing the amount of communication. We find that the self-triggered strategy performs comparably with the constant communication strategy and in some cases even outperforms

[2] https://github.com/raaslab/Self-triggered-mechanism



(a) k=1 (b) k=400 (c) k=760

Fig. 5. Self-triggered tracking with six robots moving on the boundary of a convex polygon with a known, stationary target. The robots took 760 timesteps to converge to the uniform configuration around the target.



(a) (b) (c)

Fig. 6. Comparison of the convergence time (a) and the number of communication messages (b) and the average motion in the "wrong" direction (c) in constant and self-triggered strategies with a stationary target at known position.

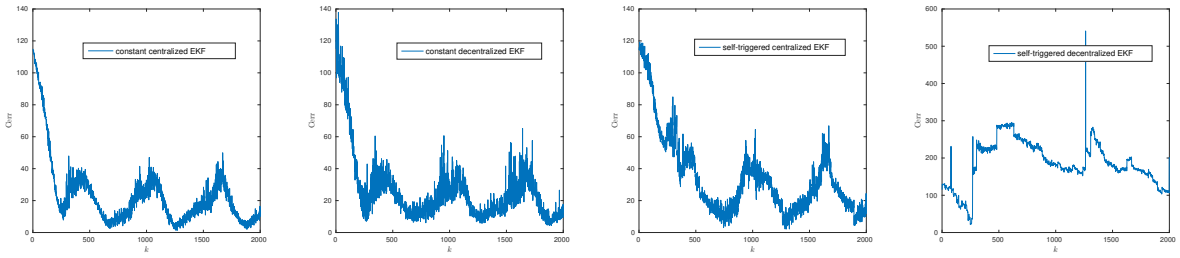


Fig. 7. Convergence error for mobile target tracking with centralized and decentralized EKF estimators steered by the constant strategy and self-triggered strategy

the baseline strategy. We conjecture that frequent communication in the constant communication strategy makes the robots subject to greedy and suboptimal actions than the less informative self-triggered strategy. Further investigating and proving this conjecture are part of our ongoing work. Future work also includes extending the self-triggered strategy to decide not only when to communicate information, but also when to obtain measurements and which robots to communicate with.

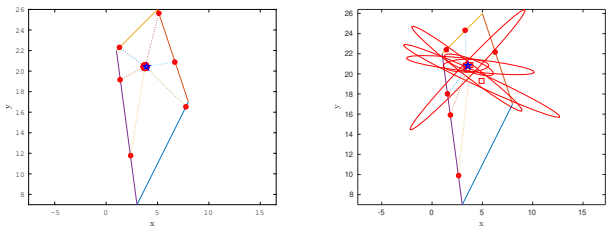
REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[2] B. Rao, H. F. Durrant-Whyte, and J. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 20–44, 1993.

[3] S. Dhillon and K. Chakrabarty, *Sensor placement for effective coverage and surveillance in distributed sensor networks*. IEEE, 2003, vol. 3.

[4] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 16–25, 2006.



(a) (b)

Fig. 8. (a) With a centralized EKF, all robots have the same estimate of the target leading to good convergence. (b) With measurements from self and immediate neighbors in decentralized EKF, the estimates of each robot are highly error-prone and uncertain.

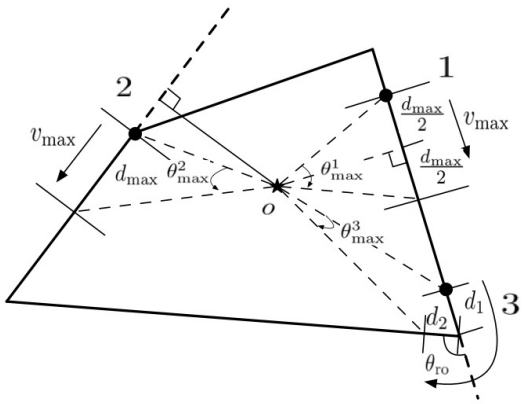


Fig. 9. Computing $\omega_{\max}$.

[5] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014.

[6] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots." in *International Symposium on Robotics Research (ISRR)*, 2015.

[7] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 1, pp. 24 –39, Mar 2012.

[8] P. Tokekar, E. Branson, J. Vander Hook, and V. Isler, "Tracking aquatic invaders: Autonomous robots for invasive fish," *IEEE Robotics and Automation Magazine*, vol. 20, no. 3, pp. 33–41, 2013.

[9] S. MartíNez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[10] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, no. 6, pp. 1077–1087, 2012.

[11] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 3270–3285.

[12] S. E. Hammel, P. T. Liu, E. J. Hilliard, and K. F. Gong, "Optimal observer motion for localization with bearing measurements," *Computers and Mathematics with Applications*, vol. 18, no. 1-3, pp. 171 – 180, 1989.

[13] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, 2009, vol. 501.

[14] W. Evans and J. Sember, "Guaranteed voronoi diagrams of uncertain sites," 2008.

APPENDIX

CALCULATION OF $\omega_{\max}$

Assume robot has a maximum speed $v_{\max}$ with which it can move on $\partial\mathcal{Q}$. Thus, it can move as far as $d_{\max} = v_{\max}\Delta t$ in one time step Δt. We assume that $d_{\max}$ is less than the length of any edge of the polygon. Hence, a robot can cross at most one vertex per time step. Then we split the calculation of $\omega_{\max}$ into three separate cases (Figure 9).

In all cases, let $\mathcal{E}^i$ be the edge on which the robot is located before moving a distance of $d_{\max}$. Let $l\mathcal{E}^i$ be the line supporting the edge. In cases 1 and 2, we compute $\omega_{\max}$ when the robot remains on $\mathcal{E}^i$ after traveling $d_{\max}$, whereas in case 3 the robot goes from $\mathcal{E}^i$ to $\mathcal{E}^{i+1}$

Case 1. The orthogonal projection of the target on $l\mathcal{E}^i$ lies within $\mathcal{E}^i$.

$\omega^{1,\mathcal{E}_i}_{\max}$ corresponds to the case where the robot covers a maximum angular distance with respect to the target in one time step. Thus, the robot should be as close as possible to the target when it moves $d_{\max}$ on the edge. $\omega^{1,\mathcal{E}_i}_{\max}$ can be calculated as $\omega^{1,\mathcal{E}_i}_{\max} = \frac{\theta^{1,\mathcal{E}_i}_{\max}}{\Delta t}$ giving $\omega^1_{\max} = \min_{\mathcal{E}_i\in\mathcal{E}}\{\omega^{1,\mathcal{E}_i}_{\max}\}$. Here $\theta^{1,\mathcal{E}_i}_{\max}$ is the angle shown in Figure 9.

Case 2. The orthogonal projection of the target on $l\mathcal{E}^i$ lies outside $\mathcal{E}^i$.

Similar to case 1, the $\omega_{\max}$ can be computed as $\omega^{2,\mathcal{E}_i}_{\max} = \frac{\theta^{2,\mathcal{E}_i}_{\max}}{\Delta t}$ where $\omega^2_{\max} = \min_{\mathcal{E}_i\in\mathcal{E}}\{\omega^{2,\mathcal{E}_i}_{\max}\}$. Here, $\theta^{2,\mathcal{E}_i}_{\max}$ is larger of the two angles made by the pair of lines joining target and either of the endpoint of $\mathcal{E}_i$ and joining target and a point $d_{\max}$ away from the corresponding endpoint.

Case 3. Robot crosses a vertex $\mathcal{V}_i$ within one time step.

We assume that within one time step Δt, robot moves $d_1^{\mathcal{V}_i}$ on one edge and $d_2^{\mathcal{V}_i}$ on another edge. Since robot must spend some time at the vertex turning in-place, we have $d_1^{\mathcal{V}_i} + d_2^{\mathcal{V}_i} < d_{\max}$. Thus, the $\omega^{3,\mathcal{V}_i}_{\max}$ can be calculated by the equation:

$$\frac{\theta^{3,\mathcal{V}_i}_{\max}}{\omega^{3,\mathcal{V}_i}_{\max}} = \frac{(d_1+d_2)}{v_{\max}} + \frac{\theta^{\mathcal{V}_i}_{\text{ro}}}{\omega_{\text{ro}}} = \Delta t$$

where $\theta^{\mathcal{V}_i}_{\text{ro}}$ and ω_{ro} denote the rotation angle at the vertex $\mathcal{V}_i$ and rotational speed of the robot, respectively and $\theta^{3,\mathcal{V}_i}_{\max}$ is as shown in Figure 9. Then $\omega^3_{\max}$ can be specified as

$$\omega^3_{\max} = \min_{(\mathcal{V}_i, d_1)}\{\omega^{3,\mathcal{V}_i}_{\max}\}.$$

Where $\mathcal{V}_i \in \mathcal{V}$ and $0 \le d_1 \le \Delta t - \frac{\theta^{\mathcal{V}_i}_{\text{ro}}}{\omega_{\text{ro}}}$.

Finally, $\omega_{\max}$ can be computed as

$$\omega_{\max} = \min\{\omega^1_{\max}, \omega^2_{\max}, \omega^3_{\max}\}. \tag{15}$$

If $d_{\max}$ is larger than the length of one edge or the sum of lengths of several edges of the polygon, $\omega_{\max}$ can also be obtained using a similar procedure.