

ENPM 667: Control of Robotic Systems Project 1

Active Target Tracking with Self-Triggered Communications in Multi-Robot Teams: A Review

Lifeng Zhou , *Student Member, IEEE*, and Pratap Tokekar , *Member, IEEE*

Sneha Nayak
UID:115586284
Arun Kumar Dhandayuthabani
UID:116953924

25 November 2019

As part of coursework for Control of Robotic Systems, University of Maryland, College
Park

Abstract

This research paper studies the problem of minimizing communication in decentralized target tracking. A scenario wherein a team of robots can maneuver on the boundary of the testing environment is considered. The fundamental objective is to look for a setup to best follow a target moving inside nature accepting that the robots can quantify distances to the target. Before, decentralized control systems have been proposed which guarantees that the robots asymptotically unite to the ideal configuration. However, these existing methods available always mandate the robots to communicate with each other at all instances. Thus, a new method is proposed with a self-triggered communication strategy. This strategy decides when should one robot inquire up-to date-information from other robot teams available as its neighbors and when it is secure to operate with previous records. For tracking mobile targets, a decentralized Kalman filter with covariance intersection method is employed to share the information of neighboring robots. The paper then presents the approaches through simulations and an experiment applying the strategy in action.

Table of Contents

| | |
|--|----|
| Abstract | 2 |
| 1 Introduction | 4 |
| 2 Background Study | 7 |
| 3 Problem Formulation | 10 |
| 4 Self Triggered Tracking Algorithm | 12 |
| 4.1 Control Law | 12 |
| 4.2 Triggering Policy | 17 |
| 4.3 Convergence Analysis | 18 |
| 5 Practical Cases | 19 |
| 5.1. Tracking of Moving Target with Noisy Measurements | 19 |
| 5.2. Limited Communication and Sensing Range | 21 |
| 6 Simulation and Results | 25 |
| 7 Conclusions and Discussions | 37 |
| References | 38 |
| Appendix | 40 |

Section 1: Introduction

Target tracking can be defined in simple terms as:

"Target Tracking is the prediction of the future location of a dynamical system based on its estimates and measurements."

Target tracking depends on the environment in which the tracking algorithm is designed and employed. The number of targets to track introduces us to Single Target tracking (STT) and Multiple Target tracking (MTT). The single target tracking (STT) system is widely used to redirect the sensor pointing angle to follow the target, whereas the MTT covers a certain area commonly called the field of view (FOV). Within the FOV several sensors can receive measurements at irregular intervals, which are required to be assigned to the possible multiple targets by a complex data association logic. Modern MTT techniques combine data association and Kalman filtering to reduce the errors.

A precise definition of target tracking found in "*Design and Analysis of Modern Tracking Systems*" by Samuel Blackman and Robert Popoli, 1999 states as:

"The target tracking objective is to collect sensor data from a field of view containing one or more potential targets of interest and to then partition the sensor data into sets of observations, or tracks, that are produced by the same sources. Once tracks are formed and confirmed (so that background and other false targets are reduced), the number of targets can be estimated and quantities, such as target velocity, future predicted position, and target classification characteristics, can be computed for each track." (qtd. in Blackman, Popoli 1999)

Target tracking is one of the more well-studied problems in the robotics community and finds many applications like:

- Surveillance

- Crowd monitoring
- Wildlife monitoring

Active target tracking methodology is employed with a team of robots wherein the focal point is on active control of the state of the robot. Conventionally, the robots exchange information with each other and afterward choose an approach to move as to best track the target (static/moving). Standard design strategies are utilized expecting that the robots will exchange their information at each time regardless of whether that data merits exchange as exchanging data is advantageous. Cost of communication between robots is costly considering smaller robots where radio communication can be a significant source of power consumption. The robots can extend their lifespan by reducing the time spent communicating (equivalently, the quantity of messages sent). The goal is thus to determine a strategy that communicates whilst required without extensively affecting the tracking performance.

After investigating this issue in a simple target tracking situation where the robots are confined to move on the boundary of a convex environment, we find that they acquire distance measurements toward a target moving in the interior part of the environment. The objective of the robots is to position themselves to maximize the information gained from the target. Our arrangement models circumstances where the robots cannot enter the interior part of the environment. To get an overview of how this arrangement model works, we explore into the study of pursuit-evasion of UAV drones, where the pursuers aren't allowed to enter "no-fly zones". If the evader enters a "no-fly zone", then the pursuers reposition themselves on the perimeter of a convex approximation of the zone. Additionally, we explore the application of tracking radio-tagged fish using ground robots that can move only along the boundary of the environment to get more suited with the problem setup.

A constant decentralized strategy proposed by Martínez and Bullo [1] where the robots communicate at all time steps with their neighbors and attain the ideal (uniform) configuration is taken as reference. Now, applying a self-triggered coordination algorithm from [1],[11] to the target tracking situation wherein each robot decides when to trigger communications with its

neighbors. The performance of the newly applied strategy is compared with the prevailing constant strategy with the help of simulations.

Subsequently, the problem of robots acquiring noisy measurements of the distance to the targets is examined. In a decentralized setting, robots can exchange information only with their nearby neighbors. As a result, their local estimates of the target's position may differ substantially, resulting in poor monitoring, whilst the robots are not in a uniform configuration. Implementing a decentralized Kalman filter estimator fuses the beliefs shared by neighboring robots (at triggered instances) to a common estimate help counter these errors.

The main results obtained from the paper assume that the robots have sufficiently large communication and sensing ranges. In Section V-B, we present necessary conditions on the sensing and communication ranges for the results to hold. Simulation results validate the theoretical evaluation, showing that the self-triggered strategy converges to the optimal, uniform configuration. The average number of communications is much less than 30% that of the regular strategy. Demonstration of the overall performance of the algorithm is done via experiments applying the newly suggested strategy with five simulated and two actual robots coordinating with each other

The rest of the report is organized as follows. Background study related to the paper are available in Section II. Section III introduces to the problem. The self-triggered tracking algorithm can be found in Section IV, with the assumption that the target's position is known and fixed. Section V presents two practical extensions without assumptions for noisy measurements and limited sensing and communication ranges. Section VI contains the simulations and results.

Section 2: Background Study

Multi-robot target tracking has been widely studied in robotics. A number of the key findings in the field of multi-robot target tracking are indexed underneath:

- Robin and Lacroix [3] reviewed multi-robot target tracking and detection frameworks and gave a scientific classification of important works.
- Khan et al. [4] classified and talked about control strategies for multi-robot , multi-target observing and recognized the significant components of communication issue between the systems.
- Hausman *et al.* [5] proposed a centralized cooperative technique for a team of robots to estimate a moving target. They showed how to use onboard sensing with limited sensing capabilities and also switching the sensor topology for effective target tracking.
- Dias *et al.* [6] proposed a multi-robot triangulation way to deal with initialization and data association issues in bearing-only sensors. The robot communicates locally to exchange and update the estimate beliefs of the target by with the help of a decentralized filter.
- Franchi *et al.* [7] introduced a decentralized technique to guarantee that the robots pursue the target while moving around it in circular fashion. They assume that the robots are labeled. Similar to the paper reviewed, the robots attempt to maintain a uniform distribution on a (moving) circle around the target. However, unlike the research paper, they mandate that the robots constantly communicate with their local neighbours.
- Sung *et al.* [8] proposed a dispersed approach for multi-robot assignment problem for multi-target tracking by considering both sensing and communication ranges. The objective of their work is

also additionally to confine the communication between the robots. However, they achieved it by limiting the number of messages sent at each time step but allowed the robots to communicate at all time steps. Instead, this research paper explicitly determines when to trigger communication with other robots.

This research paper builds on event-triggered and self-triggered communication schemes studied primarily by the control's community [2], [9]. Dimarogonas *et al.* [10] presented both centralized and decentralized event-triggered strategies for the agreement problem in multi-agent systems. They extended the outcomes to a self-triggered communication setting where the robot calculates its next communication time based on the past instance without monitoring the state error.

Data Association:

For the tracking system to perform properly, the most likely measured potential target location should be used to update the target's state estimator. This is generally known as the data association problem. The probability of the given measurement being correct is a distance function between the predicted state of the target and the measured state. Note that state is not limited to position; it may also consist of features such as color. This becomes especially important for targets that may come close to or cross one another, such as moving objects . Popular association algorithms for single-target applications are based on the following schemes:

- *Nearest Neighbor*: This algorithm always updates the tracking filter with the measurement closest to the predicted state.
- *Multi-Hypothesis Track Splitting*: This scheme creates a new hypothesis track for every measurement that is in the validation region, and prunes unlikely tracks using a likelihood ratio.
- *Probabilistic Data Association*: Each measurement affects the tracking filter to a degree based on the probability that it is correct given the predicted state.
- *Optimal Bayesian Filter*: This variation of Probabilistic Data Association splits multiple tracks, like the Multi-Hypothesis algorithm, and eliminates unlikely tracks.

For multiple sensors and multiple targets, the problem becomes increasingly complex. Common association algorithms are:

- *Joint Likelihood*: This variation on the Multi-Hypothesis Track Splitting algorithm above extends to multiple tracks.
- *Joint Probabilistic Data Association*: This algorithm updates the filter for each track based on a joint probability of association between the latest set of measurements and each track.

Multiple Hypothesis Joint Probabilistic: This variation of the Optimal Bayesian Filter uses joint probabilities among multiple track associations for multiple hypotheses. It is by far the most computationally complex algorithm and requires intelligent pruning techniques. It is NP-complete, which provides considerable incentive to find non-exhaustive ways to search the space of possible associations to maximize the joint probability.

Section 3: Problem Formulation

Here a group of N robots is considered to be moving along the boundary ∂Q of a convex polygon $Q \subset \mathbb{R}^2$. The position of the robots (which we consider to be first-order continuous-time motion models) is denoted by p_1, \dots, p_N and the task of these robots is to track a target positioned at o , that is located within Q . Any point on the boundary of Q can be mapped to a unit circle \mathbb{T} using the transformation $\phi_o : \partial Q \rightarrow \mathbb{T}$, and is given by

$$\phi_o(p) = \frac{p - o}{\|p - o\|} \quad (1)$$

The above gives the transformation of every robot's position with respect to a unit circle which belongs to \mathbb{T} , thus indicating the robot's location on the circle \mathbb{T} .

$$\theta = (\theta_1, \dots, \theta_N) \in \mathbb{T}^N,$$

denotes the vector locations of all the robots, where

$$\theta_i = \phi_o(p_i) \in \mathbb{T} \quad \text{and}$$

$$p_i \in \partial Q \subset \mathbb{R}^2, \quad \text{where } i \text{ represents each robot}$$

The following is true about each of the N robots:

1. Each robot is aware of its position at all times.
2. Neighboring robots can communicate with each other and exchange their positions
3. Each of these robots have sensors, which gives them the capability to cover the entire environment.
4. We assume ω_{\max} to denote the common maximum angular speed for all robots on the unit circle.
5. The optimal configuration between each pair of neighboring robots is given by

$$\theta_{i+1} - \theta_i = 2\pi/N, \forall i \in \{1, \dots, N\}.$$

in order to obtain distance measurements of the target. This is defined with respect to FIM (Fisher's Information Matrix), which is commonly used for active target tracking.

According to Martínez and Bullo [1], when neighboring robots are capable of constant communication at every step (known as control strategy), i.e. robot i can communicate with only $i-1$ and $i+1$, a decentralized control law can be guaranteed to converge to a uniform configuration. This can be done when the target is stationary and when its location is known to all robots. But in the case when the target is not stationary, noisy measurements can be obtained and estimated with the use of Extended Kalman filter (EKF).

The main idea behind this paper is based on the above control law but with minimum communications between robots, while maintaining the convergence properties associated with this decentralized control law. This strategy, known as the self-triggered strategy, not only decides how and when each robot should move but also when they should communicate among themselves to exchange relevant information.

Section 4: Self Triggered Tracking Algorithm

Stationary Target

In order to achieve a uniform configuration along a unit circle, we are required to know the fixed location of the target (we take this to be the center of the circle). These assumptions are necessary to hold convergence properties. The algorithm that is being used here achieves Voronoi partitions in a convex Environment, which is needed for Optimal deployment. This algorithm, that is invigorated by the self-triggered centroid algorithm, uses a decentralized control law to achieve optimality.

A motion prediction set of the neighbors of each of the robots, which contains the last known positions of the neighbors, is employed in order to formulate a control law. Followed by this is an update policy which is to decide if a robot should communicate with its neighbors in order to exchange information.

4.1. Control Law

The constant control law steers robot i toward the midpoint of its Voronoi segment V_{mid}^i which is the part of the unit circle extending from $(\theta_{i-1} + \theta_i)/2$ to $(\theta_{i+1} + \theta_i)/2$. This movement towards the midpoint involves exchange of real-time information between its neighbors, θ_{i-1} and θ_{i+1} , as illustrated in Fig. 4.1.

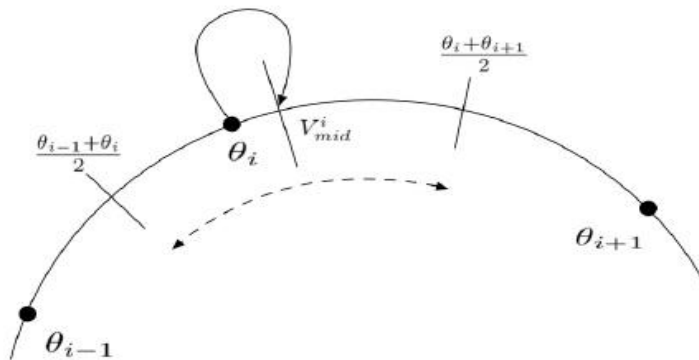


Fig 4.1 Robot i go towards the midpoint of its Voronoi segment via exact information from its neighbors.

However, in our case where we use distributed self-triggered strategies, this real time exchange does not occur, which leads to inexact data being taken in by the algorithm.

The information that each robot i holds about its neighbor j is the last known position of j , denoted by θ_j^i , and the time elapsed since the position of robot j was collected, denoted by τ_j^i . Robot i can then compute the furthest distance that j could have moved in τ_j^i time as $\pm\phi_j^i$ where,

$$\phi_j^i = \omega_{max}\tau_j^i \quad (2)$$

Robot i can now build a prediction motion set $\mathcal{R}_j^i(\theta_j^i, \phi_j^i)$, that contains all the possible locations where robot j could have moved to in τ_j^i time (Fig. 4.2).

In our algorithm, robot i is needed to only communicate with its neighbors $i - 1$ and $i + 1$. The prediction motion range that robot i stores is given as,

$$\mathcal{R}^i := \{\mathcal{R}_{i-1}^i(\theta_{i-1}^i, \phi_{i-1}^i), \mathcal{R}_{i+1}^i(\theta_{i+1}^i, \phi_{i+1}^i)\}$$

In this self-triggered mechanism, we can only use the prediction model to move the robot and this uses inexact information of the neighbors, which is why the midpoint of its Voronoi segment is a set instead of a point (Fig. 4.3)

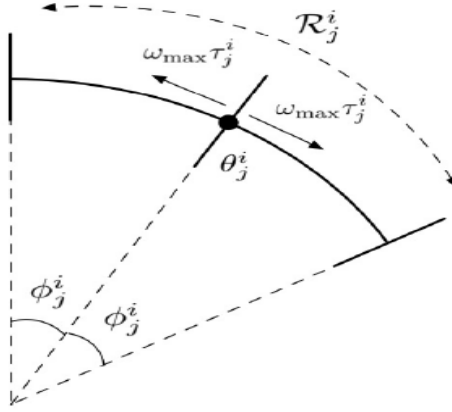


Fig 4.2 Motion prediction set, \mathcal{R}_{i-1}^i , that each robot i maintains for its neighbors j . θ_j^i is the last known position of the robot j and τ_j^i is the time elapsed since this last known position.

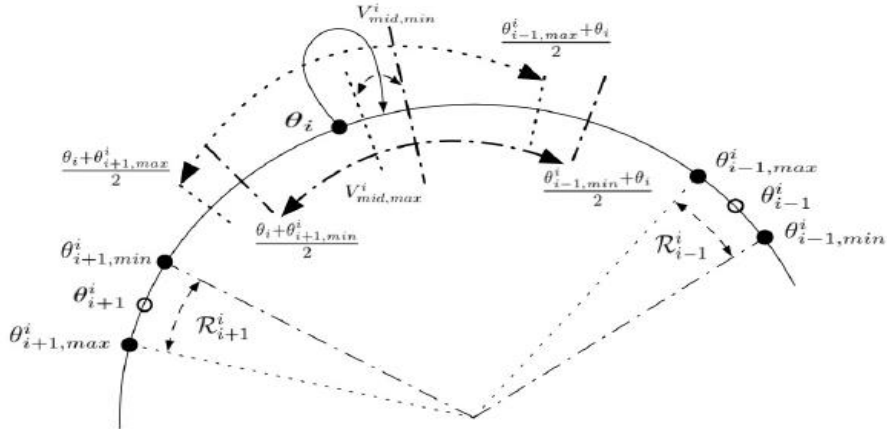


Fig 4.3 Robot i go towards the midpoint of its Voronoi segment via inexact information from its neighbors.

Define,

$$\theta_{i-1,min}^i = (\theta_{i-1}^i - \phi_{i-1}^i), \quad \theta_{i-1,max}^i = (\theta_{i-1}^i + \phi_{i-1}^i)$$

$$\theta_{i+1,min}^i = (\theta_{i+1}^i - \phi_{i+1}^i), \quad \theta_{i+1,max}^i = (\theta_{i+1}^i + \phi_{i+1}^i)$$

Thus, we have,

$$\mathcal{R}_{i-1}^i (\theta_{i-1}^i, \phi_{i-1}^i) = \{ \beta \in \mathbb{T} \mid \theta_{i-1,min}^i \leq \beta \leq \theta_{i-1,max}^i \}$$

$$\mathcal{R}_{i+1}^i (\theta_{i+1}^i, \phi_{i+1}^i) = \{ \beta \in \mathbb{T} \mid \theta_{i+1,min}^i \leq \beta \leq \theta_{i+1,max}^i \}$$

Then the min and max midpoints of robot i 's Voronoi segment can be computed as,

$$V_{mid,min}^i = \frac{(\theta_{i-1,min}^i + \theta_i)/2 + (\theta_i + \theta_{i+1,min}^i)/2}{2} \quad (3)$$

$$V_{mid,max}^i = \frac{(\theta_{i-1,max}^i + \theta_i)/2 + (\theta_i + \theta_{i+1,max}^i)/2}{2} \quad (4)$$

The midpoint of its Voronoi segment $V_{mid}^i \in [V_{mid,min}^i, V_{mid,max}^i]$, i.e.

$$V_{mid,min}^i \leq V_{mid}^i \leq V_{mid,max}^i \quad (5)$$

Substituting 3, 4 and 5.

$$\begin{aligned} \frac{\theta_{i+1}^i + 2\theta_i + \theta_{i-1}^i - 2\omega_{max}\tau^i}{4} &\leq V_{mid}^i \\ V_{mid}^i &\leq \frac{\theta_{i+1}^i + 2\theta_i + \theta_{i-1}^i + 2\omega_{max}\tau^i}{4} \end{aligned}$$

Then

$$\frac{-\omega_{max}\tau^i}{2} \leq V_{mid}^i - \frac{\theta_{i+1}^i + 2\theta_i + \theta_{i-1}^i}{4} \leq \frac{\omega_{max}\tau^i}{2}$$

Therefore

$$\left| V_{mid}^i - \frac{\theta_{i+1}^i + 2\theta_i + \theta_{i-1}^i}{4} \right| \leq \frac{\omega_{max}\tau^i}{2} \quad (6)$$

$$gVs_i = \left\{ \beta \in \mathbb{T} \mid \max_{\theta_i \in S_i} |\beta - \theta_i| \leq \max_{\theta_j \in S_j} |\beta - \theta_j|, \quad \forall j \neq i \right\}$$

where $T_1, \dots, T_n \subset T$ are a set of connected segments in T .

The guaranteed Voronoi segment of robot i can be computed as,

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

Instead of moving to actual midpoint of the Voronoi segment (V_{mid}^i) which isn't known, the robot i can move towards the guaranteed Voronoi segment gV_{mid}^i , which is given by,

$$gV_{mid}^i = \frac{(\theta_i + \theta_{i+1,min}^i)/2 + (\theta_{i-1,max}^i + \theta_i)/2}{2} = \frac{\theta_{i+1}^i + 2\theta_i + \theta_{i-1}^i}{4} \quad (8)$$

We cannot guarantee the fact that when the robot moves along gV_{mid}^i , it moves closer to the midpoint of its Voronoi segment.

There are certain conditions, however, where this is true.

Lemma 1: Let robot i move from θ_i towards gV_{mid}^i . Let θ'_i be the new position after one-time step. Only if $|\theta'_i - gV_{mid}^i| \geq |V_{mid}^i - gV_{mid}^i|$, then $|\theta'_i - V_{mid}^i| \leq |\theta_i - V_{mid}^i|$.

Now the robot i just has to ensure that its new position θ'_i satisfies $|\theta'_i - gV_{mid}^i| \geq |V_{mid}^i - gV_{mid}^i|$. This indirectly assures that the distance between the actual midpoint (V_{mid}^i) and the robot doesn't increase.

However, the right-hand side of the above inequality isn't strong enough, since V_{mid}^i is unknown to the robot. Thus, we can set an upper bound using (6), which is given by $ubd_i := \frac{\omega_{max}\tau^i}{2}$.

Corollary 1: Suppose robot i moves from θ_i towards gV_{mid}^i . Let θ'_i be the position after a single time step. If

$$|\theta'_i - gV_{mid}^i| \geq ubd_i \quad (9)$$

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

Now we move on to what configurations the robots take on the circle and what control law steers them.

Since we already compute the guaranteed Voronoi segment from (7) from its last known location of its neighbors as well as its real time positions.

The mid-point of the guaranteed Voronoi segment (8) and can move towards the mid-point until it stays within the ubd_i .

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

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

Where

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

4.2. Triggering Policy

When ubd_i grows larger until the condition in (9) isn't met, as time elapses, the robot is triggered to collect important information from its neighbor. There can be 2 scenarios when this could happen to this:

- The ubd_i grows larger as time elapses. If not, the robot i moves towards gV^i_{mid} which would make the ubd_i smaller.
- When the robots are close to convergence which leads to frequent triggering. For this particular case we use a user-defined parameter $\sigma \geq 0$, to ease the triggering condition.

In either of these 2 cases, when the robot needs to trigger communication, the following condition is violated:

$$ubd_i < \max\{\|\theta' - gV^i_{mid}\|, \sigma\}$$

The robots must also satisfy the following condition in order to account for any interchange of swapping of robots with their neighbors. The condition is as follows:

$$\theta_{i+1}^i - \omega_{max}\tau_{i+1}^i > \theta_i > \theta_{i-1}^i + \omega_{max}\tau_{i-1}^i$$

In any case where this condition is violated, a new communication is triggered.

The algorithm for this is shown as below.

Algorithm 1 Self-Triggered Midpoint

```

1: while all robots have not converged:
2:   for each robot  $i \in \{1, \dots, N\}$  perform:
3:     increment  $\tau_{i-1}^i$  and  $\tau_{i+1}^i$  by  $\Delta t$ 
4:     compute  $\mathcal{R}^i$ ,  $gVs_i$ ,  $gV_{mid}^i$ , and  $ubd_i$ 
5:     if condition in (11) OR condition in (12) is violated:
6:       trigger communication with  $i + 1$  and  $i - 1$ 
7:       reset  $\tau_{i+1}^i$  and  $\tau_{i-1}^i$  to zero
8:       recompute  $\mathcal{R}^i$ ,  $gVs_i$ ,  $gV_{mid}^i$ , and  $ubd_i$ 
9:     end if
10:    compute and apply  $u_i$  as defined in (10)
11:  end for
12: end while

```

4.3. Convergence Analysis

We see that the above presented algorithm is guaranteed to converge asymptotically, to a configuration such that all the robots are equidistant from each other on the unit circle. This is irrespective of the initial configuration, under the assumption that any two robots are not co-located.

Section 5: Practical Cases

Let's look at two practical extensions of our algorithm relaxing some of the assumptions given in Section III.

5.1. Tracking of Moving Target with Noisy Measurements

On the off chance that the actual position of the target o^* is known, then a unit circle centred at the target can be drawn and we can use the methodology employed in Algorithm 1 to merge to a standardized configuration along the circle. From the results obtained in [9], this design maximizes the determinant of the FIM (Fisher Information Matrix). In practical cases, however, we do not know the actual position of the target. The main goal is to utilize the use of the noisy measurements from the robots to estimate the position of the target. In addition, the target may be mobile. This suggests that the unknown centre of the circle (centre considered as the target) is also moving, furthermore complicating the control strategy for the robots.

Kalman filtering, also known as **linear quadratic estimation (LQE)**, is an optimal estimation algorithm that uses a series of measurements (mostly indirect) observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe. The filter is named after Rudolf E. Kálmán, one of the primary developers of its theory. Common application of Kalman Filters includes guidance and navigation systems, control systems, computer vision systems and signal processing. One of the very first application of Kalman Filter was on the Apollo Project where the engineers used it to estimate trajectories of the manned spacecraft to the Moon and back.

State estimators are one of the de-facto elements present in Kalman filters. Here, the paper deals with a non-linear system and hence an extended Kalman filter (EKF) is used which linearizes the non-linear function around the mean of the current state estimate. At each time step, the linearization is performed locally, and the resulting Jacobian matrices are then used in the prediction and update states of the Kalman filter algorithm.

We utilize an EKF that estimates the position of the target (*i.e.*, centre) and predicts its motion at every time step. The prediction and the estimate of the target from an EKF is a 2-D Gaussian distribution parameterized by its mean, $\hat{o}(k)$, and covariance, $\hat{\Sigma}(k)$. The target's state prediction and update by EKF are described in the following.

Prediction:

$$\begin{aligned}\hat{o}^-(k) &= \hat{o}(k-1) \\ \hat{\Sigma}^-(k) &= \hat{\Sigma}(k-1) + R(k).\end{aligned}$$

Update:

$$\begin{aligned}K(k) &= \hat{\Sigma}^-(k)H^T(k)(H(k)\hat{\Sigma}^-(k)H^T(k) + Q(k))^{-1} \\ \hat{o}(k) &= \hat{o}^-(k) + K(k)(z(k) - h(\hat{o}^-(k))) \\ \hat{\Sigma}(k) &= (I - K(k)H(k))\hat{\Sigma}^-(k)\end{aligned}$$

where $R(k)$ = Covariance matrix of the noise from target's motion model $Q(k)$ = Covariance matrix of the noise from target robot's measurement.

$h(\hat{o}^-(k)) = \|\hat{p}(k) - \hat{o}^-(k)\|_2$, $z(k)$ denotes the noisy distance measurement from the robot, and $H(k)$ is the Jacobean of $h(\hat{o}^-(k))$. 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 given by:

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

In the centralized strategy, a common fusion center can obtain the measurements from all the robots and compute a single target estimate, $\hat{o}(k)$, at every time step. Consequently, each robot will have equivalent estimated mean, $\hat{o}(k)$, and therefore the same center for the unit circle. However, in the decentralized case, each robot runs its own EKF estimator and has its own target estimate, $\hat{o}^i(k)$, based on its own measurements of the target. From the observations above, the centers of the unit circle will not be the same, making convergence challenging.

On the off chance that at any time step, a robot communicates with its neighbors, then it can share its estimate, mean $\hat{o}(k)$ and covariance Σ^k , with its neighbors. Therefore, at these triggered instances, each robot can update its own estimate by fusing the estimates from its neighbors. We use the covariance intersection algorithm, which is a standard decentralized EKF technique, to fuse estimates under unknown corrections.

The covariance intersection algorithm takes two Gaussian beliefs, (x_a, Σ_a) and (x_b, Σ_b) , and combines them into a common belief, (x_c, Σ_c) :

$$x_c = \Sigma_c((\Sigma_a)^{-1}x_a + (\Sigma_b)^{-1})^{-1}x_b$$

$$\Sigma_c = (\lambda(\Sigma_a)^{-1} + (1 - \lambda)(\Sigma_b)^{-1})^{-1}$$

where $\lambda \in [0,1]$ is a design parameter obtained by optimizing some criteria, i.e., determinant or trace of Σ_c .

The rest of the process is like that in Algorithm 1. The centralized EKF scheme is a baseline that is compared against for the realistic decentralized strategy. The results are presented in Section VI.

5.2. Limited Communication and Sensing Range

The main result assumes that the robots have sufficiently large communication band and sensing range. In this section, we first present a necessary condition for the communication range r_c and sensing range r_s . Then, we present a condition on the communication range for a modified version of the employed decentralized algorithm.

Theorem 1 (Necessary Condition):

Let N be the total number of robots. To ensure the convergence to the best configuration when the robots do not know N , the communication range $r_c < D_{in} * \sin(\pi/N)$ and the sensing range $r_s < D_{in}/2$ where D_{in} = the diameter of the largest radius circle contained completely inside the environment.

Proof:

Consider an arbitrary convex boundary as shown in Fig. 5(a). We draw its inscribed circle C_{in} with radius r_{in} and diameter D_{in} . To ensure convergence

without knowing N , the robots must be able to communicate with both its nearby neighbors when they reach a uniform configuration. When $N=4$ [see Fig. 5(a)], if the communication range among any two robots $r_c < 2 - \sqrt{r_{in}} = (2 - \sqrt{D_{in}}/2)$, these four robots cannot communicate with each other, even when they are at the uniform configuration. For any N , r_c can be calculated by using the cosine law:

$$\begin{aligned} r_c^2 &= 2r_{in}^2 - 2r_{in}^2 \cos(2\pi/N). \\ &= 2r_{in}^2 (1 - \cos(2\pi/N)) \\ &= 2r_{in}^2 * [2[\sin(\pi/N)]^2] \end{aligned}$$

Thus,

$$r_c = 2r_{in} * \sin(\pi/N) = D_{in} * \sin(\pi/N).$$

Thus, we obtain the necessary condition that $r_c < D_{in} * \sin(\pi/N)$.

If $r_s < r_{in}$, no robot can sense the target when the target is at the center of the circle. Thus, the sensing range r_s cannot be less than $D_{in}/2$.

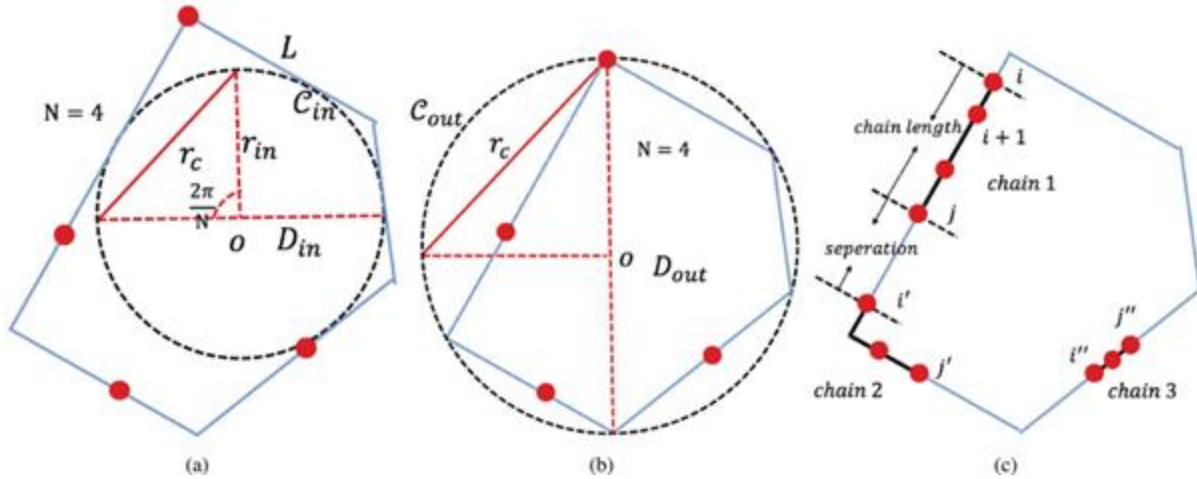


Fig5: Example of four robots moving on an arbitrary convex boundary to show (a) necessary, and (b), (c) sufficient conditions. Red solid circle: robot.

Next, we propose a sufficient condition on the communication and sensing ranges to ensure convergence to the uniform configuration. We need to make an additional assumption that each robot can uniquely identify its

forward and backward neighbors. We also assume that the communication range is the same for all the robots and is known to all the robots. Now presenting a modified version of the self-triggered strategy that works with limited communication range.

Modified Self-Triggered Strategy: If robot i cannot communicate with either of its two neighbors, it does not move. If robot i can only communicate with one of its neighbors, it moves in the direction of the other neighbor with maximum velocity. A robot keeps moving unless its motion will cause it to lose communication with its neighbors. If robot i can communicate with both of its neighbors, it applies the proposed control law

Theorem 2 (Sufficient Condition):

If the communication range $r_c \geq L/N$ and the sensing range $r_s \geq D_{out}$, then the modified strategy converges to the optimal configuration. Here,

L = the environment's perimeter

D_{out} = the length of the longest segment contained completely inside the environment.

Proof:

Let's define a *communication chain* [see Fig. 5(c)] to be the maximal arrangement of consecutive robots, $i, i+1, \dots, j$, such that i can communicate with $i+1$, $i+1$ can communicate with $i+2$, and so on until j . We present that irrespective of the starting setup implementing the modified control law, all the N robots will shape a single chain.

Also let's characterize the length of a chain to be the distance along the boundary (in the direction of the chain) between the two robots in extreme end of the chain [see Fig. 5(c)]. We signify these two extreme robots as the robots at the two endpoints of the robot chain.

Consider a chain of K robots. We show that the robots in this chain will continue to keep moving unless the length is greater than or equal to $(L/N) \cdot (K-1)$ or the chain merges with another. Extreme robots in a chain have only one neighbor to communicate with. As per the strategy, these robots will continuously move (in a direction away from the chain) with

maximum velocity. Other robots between the two extreme robots in the communication chain apply self-triggered control law (10) to go toward the midpoint of its two neighbors. Accordingly, the length of the chain keeps increasing as long as the robots are moving. Except if the the chain converges with another, the robots will stop moving when the distance between all consecutive pairs of robots is r_c . Here, $r_c \geq L/N$. In the event that two consecutive robots are on the same environment edge, then the distance along the boundary between the robots is exactly equal to r_c . If the two robots are on different boundary edges, then the distance between the robots along the boundary will be greater than r_c (due to the convexity of the environment). Thus, the length of the chain when all K robots stop moving will be greater than or equal to $(L/N)(K-1)$.

Next, let's demonstrate and prove the claim that eventually all robots form a single chain, by contradiction. Signify the *detachment* between two consecutive chains as the separation distance between the starting (ending) robot of one chain and the ending (starting) robot of another chain along the boundary of the environment [see Fig. 5(c)]. Assume, for logical contradiction, that there exists $M > 1$ chains after all robots have stopped moving. Let K_1, \dots, K_M be the number of robots in M chains. $K_1 + \dots + K_M = N$.

The separation between any two sequential chains is strictly greater than L/N . Moreover, the length of any robot chain is greater than or equal to $(L/N)(K_i - 1)$. The perimeter of the environment must be equal to the length of all chains and the separation between all consecutive chains. Therefore, the perimeter must be strictly greater than:

$$\frac{L}{N}(K_1 - 1) + \dots + \frac{L}{N}(K_M - 1) + M \frac{L}{N} = L.$$

This negates the fact that the perimeter of the environment is exactly L . Thus, we demonstrate that all the robots eventually form a single chain.

Finally, once we guarantee that robots form a single chain, then the convergence proof follows from the convergence of the self-triggered policy.

Section 6: Simulations and Results

In this section, we examine the performance of the proposed self-triggered tracking coordination algorithm. First, we compare the convergence time for the self-triggered and constant or periodic communication strategies to achieve a uniform configuration on a convex boundary. Then, we demonstrate the performance of the self-triggered and constant strategies for moving targets. The code for this is available in the following github Repository:

https://github.com/snehanyk05/Controls_Project

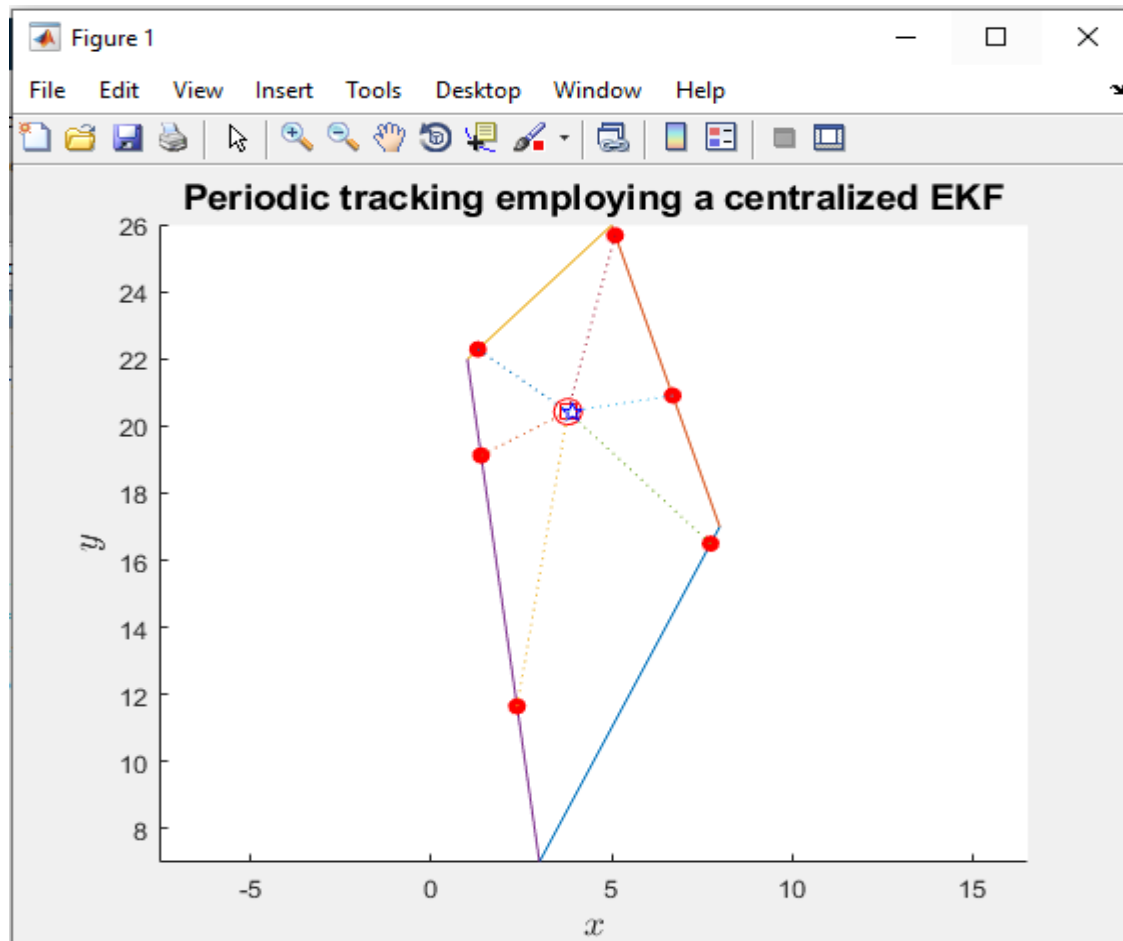


Fig 6.1: Constant communication strategy with centralized EKF on a polygon boundary

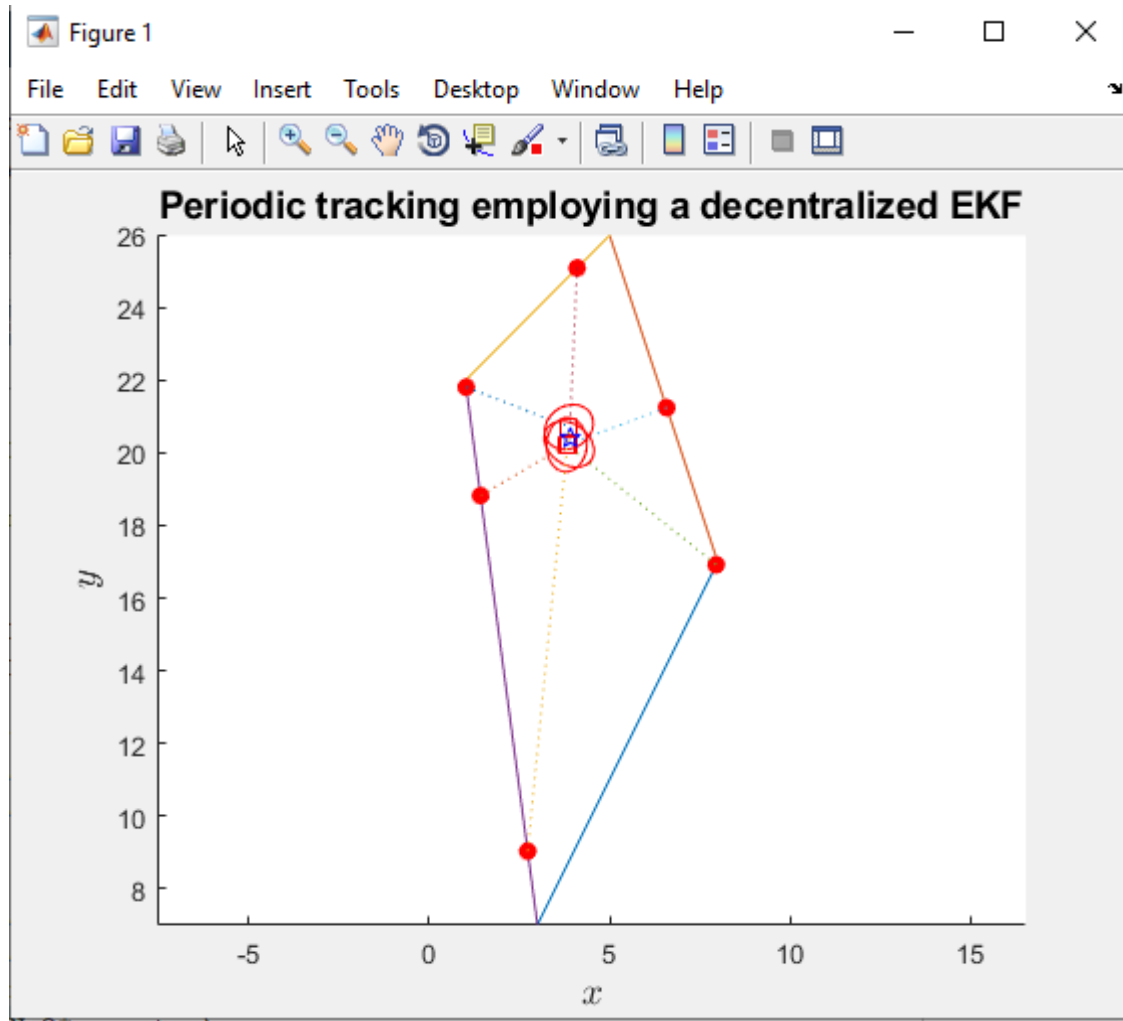


Fig 6.2: Constant communication strategy with decentralized EKF on a polygon boundary

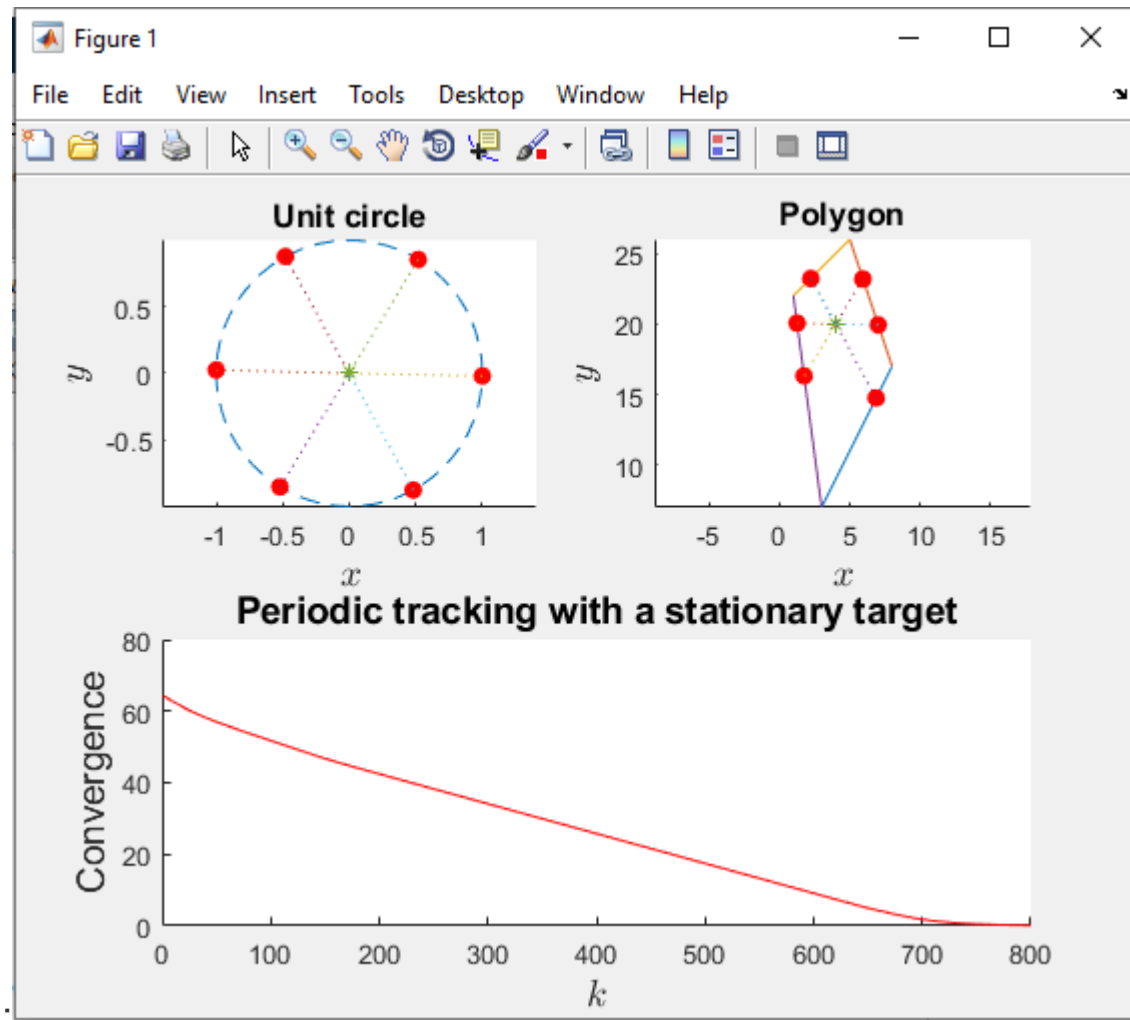


Fig. 6.3 shows snapshot of the active tracking process under the constant communication strategy starting with the initial configuration at time step $k = 1$ and ending in a uniform configuration around the target at $k = 760$

Here, the convergence time is specified as the time step k when the convergence error, C_{err} , drops below a threshold. Setting $0.1N$ as the threshold, where N is the number of robots. The convergence error term, C_{err} , is defined as:

$$C_{err} = \sum_{i=1}^N |\theta_i - V_{mid}^i|$$

for constant communication strategy and

$$C_{err} = \sum_{i=1}^N |\theta_i - g V_{mid}^i|$$

For the proposed self-triggered strategy.

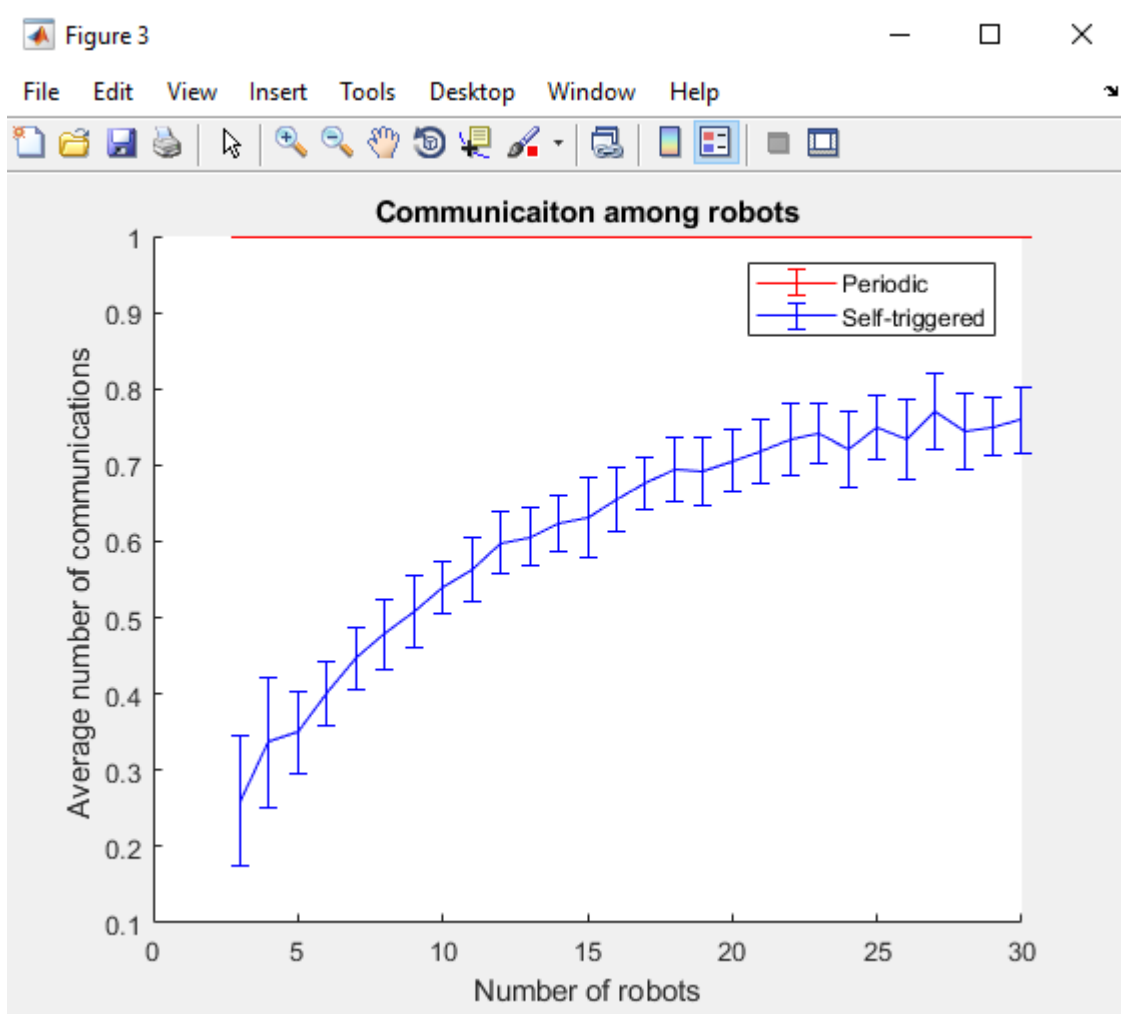


Fig 6.4: Comparison of number of communication messages in periodic and self-triggered strategy.

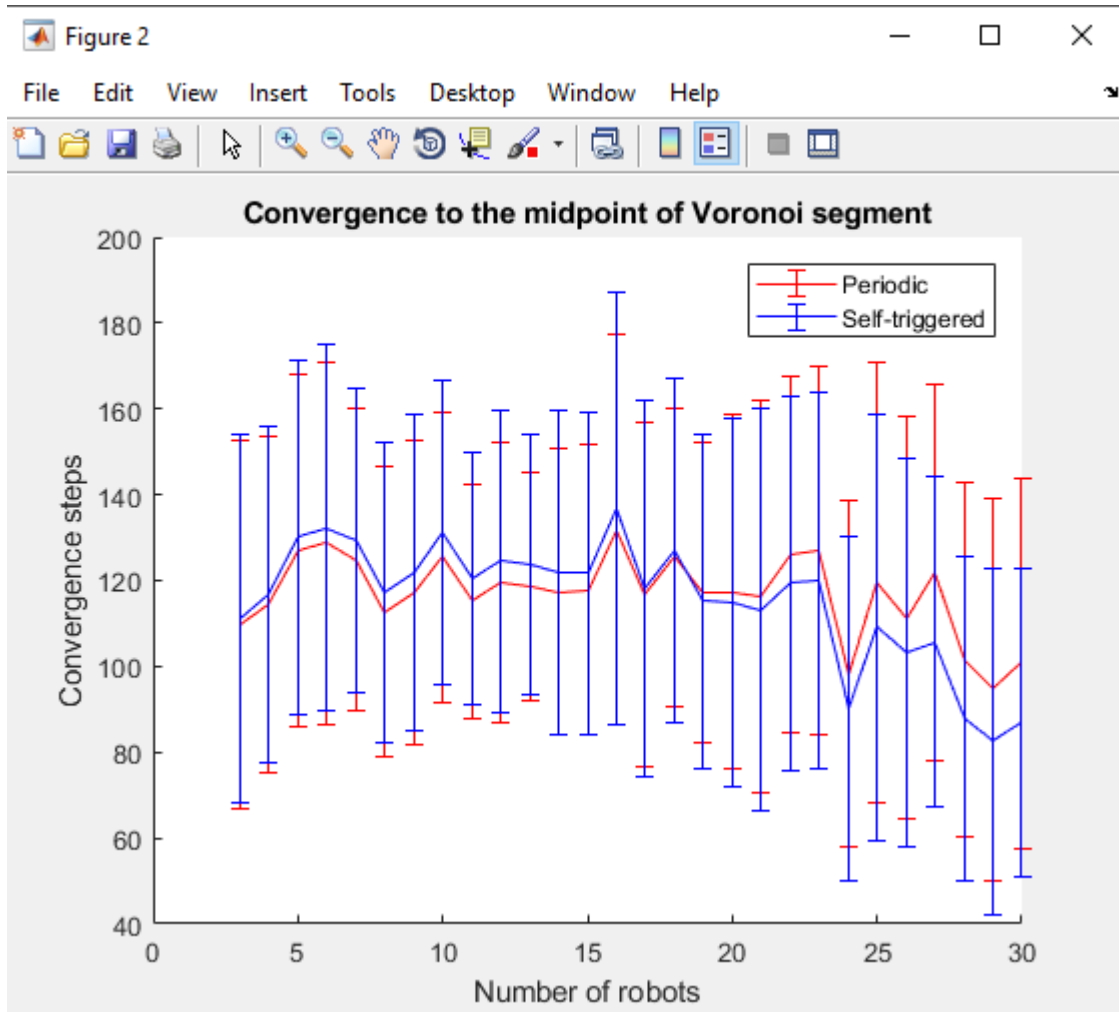


Fig 6.5: Comparison of Convergence time in constant and self-triggered strategies.

Now, we investigate the change in control input value with respect to the communication strategy implemented. Note that the control input is given individually with respect to each robot, N .

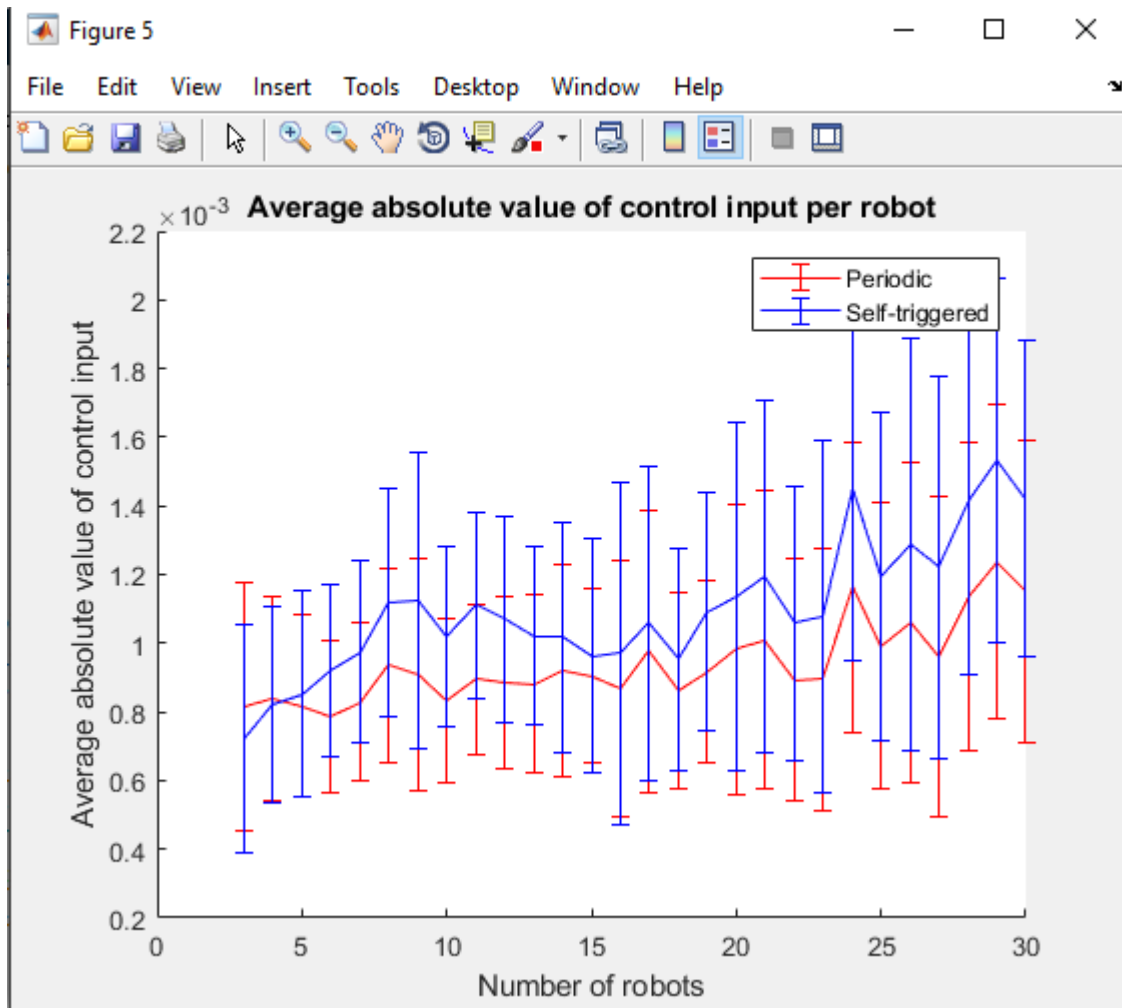


Fig 6.6: Change in average control value input to the robot in periodic and self-triggering strategies.

Now, looking into the employed self-triggering algorithm implementation with centralized and decentralized strategies.

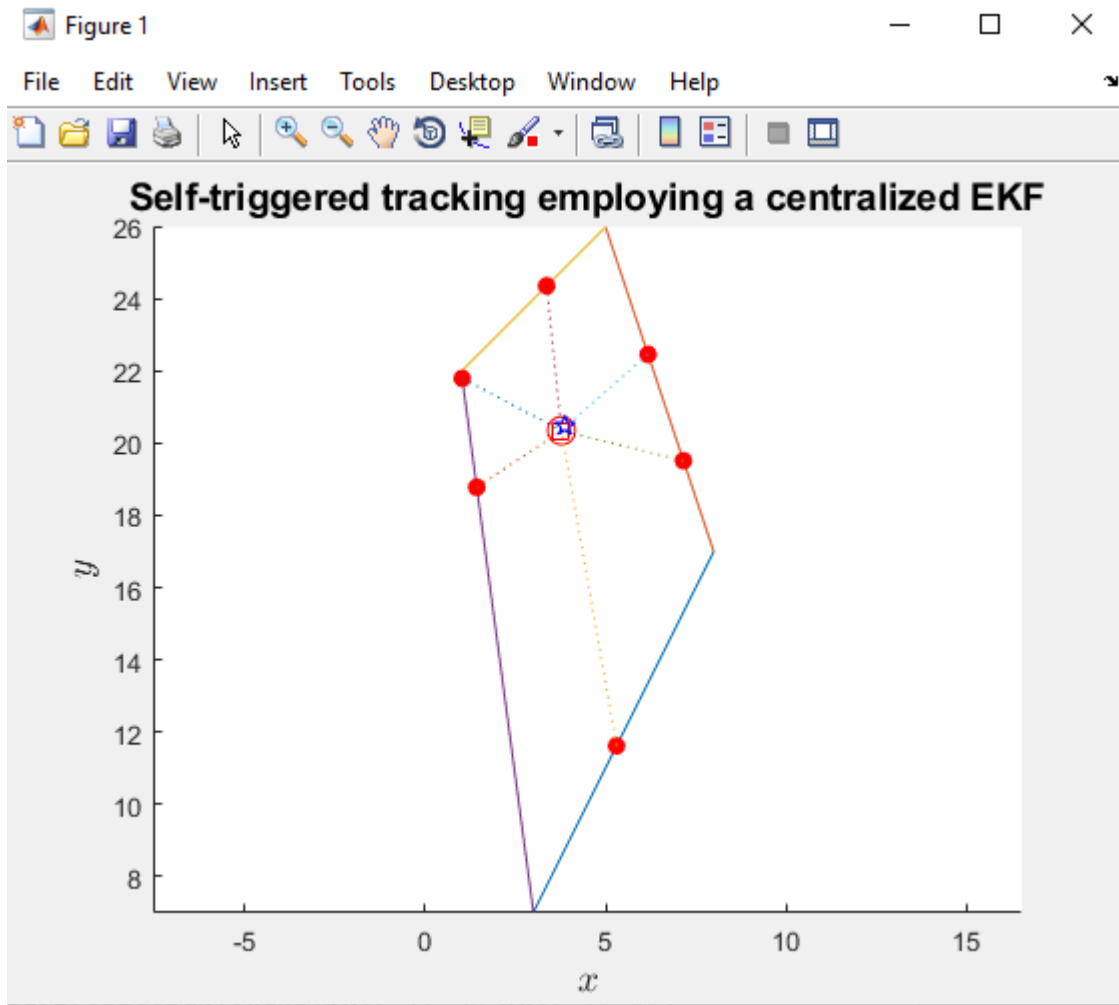


Fig 6.7: Self-triggering tracking strategy with centralized EKF with respect to convex boundary

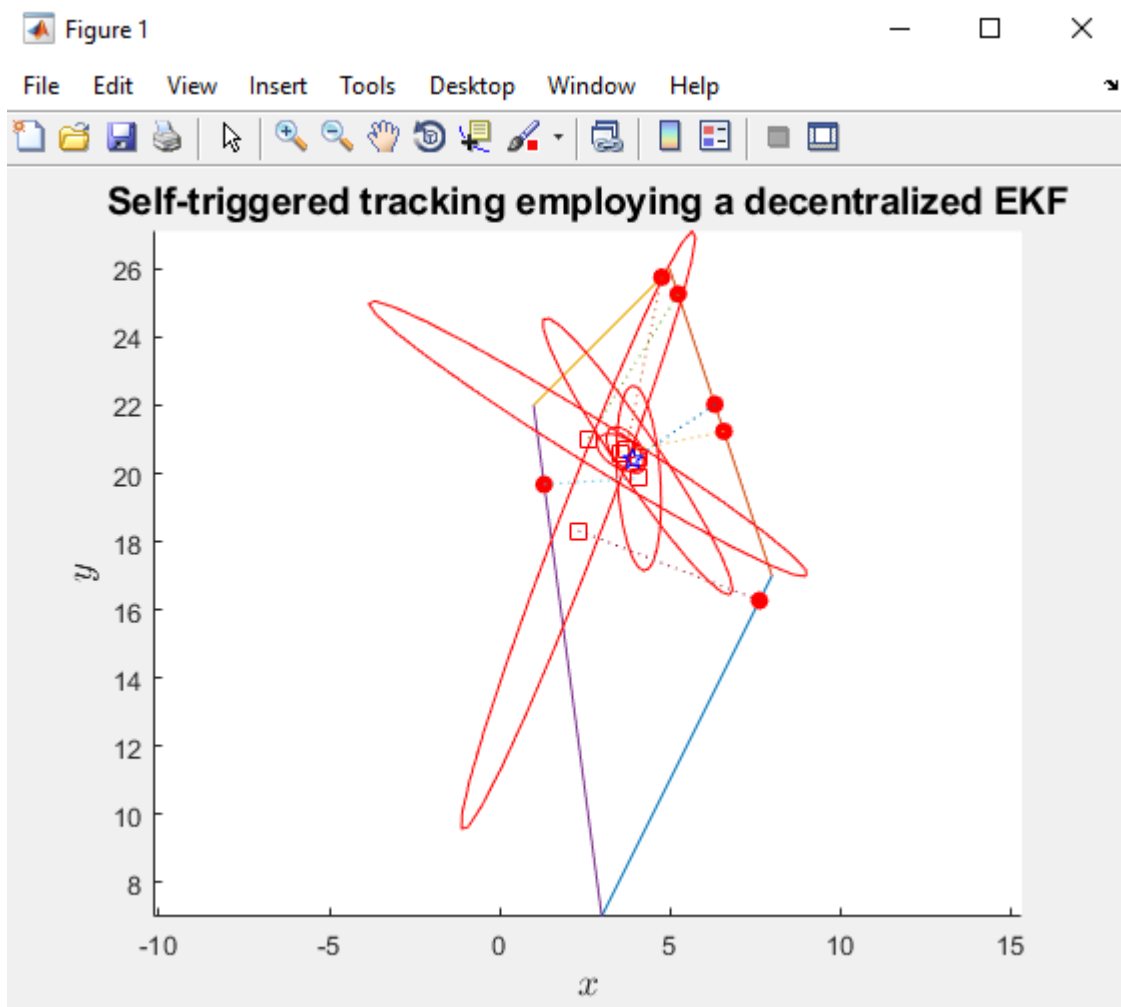


Fig 6.8: Self-triggering tracking strategy with decentralized EKF with respect to convex boundary

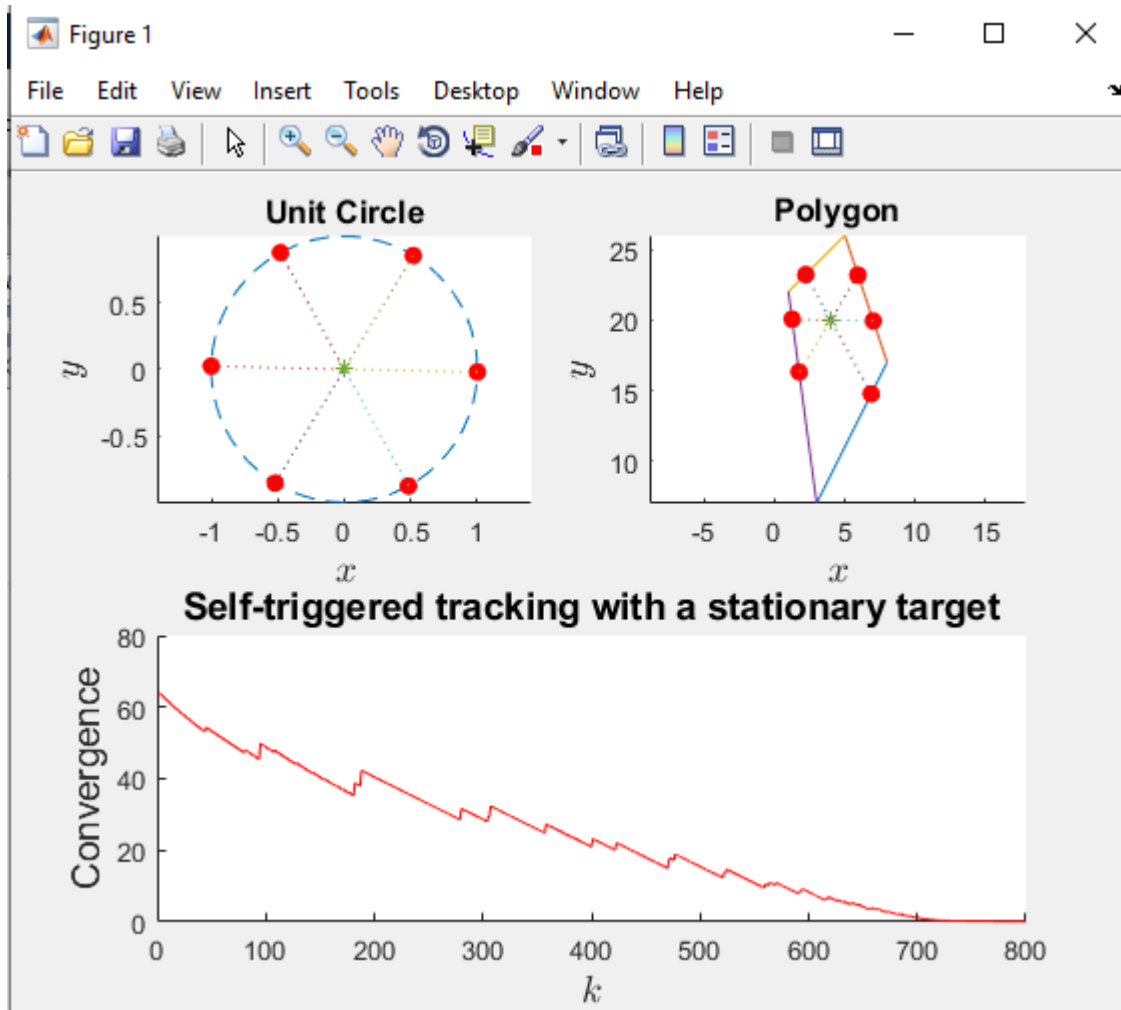


Fig 6.9 shows snapshot of the active tracking process under the self-triggering communication strategy starting with the initial configuration at time step $k = 1$ and ending in a uniform configuration around the target at $k = 760$

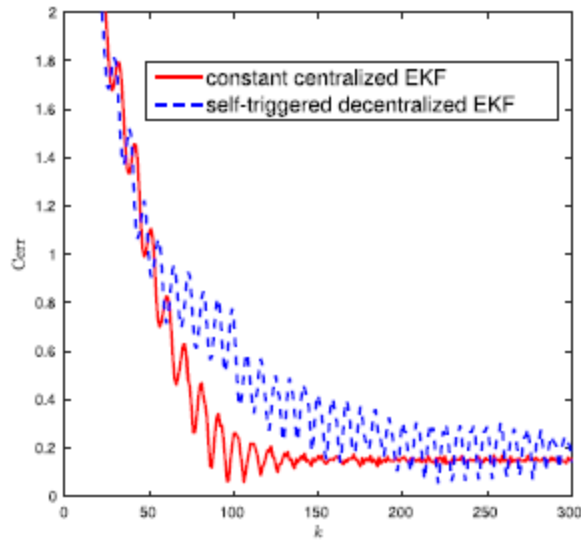


Fig 6.10: Comparison of C_{err} for mobile target tracking with constant centralized EKF and self-triggered decentralized EKF with respect to the radius and velocities (v and ω) of the moving target for $r = (0.6 \text{ m/s})/(0.6 \text{ rad/s}) = 1 \text{ m}$.

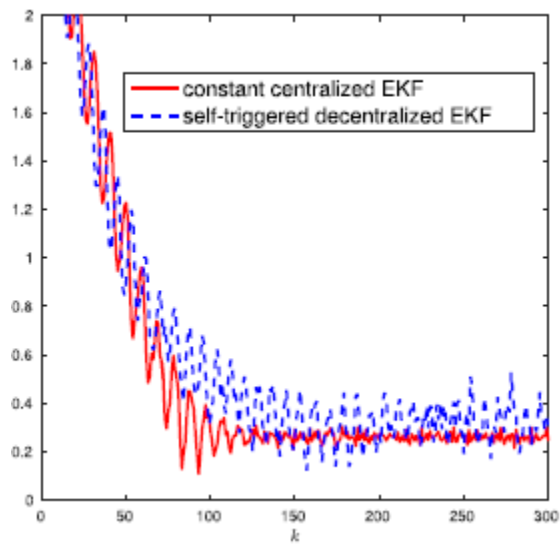


Fig 6.11: Comparison of C_{err} for mobile target tracking with constant centralized EKF and self-triggered decentralized EKF with respect to the radius and velocities (v and ω) of the moving target for $r = (1.0 \text{ m/s})/(0.6 \text{ rad/s}) = 1.67 \text{ m}$.

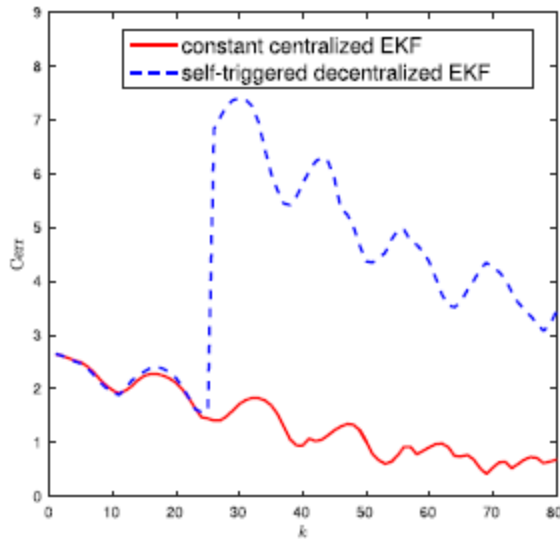


Fig 6.12: Comparison of C_{err} for mobile target tracking with constant centralized EKF and self-triggered decentralized EKF with respect to the radius and velocities (v and ω) of the moving target for $r = (1.0 \text{ m/s})/(0.4 \text{ rad/s}) = 2.5 \text{ m}$.

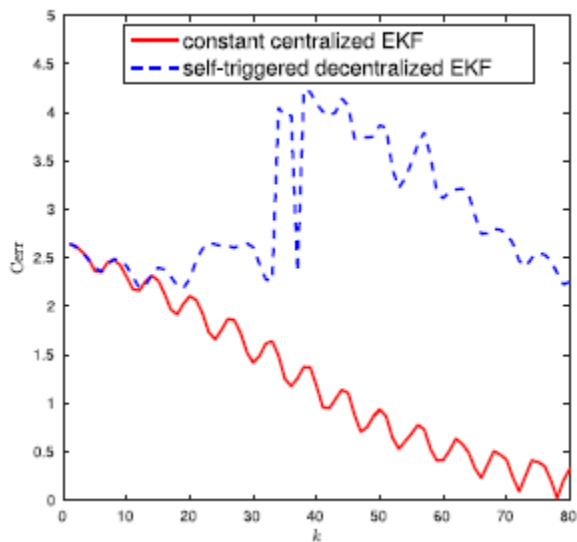


Fig 6.13: Comparison of C_{err} for mobile target tracking with constant centralized EKF and self-triggered decentralized EKF with respect to the radius and velocities (v and ω) of the moving target for $r = (1.0 \text{ m/s})/(0.8 \text{ rad/s}) = 1.25 \text{ m}$

The tracking performance of the self-triggered communication decentralized EKF strategy in comparison to constant periodic communication centralized EKF strategy as a function of the linear and angular velocities of the target's motion. We observe that the performance of the self-triggered strategy is comparable to the constant periodic algorithm, except when the target moves in a large circle [see Fig. 6.12] and when the target moves too fast [see Fig. 6.13].

Section 7: Conclusions and Discussions

We first formulated a self-triggered strategy such that robots can communicate with each other when specific conditions don't satisfy, unlike the constant control law which needs constant communication with its neighbors, which leads to asymptotic convergence to ensure a uniform configuration of the robots on the unit circle T , when the target is stationary. We see that different communications are triggered at different time steps. We then move on to use the above strategy to then experiment with moving targets which involve the Kalman filter due to noisy distributions. Here we experiment with centralized as well as distributed Kalman Filter Strategies and evaluate their performance based on this self-triggered strategy.

References

- [1] S. Martínez and F. Bullo, “Optimal sensor placement and motion coordination for target tracking,” *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.
- [2] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *Proc. IEEE 51st Annu. Conf. Decis. Control (CDC)*, Dec. 2012, pp. 3270–3285.
- [3] C. Robin and S. Lacroix, “Multi-robot target detection and tracking: Taxonomy and survey,” *Auton. Robots*, vol. 40, no. 4, pp. 729–760, 2016.
- [4] A. Khan, B. Rinner, and A. Cavallaro, “Cooperative robots to observe moving targets,” *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 187–198, Jan. 2018.
- [5] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. S. Sukhatme, “Cooperative multi-robot control for target tracking with onboard sensing,” *Int. J. Robot. Res.*, vol. 34, no. 13, pp. 1660–1677, 2015.
- [6] A. Dias, J. Capitan, L. Merino, J. Almeida, P. Lima, and E. Silva, “Decentralized target tracking based on multi-robot cooperative triangulation,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 3449–3455.
- [7] A. Franchi, P. Stegagno, and G. Oriolo, “Decentralized multi-robot encirclement of a 3D target with guaranteed collision avoidance,” *Auton. Robots*, vol. 40, no. 2, pp. 245–265, 2016.
- [8] Y. Sung, A. K. Budhiraja, R. K. Williams, and P. Tokekar. (2017). “Distributed simultaneous action and target assignment for multi-robot multitarget tracking.” [Online]. Available: <https://arxiv.org/abs/1706.02245>
- [9] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.

- [10] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, May 2012.
- [11] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, no. 6, pp. 1077–1087, 2012.
- [12] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2004.
- [13] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *Int. J. Robot. Res.*, vol. 12, no. 1, pp. 20–44, 1993.

Appendix

Active Target Tracking With Self-Triggered Communications in Multi-Robot Teams

Lifeng Zhou¹, *Student Member, IEEE*, and Pratap Tokekar², *Member, IEEE*

Abstract—We study the problem of reducing the amount of communication in decentralized target tracking. We focus on the scenario, where a team of robots is 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, which guarantees 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 decentralized strategies to reduce 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. We prove that this strategy converges asymptotically to the desired formation when the target is stationary. For the case of a mobile target, we use a decentralized Kalman filter with covariance intersection to share the beliefs of neighboring robots. We evaluate all the approaches through simulations and a proof-of-concept experiment.

Note to Practitioners—We study the problem of tracking a target using a team of coordinating robots. Target tracking problems are prevalent in a number of applications, such as co-robots, surveillance, and wildlife monitoring. Coordination between robots typically requires communication amongst them. Most multi-robot coordination algorithms implicitly assume that the robots can communicate at all time steps. Communication can be a considerable source of energy consumption, especially for small robots. Furthermore, communicating at all time steps may be redundant in many settings. With this as motivation, we propose an algorithm where the robots do not necessarily communicate at all times and instead choose specific triggering time instances to share information with their neighbors. Despite the limitation of limited communication, we show that the algorithm converges to the optimal configuration both in theory as well as in simulations.

Index Terms—Multi-robot systems, networked control, target tracking.

Manuscript received February 23, 2018; accepted August 22, 2018. Date of publication September 14, 2018; date of current version July 1, 2019. This work was supported by the National Science Foundation under Grant 1566247 and Grant 1637915. This paper was recommended for publication by Associate Editor D. Dimarogonas and Editor K. Saitou upon evaluation of the reviewers' comments. (*Corresponding author: Lifeng Zhou.*)

The authors are with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060 USA (e-mail: lfzhou@vt.edu; tokekar@vt.edu).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. It contains a video that illustrates six robots moving on the boundary of a convex polygon to track a target inside the polygon by using a self-triggered tracking strategy. The video shows the active target tracking in Matlab simulation, Gazebo experiment, and real-world experiment. The total file size is 72.3 MB.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TASE.2018.2867189

I. INTRODUCTION

TARGET tracking is one of the more 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]. We study *active* target tracking with a team of robots, where the focus is on actively controlling the state of the robot. The robots can exchange information with each other and then decide how to move so as to best track the target. It is typically assumed that exchanging information is beneficial. It is typical to design strategies by assuming that the robots will exchange their information at each time step irrespective of whether that information is worth exchanging. In this paper, we investigate the problem of deciding when it is worthwhile for the robots to exchange information and when it is okay to use possibly outdated information.

The motivation for this paper stems from the observation that communication can be costly. For example, for smaller robots, radio communication can be a significant source of power consumption. The robots can extend their lifetime by reducing the time spent communicating (equivalently, number of messages sent). Our goal is thus to determine a strategy that communicates only when required without considerably affecting the tracking performance.

We study this problem in a simple target tracking scenario first introduced by Martínez and Bullo [9]. Here, the robots are restricted to move on the boundary of a convex environment. They can obtain distance measurements toward a target moving in the interior. The goal of the robots is to position themselves so as to maximize the information gained from the target. Our problem setup models scenarios where the robots cannot enter into the interior of the environment. For example, Pierson *et al.* [10] studied pursuit-evasion, where the pursuers are not allowed to enter “no-fly zones”. If the evader enters a “no-fly zone”, then the pursuers reposition themselves on the perimeter of a convex approximation of the zone. Another motivating application is that of tracking radio-tagged fish [8] using ground robots that can move only along the boundary of the environment.

Martínez and Bullo [9] proposed a decentralized strategy where the robots communicate at all time steps with their neighbors and proved that it converges to the optimal (uniform) configuration. Instead, we apply a self-triggered coordination algorithm (following recent works [11], [12]) 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 relative to the constant strategy in simulations.

Next, we study the problem where the robots obtain noisy measurements of the distance to the targets. In a decentralized setting, robots can exchange information only with their neighbors. As a result, their local estimates of the target's position may differ considerably, resulting in poor tracking, especially when the robots are not in a uniform configuration. We show how to use a decentralized Kalman filter estimator that fuses the beliefs shared by neighboring robots (at triggered instances) to a common estimate.

Our main results assume that the robots have sufficiently large communication and sensing ranges. In Section V-B, we present necessary conditions on the sensing and communication ranges for our results to hold. We also present sufficient conditions for a modified version of the self-triggered strategy to guarantee convergence.

Simulation results validate the theoretical analysis, showing that the self-triggered strategy converges to the optimal, uniform configuration. The average number of communication is less than 30% that of the constant strategy. We also demonstrate the performance of the algorithm through proof-of-concept experiments with five simulated and two actual robots coordinating with each other.

The rest of this paper is organized as follows. We start by surveying the related works in Section II. We formalize the problem in Section III. The self-triggered tracking strategy is presented in Section IV, assuming that the target's position is known and is fixed. We relax these assumptions and present two practical extensions in Section V for noisy measurements and limited sensing and communication ranges. The simulation results are presented in Section VI. We conclude with a discussion of future work in Section VII.

A preliminary version of this paper was first presented in [13] without the decentralized Kalman filter with covariance intersection and the analysis for the limited communication and sensing ranges (see Section V), the Gazebo simulation experiments, and the proof-of-concept experiment (see Section VI).

II. RELATED WORK

Multi-robot target tracking has been widely studied in robotics [14], [15]. Robin and Lacroix [14] surveyed multi-robot target detection and tracking systems and presented a taxonomy of relevant works. Khan *et al.* [15] classified and discussed control techniques for multi-robot multi-target monitoring and identified the major elements of this problem. Hausman *et al.* [16] proposed a centralized cooperative approach for a team of robots to estimate a moving target. They showed how to use onboard sensing with limited sensing range and switch the sensor topology for effective target tracking. Dias *et al.* [17] proposed a multi-robot triangulation method to deal with initialization and data association issues in bearing-only sensors. The robot communicates locally to exchange and update the estimate beliefs of the target by a decentralized filter. Franchi *et al.* [18] presented a decentralized strategy to ensure that the robots follow the target while moving around it in a circle. They assume that the robots are labeled. Similar to this paper, the robots attempt to maintain a uniform distribution on a (moving) circle around

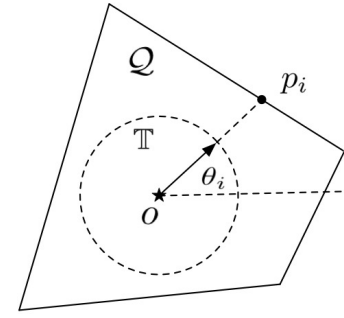


Fig. 1. Mapping from convex boundary ∂Q to unit circle \mathbb{T} .

the target. However, unlike this paper, they require that the robots constantly communicate with their local neighbors.

Sung *et al.* [19] proposed a distributed approach for multi-robot assignment problem for multi-target tracking by taking both sensing and communication ranges into account. The goal of their work is also to limit the communication between the robots. However, they do so by limiting the number of messages sent at each time step but allow the robots to communicate at all time steps. Instead, this paper explicitly determines when to trigger communication with other robots.

This paper builds on event-triggered and self-triggered communication schemes studied primarily by the controls community [12], [20]. Dimarogonas *et al.* [21] presented both centralized and decentralized event-triggered strategies for the agreement problem in multi-agent systems. They extended the results to a self-triggered communication setting where the robot calculates its next communication time based on the previous one without monitoring the state error. Nowzari and Cortés [11] proposed a decentralized self-triggered coordination algorithm for the optimal deployment of a group of robots based on spatial partitioning techniques. The synchronous version of this algorithm converges comparatively with an all-time communication strategy.

To the best of our knowledge, our paper is the first to simultaneously handle both robot coordination [18] and target tracking [17]. We focus on applying self-triggered control to reduce the amount of local communication between neighbors.

III. PROBLEM FORMULATION

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

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

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

We assume that all robots follow simple first-order continuous-time motion model. Each robot i knows its own position exactly at all times. When two robots communicate,

they can exchange their respective positions. We also assume that all robots have sensors that cover the environment and can always communicate with their neighbors (i.e., robot i can communicate with $i + 1$ and $i - 1$). In Section V-B, we derive necessary and sufficient conditions of the sensing and communication range.

Let ω_{\max} denote the common maximum angular speed¹ for all robots on the unit circle. Our results can be extended to the situation where each robot has its own maximum angular speed.

Martínez and Bullo [9] showed that the optimal configuration for the robots that can obtain distance measurements toward the target is a uniform configuration along the circle where each pair of neighboring robots is equally spaced around the target. That is, $\theta_{i+1} - \theta_i = 2\pi/N, \forall i \in \{1, \dots, N\}$. 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. Martínez 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 only two of its immediate neighbors. That is, a robot i can communicate with only $i - 1$ and $i + 1$ along the circle. The analysis requires that all robots know the position of the target exactly and that the target remains stationary. Martínez and Bullo [9] showed how to apply the same control law in situations where the target's position is not known exactly and is instead estimated by combining noisy range measurements in an extended Kalman filter (EKF). They also evaluated the performance of the algorithm empirically in cases where the target is allowed to move.

The control law in [9] assumes that the neighboring robots communicate at every time step. We call this the *constant strategy*. Our objective in this paper is to reduce the number of communications between the robots while still maintaining 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 it should 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 model and assumptions described in this section.

IV. SELF-TRIGGERED TRACKING ALGORITHM

In this section, we present the self-triggered tracking algorithm for achieving a uniform configuration 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 Section V.

Our algorithm builds on the self-triggered centroid algorithm [11] that is a decentralized control law that achieves optimal deployment (i.e., uniform Voronoi partitions) in a convex environment. We suitably

¹Strictly 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 ω_{\max} .

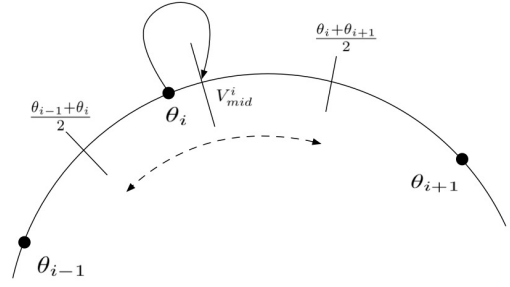


Fig. 2. Robot i goes toward the midpoint of its Voronoi segment via exact information from its neighbors.

modify this algorithm for the cases where the robots are restricted to move only on the boundary, $\partial\mathcal{Q}$, and can communicate with only two neighbors, as described in Section III. We first present the control law for each of the robots that use the motion prediction set of its neighbors based on their last known positions. Then, we present an update policy to decide when a robot should communicate and seek new information from its neighbors.

A. Control Law

The constant control law in [9] drives every robot toward the midpoint of its Voronoi segment. The Voronoi segment of the robot i is the part of the unit circle extending from $(\theta_{i-1} + \theta_i)/2$ to $(\theta_i + \theta_{i+1})/2$. The constant control law steers robot i toward the midpoint of its Voronoi segment V_{mid}^i by using real-time (at every time step)² information from its neighbors, θ_{i-1} and θ_{i+1} , as illustrated in Fig. 2. We refer to the book [22] for a comprehensive treatment on Voronoi segment.

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

$$\phi_j^i = \omega_{\max} \tau_j^i. \quad (2)$$

Thus, robot i can build a prediction motion set $\mathcal{R}_j^i(\theta_j^i, \phi_j^i)$ that contains all the possible locations where robot j could have moved to in τ_j^i time (see Fig. 3).

In our algorithm, it is sufficient for robot i to only communicate with its neighbors $i - 1$ and $i + 1$. The prediction motion range that robot i stores is given as $\mathcal{R}^i := \{\mathcal{R}_{i-1}^i(\theta_{i-1}^i, \phi_{i-1}^i), \mathcal{R}_{i+1}^i(\theta_{i+1}^i, \phi_{i+1}^i)\}$.

The 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 (see Fig. 4).

²Denote one time step as a small time interval, Δt .

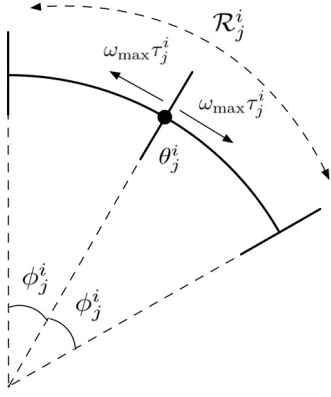


Fig. 3. Motion prediction set, \mathcal{R}_j^i , that each robot i maintains for its neighbors j . θ_j^i is the last known position of robot j , and τ_j^i is the time elapsed since this last known position.

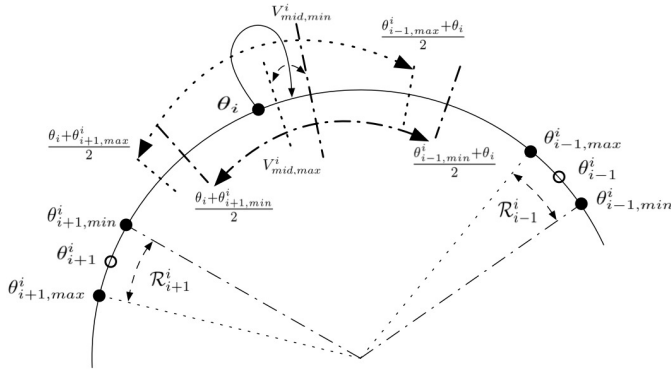


Fig. 4. Robot i goes toward the midpoint of its Voronoi segment via inexact motion prediction about its neighbors.

Define

$$\begin{aligned} \theta_{i-1,\min}^i &= (\theta_{i-1}^i - \phi_{i-1}^i) & \theta_{i-1,\max}^i &= (\theta_{i-1}^i + \phi_{i-1}^i) \\ \theta_{i+1,\min}^i &= (\theta_{i+1}^i - \phi_{i+1}^i) & \theta_{i+1,\max}^i &= (\theta_{i+1}^i + \phi_{i+1}^i). \end{aligned}$$

Thus, we have

$$\begin{aligned} \mathcal{R}_{i-1}^i(\theta_{i-1}^i, \phi_{i-1}^i) &= \{\beta \in \mathbb{T} \mid \theta_{i-1,\min}^i \leq \beta \leq \theta_{i-1,\max}^i\} \\ \mathcal{R}_{i+1}^i(\theta_{i+1}^i, \phi_{i+1}^i) &= \{\beta \in \mathbb{T} \mid \theta_{i+1,\min}^i \leq \beta \leq \theta_{i+1,\max}^i\}. \end{aligned}$$

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

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

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

The midpoint of its Voronoi segment $V_{\text{mid}}^i \in [V_{\text{mid},\min}^i, V_{\text{mid},\max}^i]$. That is

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

Substituting (3) and (4) into (5) yields

$$\frac{\theta_{i+1}^i + 2\theta_i + \theta_{i-1}^i - 2\omega_{\max}\tau^i}{4} \leq V_{\text{mid}}^i$$

and

$$V_{\text{mid}}^i \leq \frac{\theta_{i+1}^i + 2\theta_i + \theta_{i-1}^i + 2\omega_{\max}\tau^i}{4}$$

then

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

Therefore

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

Thus, the angular distance between V_{mid}^i and $(\theta_{i+1}^i + 2\theta_i + \theta_{i-1}^i)/4$ is bounded by $\omega_{\max}\tau^i/2$. In fact, the point $(\theta_{i+1}^i + 2\theta_i + \theta_{i-1}^i)/4$ indicates the midpoint of i 's guaranteed Voronoi segment gV_{mid}^i , defined as

$$gV_{\text{mid}}^i = \left\{ \beta \in \mathbb{T} \mid \max_{\theta_j \in \mathcal{S}_i} |\beta - \theta_i| \leq \min_{j \in \mathcal{S}_i} |\beta - \theta_j|, \forall j \neq i \right\}$$

where $T_1, \dots, T_n \subset \mathbb{T}$ are a set of connected segments in \mathbb{T} . We refer to the report [23] for more details on the guaranteed Voronoi segment. Thus, the guaranteed Voronoi segment of robot i can be computed as

$$gV_{\text{mid}}^i = \left\{ \beta \mid \frac{\theta_i + \theta_{i+1,\min}^i}{2} \leq \beta \leq \frac{\theta_{i-1,\max}^i + \theta_i}{2} \right\}. \quad (7)$$

Although robot i does not know the exact midpoint of its Voronoi segment V_{mid}^i , it can move toward the midpoint of its guaranteed Voronoi segment gV_{mid}^i instead, which is given by

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

In general, moving toward gV_{mid}^i does not guarantee that the robot moves closer to the midpoint of its Voronoi segment. However, the statement holds under the following condition.

Lemma 1: Suppose robot i moves from θ_i toward gV_{mid}^i . Let θ'_i be its position after one time step. If $|\theta'_i - gV_{\text{mid}}^i| \geq |V_{\text{mid}}^i - gV_{\text{mid}}^i|$, then $|\theta'_i - V_{\text{mid}}^i| \leq |\theta_i - V_{\text{mid}}^i|$.

The proof for Lemma 1 follows directly from [11, Proof of Lemma 5.1]. Consequently, as long as the robot can ensure that its new position θ'_i satisfies $|\theta'_i - gV_{\text{mid}}^i| \geq |V_{\text{mid}}^i - gV_{\text{mid}}^i|$, 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_{mid}^i . Instead, we can set an upper bound on this term using (6). We denote this upper bound by $\text{ubd}_i := \omega_{\max}\tau^i/2$. Thus, we get the following result.

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

$$|\theta'_i - gV_{\text{mid}}^i| > \text{ubd}_i \quad (9)$$

then $|\theta'_i - V_{\text{mid}}^i| \leq |\theta_i - V_{\text{mid}}^i|$.

Next, we present a motion control law that steers the robots toward 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 (8) and moves toward the midpoint until it is within ubd_i of it. Formally, the control, $u_i(t_k)$, for robot i at time t_k is given by

$$u_i(t_k) = \omega_i \text{ unit}(gV_{\text{mid}}^i - \theta_i) \quad (10)$$

where

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

B. Triggering Policy

As time elapses, without new information, the upper bound ubd_i grows larger until the condition in (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 (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. Otherwise, robot i might move close to gV_{mid}^i , 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:

$$\text{ubd}_i < \max \{ \|\theta' - gV_{\text{mid}}^i\|, \sigma \}. \quad (11)$$

Furthermore, the motion control law is designed under the assumption that the robot i and its two neighbors are located in the counter-clockwise order. That is, $\theta_{i+1} > \theta_i > \theta_{i-1}$. Since the robots are identical, it is clear that there is no advantage gained by changing the order of robots along the circle. In a constant strategy, since the robots always communicate, they know the real-time position of their neighbors and can thus avoid the order being swapped. In a self-triggered strategy, however, 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 condition:

$$\theta_{i+1}^i - \omega_{\max} \tau_{i+1}^i > \theta_i > \theta_{i-1}^i + \omega_{\max} \tau_{i-1}^i. \quad (12)$$

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 midpoint strategy is presented in Algorithm 1.

Algorithm 1 Self-Triggered Midpoint

```

1: while all robots have not converged:
2:   for each robot  $i \in \{1, \dots, N\}$  perform:
3:     increment  $\tau_{i-1}^i$  and  $\tau_{i+1}^i$  by  $\Delta t$ 
4:     compute  $\mathcal{R}^i$ ,  $gVs_i$ ,  $gV_{\text{mid}}^i$ , and  $\text{ubd}_i$ 
5:     if condition in (11) OR condition in (12) is violated:
6:       trigger communication with  $i+1$  and  $i-1$ 
7:       reset  $\tau_{i+1}^i$  and  $\tau_{i-1}^i$  to zero
8:       recompute  $\mathcal{R}^i$ ,  $gVs_i$ ,  $gV_{\text{mid}}^i$ , and  $\text{ubd}_i$ 
9:     end if
10:    compute and apply  $u_i$  as defined in (10)
11:   end for
12: 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 convergence follows directly from [11, Proof of Proposition 6.1] with suitable modifications. In the following, we sketch these modifications.

In [11], 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 [11] change to L_1 distances. Instead of moving to the midpoint of the guaranteed Voronoi segment, the robots in [11] move to the centroid of a guaranteed Voronoi region. Instead of communicating with the two clockwise and counter-clockwise neighbors, the robots in [11] communicate with all possible Voronoi neighbors. None of these changes affect the correctness of the proof. We add an extra condition that triggers communications to prevent robots from changing their order along \mathbb{T} . Since this condition only results in additional triggers, it can only help convergence. Finally, since there is a one-to-one and onto mappings between ∂Q and \mathbb{T} , convergence along \mathbb{T} implies convergence along ∂Q .

V. PRACTICAL EXTENSIONS

In this section, we present two practical extensions of our algorithm relaxing some of the assumptions given in Section III.

A. 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 configuration 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.

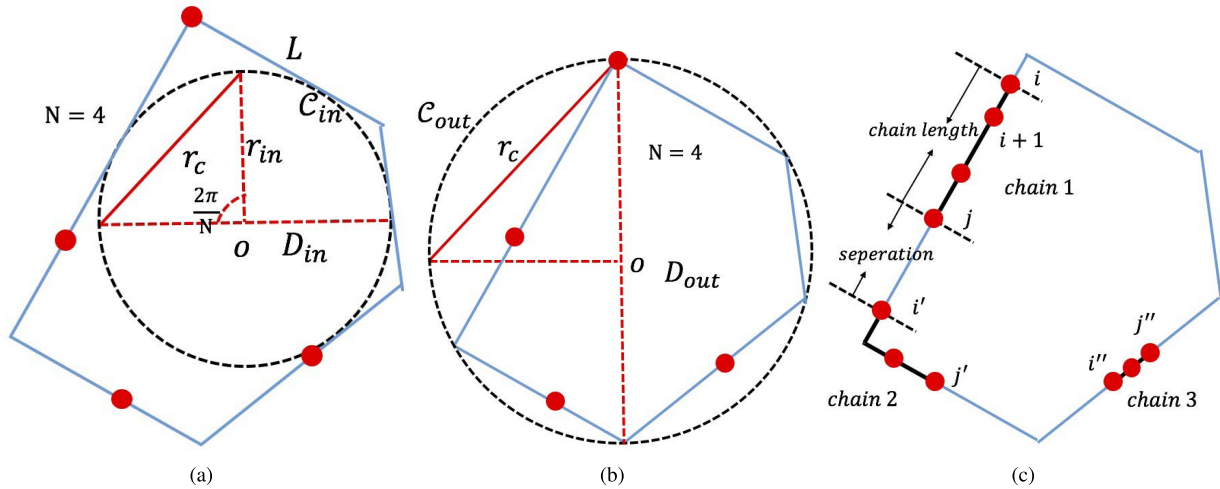


Fig. 5. Example of four robots moving on an arbitrary convex boundary to show (a) necessary, and (b), (c) sufficient conditions. Red solid circle: robot.

We use an 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 2-D Gaussian distribution parameterized by its mean, $\hat{o}(k)$, and covariance, $\hat{\Sigma}(k)$. The target's state prediction and update by EKF are described in the following.

Prediction:

$$\begin{aligned}\hat{o}^-(k) &= \hat{o}(k-1) \\ \hat{\Sigma}^-(k) &= \hat{\Sigma}(k-1) + R(k).\end{aligned}$$

Update:

$$\begin{aligned}K(k) &= \hat{\Sigma}^-(k)H^T(k)(H(k)\hat{\Sigma}^-(k)H^T(k) + Q(k))^{-1} \\ \hat{o}(k) &= \hat{o}^-(k) + K(k)(z(k) - h(\hat{o}^-(k))) \\ \hat{\Sigma}(k) &= (I - K(k)H(k))\hat{\Sigma}^-(k)\end{aligned}$$

where $R(k)$ and $Q(k)$ are the covariance matrices of the noise from target's motion model and robot's measurement, respectively, $h(\hat{o}^-(k)) := \|p(k) - \hat{o}^-(k)\|_2$, $z(k)$ denotes the noisy distance measurement from the robot, and $H(k)$ is the Jacobean of $h(\hat{o}^-(k))$. 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 (1).

In the centralized setting, a common fusion center can obtain the measurements from all the robots and compute a single target estimate, $\hat{o}(k)$, at every time step. Therefore, each robot will have the same estimated mean, $\hat{o}(k)$, and therefore the same center for the unit circle. However, in the decentralized case, 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. As a result, the centers of the unit circle will not be the same, making convergence challenging.

If at any time step, a robot communicates with its neighbors, then it can also share its estimate, mean $\hat{o}(k)$ and covariance $\hat{\Sigma}(k)$, with its neighbors. Therefore, at these triggered instances, each robot can update its own estimate by fusing the estimates from its neighbors. We use the covariance intersection algorithm, which is a standard decentralized EKF technique, to fuse estimates under unknown corrections [24].

The covariance intersection algorithm takes two Gaussian beliefs, (x_a, Σ_a) and (x_b, Σ_b) , and combines them into a common belief, (x_c, Σ_c)

$$\begin{aligned}x_c &= \Sigma_c((\Sigma_a)^{-1}x_a + (\Sigma_b)^{-1}x_b) \\ \Sigma_c &= (\lambda(\Sigma_a)^{-1} + (1-\lambda)(\Sigma_b)^{-1})^{-1}\end{aligned}$$

where $\lambda \in [0, 1]$ is a design parameter obtained by optimizing some criteria, *i.e.*, determinant or trace of Σ_c .

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

B. Limited Communication and Sensing Range

Our main result assumes that the robots have sufficiently large communication and sensing ranges. In this section, we first present a necessary condition for the communication range r_c and sensing range r_s . We then present a sufficient condition on the communication range for a modified version of our algorithm.

Theorem 1 (Necessary Condition): Let N be the total number of robots. To guarantee the convergence to the optimal configuration when the robots do not know N , the communication range r_c cannot be less than $D_{in} \sin(\pi/N)$ and the sensing range r_s cannot be less than $D_{in}/2$. D_{in} indicates the diameter of the largest radius circle contained completely inside the environment.

Proof:

Consider an arbitrary convex boundary as shown in Fig. 5(a). We draw its inscribed circle C_{in} with radius r_{in} and diameter D_{in} . To guarantee convergence without knowing N , the robots must be able to communicate with both neighbors when they reach a uniform configuration. When $N = 4$ [see Fig. 5(a)], if the communication range among any two robots $r_c < \sqrt{2}r_{in} = (\sqrt{2}D_{in}/2)$, these four robots cannot communicate with each other, even when they are at the uniform configuration. For any N , r_c can be

calculated by using the cosine law

$$r_c^2 = 2r_{in}^2 - 2r_{in}^2 \cos \frac{2\pi}{N}.$$

Thus

$$r_c = 2r_{in} \sin \frac{\pi}{N} = D_{in} \sin \frac{\pi}{N}.$$

Thus, we obtain the necessary condition that r_c cannot be less than $D_{in} \sin(\pi/N)$.

If $r_s < r_{in}$, no robot can sense the target when the target is at the center of the circle. Thus, the sensing range r_s cannot be less than $D_{in}/2$. \square

Next, we propose a sufficient condition on the communication and sensing ranges to ensure convergence to the uniform configuration. We need to make an additional assumption that each robot can uniquely identify its forward and backward neighbors. We also assume that the communication range is the same for all the robots and is known to all the robots. We present a modified version of our strategy that works with limited communication range.

Modified Self-Triggered Strategy: If robot i cannot communicate with either of its two neighbors, it does not move. If robot i can only communicate with one of its neighbors, it moves in the direction of the other neighbor with maximum velocity. A robot keeps moving unless its motion will cause it to lose communication with its neighbors. If robot i can communicate with both of its neighbors, it applies the proposed control law (10).

Theorem 2 (Sufficient Condition): If the communication range $r_c \geq L/N$ and the sensing range $r_s \geq D_{out}$, then the modified strategy converges to the optimal configuration. Here, L and D_{out} indicate the environment's perimeter and the length of the longest segment contained completely inside the environment, respectively.

Proof: We define a *communication chain* [see Fig. 5(c)] to be the maximal set of consecutive robots, $i, i+1, \dots, j$, such that i can communicate with $i+1$, $i+1$ can communicate with $i+2$, and so on until j . We now show that irrespective of the starting configuration, using the modified control law, all the N robots will form a single chain.

We define the length of a chain to be the distance along the boundary (in the direction that contains the chain) between the two extreme robots in a chain [see Fig. 5(c)]. We denote two extreme robots as the robots at the two endpoints of the chain.

Consider a chain of K robots. We show that the robots in this chain will keep moving unless the length is greater than or equal to $(L/N)(K-1)$ or the chain merges with another. Extreme robots in a chain have only one neighbor with which they can communicate. According to the strategy, these robots will continuously move (in a direction away from the chain) with maximum velocity. Other robots between the two extreme robots in the communication chain apply self-triggered control law (10) to go toward the midpoint of its two neighbors. Therefore, the length of the chain keeps increasing as long as the robots are moving. Unless the chain merges with another one, the robots will stop moving when the distance between all consecutive pairs of robots is r_c . Here, $r_c \geq L/N$.

If two consecutive robots are on the same environment edge, then the distance along the boundary between the robots is exactly equal to r_c . If the two robots are on different boundary edges, then the distance between the robots along the boundary will be greater than r_c (due to the convexity of the environment). Therefore, the length of the chain when all K robots stop moving will be greater than or equal to $(L/N)(K-1)$.

Next, we prove our claim that eventually all robots form a single chain, by contradiction. Denote the *separation* between two consecutive chains as the distance between the starting (ending) robot of one chain and the ending (starting) robot of another chain along the boundary of the environment [see Fig. 5(c)]. Suppose, for contradiction, that there exists $M > 1$ chains after all robots have stopped moving. Let K_1, \dots, K_M be the number of robots in M chains. $K_1 + \dots + K_M = N$.

The separation between any two consecutive chains is strictly greater than L/N . Furthermore, the length of any chain is greater than or equal to $(L/N)(K_i - 1)$. The perimeter of the environment must be equal to the length of all chains and the separation between all consecutive chains. Therefore, the perimeter must be strictly greater than

$$\frac{L}{N}(K_1 - 1) + \dots + \frac{L}{N}(K_M - 1) + M \frac{L}{N} = L.$$

This contradicts with the fact that the perimeter of the environment is exactly L . Thus, we prove all the robots eventually form a single chain.

Finally, once we ensure that robots form a single chain, then the convergence proof follows from the convergence of the self-triggered policy. \square

VI. SIMULATION AND OUTDOOR EXPERIMENT

In this section, we evaluate the performance of the proposed self-triggered tracking coordination algorithm. We first compare the convergence time for the self-triggered and constant communication strategies to achieve a uniform configuration on a convex boundary (see Section IV). Then, we demonstrate the performance of the self-triggered and constant strategies for moving targets.

A. Stationary Target Case

In this section, we 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 where the initial positions of the robots are drawn uniformly at random on the boundary. Our MATLAB implementation is also available online.³

Fig. 6 shows snapshots of the active tracking process under the proposed self-triggered strategy starting with the initial configuration at time step $k = 1$ in Fig. 6(a) and ending in a uniform configuration around the target at $k = 760$, as shown

³<https://github.com/raaslab/Self-triggered-mechanism>

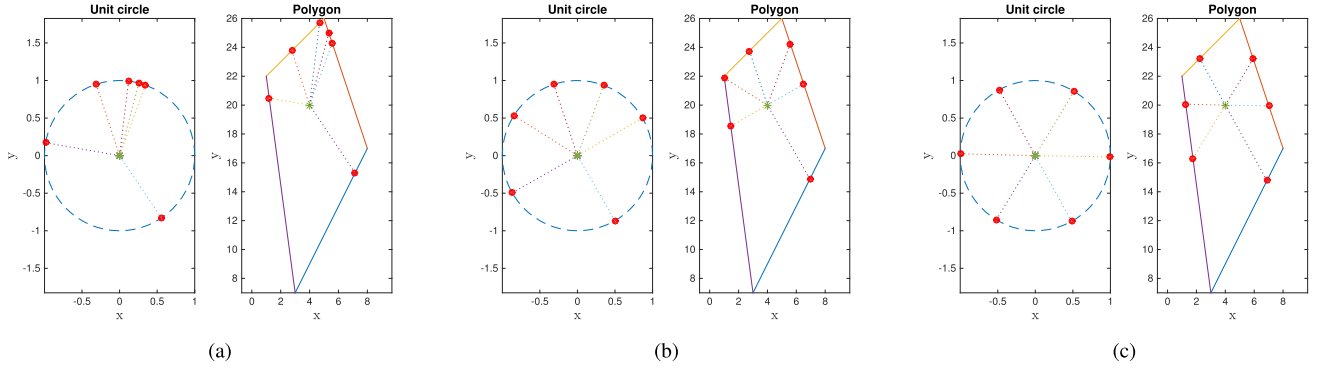


Fig. 6. Self-triggered tracking with six robots moving on the boundary of a convex polygon with a known, stationary target. The robots took 760 time steps to converge to the uniform configuration around the target. (a) $k = 1$. (b) $k = 400$. (c) $k = 760$.

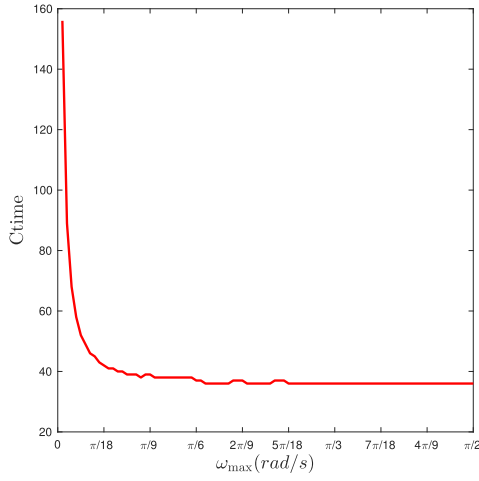


Fig. 7. Convergence time for six robots starting with same initial configuration for increasing values of maximum angular velocity ω_{\max} .

in Fig. 6(c). For this example, we assume that the robots know the position of the stationary target. At each time step, we use the map φ_o to find θ_i on the unit circle (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 $\omega_{\max} = \pi/180$ rad/s. In general, one can use the procedure given in the Appendix to compute ω_{\max} for a given environment. Note that the convergence time depends on ω_{\max} , which in turn depends on the shape of the environment assuming a fixed maximum linear velocity. In Fig. 7, we plot the convergence time for six robots starting from a fixed configuration by varying ω_{\max} from $\pi/180$ rad/s to $\pi/2$ rad/s. It shows that the convergence time approaches a limit with increasing ω_{\max} .

We first compare the convergence time of the two strategies with the same starting configurations for 30 trials [see Fig. 8(a)]. The convergence time, Ctime, is specified as the time step 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

$$\text{Cerr} = \sum_{i=1}^N |\theta_i - V_{\text{mid}}^i| \quad (13)$$

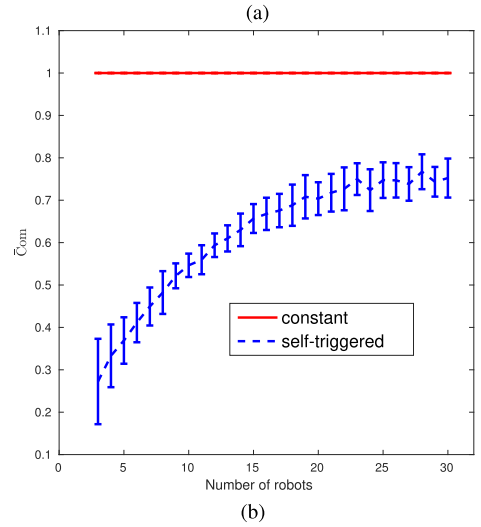
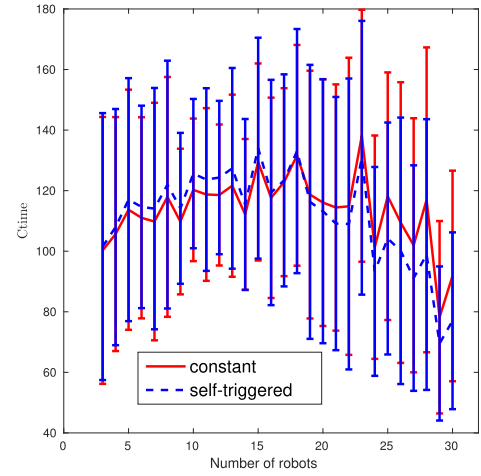


Fig. 8. Comparison of (a) convergence time and (b) number of communication messages in constant and self-triggered strategies with a stationary target at known position. The error bar indicates standard deviation.

in the constant communication case, and

$$\text{Cerr} = \sum_{i=1}^N |\theta_i - g V_{\text{mid}}^i| \quad (14)$$

in the self-triggered case.

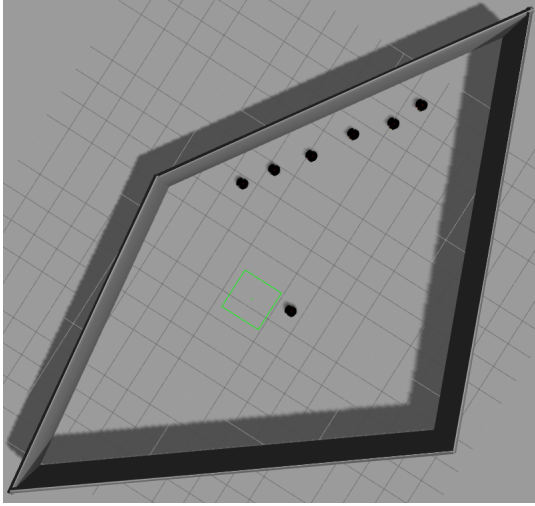
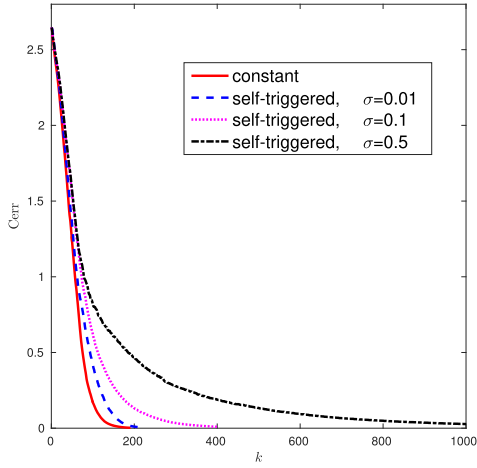
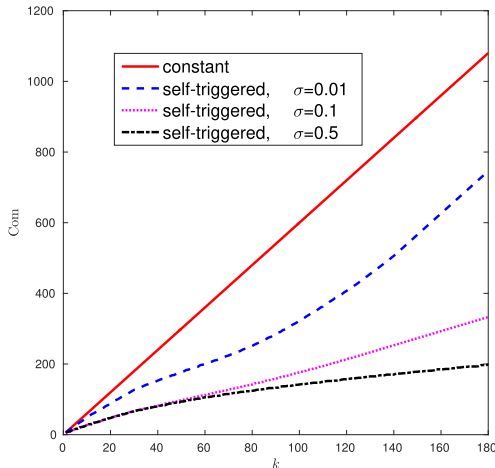


Fig. 9. Gazebo environment, where six simulated Pioneer 3DX robots are tasked to track a target moving in the interior.



(a)



(b)

Fig. 10. Comparison of (a) convergence error and (b) communication messages in constant and self-triggered communication strategies using the setup shown in Fig. 9.

The average number of communication messages is found as

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

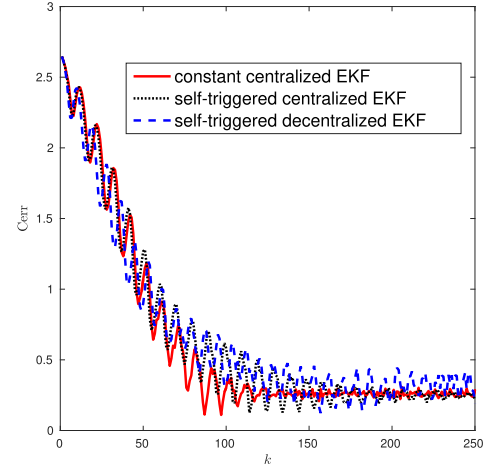


Fig. 11. Convergence error for mobile target tracking in constant communication with centralized EKF, self-triggered communication with centralized EKF, and self-triggered communication with decentralized EKF.

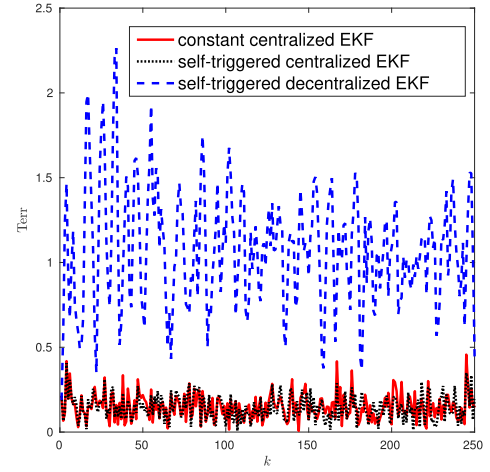


Fig. 12. Error in target's estimate for mobile target tracking in constant communication with centralized EKF, self-triggered communication with centralized EKF, and self-triggered communication with decentralized EKF.

where $com(i, Ctime)$ gives the total number of communications of a robot with its neighbors i at the end of $Ctime$. Fig. 8(b) shows the \overline{Com} in the self-triggered case. The number of communication messages in the constant communication case is a constant. Fig. 8(a) shows that the self-triggered mechanism converges comparatively with the constant strategy.

We also implemented our algorithm in Robot Operating System (ROS) and performed simulations in the Gazebo environment [25]. Fig. 9 shows an instance with six differential-drive Pioneer 3DX robots [26] that can move in forward and backward direction.

Fig. 10(a) shows that the constant communication strategy converges faster than the self-triggered one with six simulated robots. Changing the tolerance parameter σ affects the convergence time of the self-triggered strategy. The smaller the convergence tolerance σ , the faster the convergence, which comes at the expense of an increased number of messages. Fig. 10(b) shows communication messages for both strategies. The smaller the tolerance σ , the larger the number of messages. The convergence tolerance σ acts as a tradeoff between the communication messages and the convergence speed in the self-triggered case.

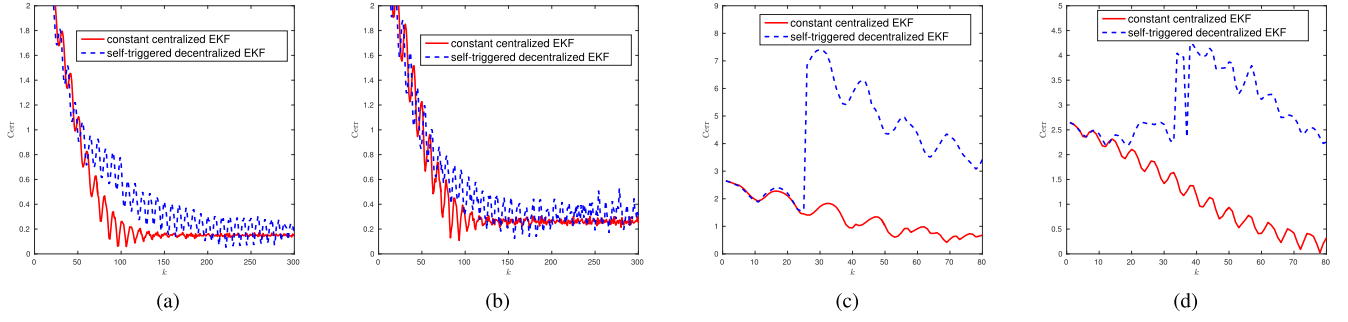
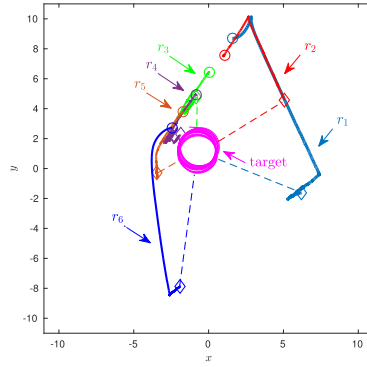


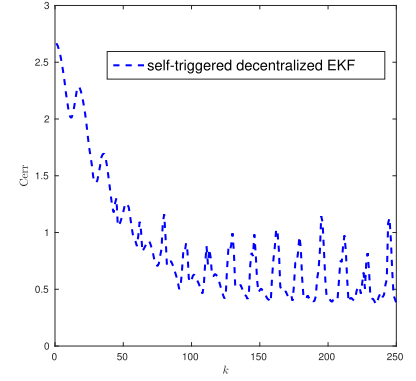
Fig. 13. Comparison of Cerr for mobile target tracking with constant centralized EKF and self-triggered decentralized EKF with respect to the radius and velocities (v and ω) of the moving target. (a) $r = (0.6 \text{ m/s})/(0.6 \text{ rad/s}) = 1 \text{ m}$. (b) $r = (1.0 \text{ m/s})/(0.6 \text{ rad/s}) \simeq 1.67 \text{ m}$. (c) $r = (1.0 \text{ m/s})/(0.4 \text{ rad/s}) = 2.5 \text{ m}$. (d) $r = (1.0 \text{ m/s})/(0.8 \text{ rad/s}) = 1.25 \text{ m}$.



(a)



(b)



(c)

Fig. 14. We use five simulated robots, r_1 – r_5 , and (a) two real robots, r_6 and the target, for the experiments. (b) Trajectories of all the robots and (c) the convergence error for self-triggered communication with decentralized EKF strategy.

B. Moving Target Case

Next, we present simulation results for the realistic case of mobile, uncertain target (see Section V). We evaluate three strategies: constant communication with centralized EKF, self-triggered communication with centralized EKF, and self-triggered communication with decentralized EKF. All three algorithms were implemented in Gazebo with six simulated Pioneer robots and a simulated Pioneer target moving on a circular trajectory. We assume that all the robots have the same maximum linear velocity, $v_{\max} = 0.2 \text{ m/s}$. We calculate the linear velocity for each robot i by $v_i = \omega_i \|p_i - \hat{o}\|_2$.

For a moving target with $v_o = 1.0 \text{ m/s}$ and $\omega_o = 0.6 \text{ rad/s}$, Fig. 11 shows that all three algorithms have similar tracking performance with respect to the convergence error, Cerr, over time. However, the target estimate error Terr is smaller in the centralized EKF cases than the decentralized case, as shown in Fig. 12. The target estimate error is defined as

$$\text{Terr} = \|\hat{o} - o\|$$

for the centralized case with \hat{o} indicating the centralized estimate of the target, and

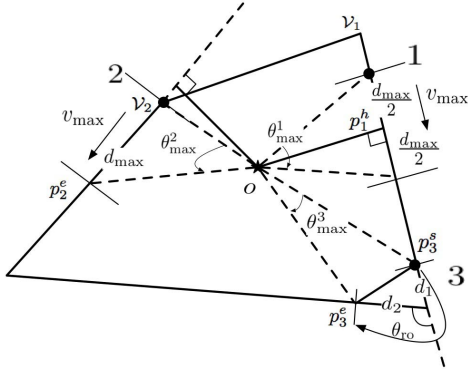
$$\text{Terr} = \frac{\sum_{i=1}^N \|\hat{o}_i - o\|}{N}$$

for the decentralized case with \hat{o}_i indicating the target estimate from each robot i .

Fig. 13 shows the tracking performance of the self-triggered communication decentralized EKF strategy in relation to the baseline constant communication centralized EKF strategy as a function of the linear and angular velocities of the target's motion. We observe that the performance of the self-triggered strategy is comparable to the baseline algorithm, except when the target moves in a large circle [see Fig. 13(c)] and when the target moves too fast [see Fig. 13(d)].

C. Proof-of-Concept Experiment

To further verify the tracking performance of the self-triggered decentralized EKF strategy, we also conducted a proof-of-concept mixed reality experiment. Due to limited resources, we used five simulated Pioneer 3DX robots ($r_1 \sim r_5$) cooperating with one real Pioneer 3DX robot (r_6) to track one real Pioneer 3DX target moving with $v_o = 1.0 \text{ m/s}$ and $\omega_o = 0.6 \text{ rad/s}$. The initial deployment for all seven robots is the same as Gazebo experiment (see Fig. 9). The two real Pioneer robots (robot 6 and target) and the trajectories of all robots during tracking are shown in Fig. 14(a) and (b). Fig. 14(c) shows the self-triggered communication decentralized EKF strategy achieves a comparable tracking performance

Fig. 15. Computing ω_{\max} .

with respect to the convergence error. The video showing all the simulations and experiments is available online.⁴

VII. CONCLUSION AND DISCUSSION

In this paper, we investigated 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 [9]. We applied a self-triggered coordination strategy that asymptotically converges to a uniform configuration around the target while reducing the number of communication to less than 30% of a constant strategy. We find that the self-triggered strategy performs comparably with the constant communication strategy. Future work 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. We conjecture that the latter question is crucial for better performance while tracking mobile targets. The self-triggered strategy can also be applied to other domains with networked controllers, e.g., for optimization of the networked industrial processes [27], [28].

APPENDIX

CALCULATION OF ω_{\max}

Assume the boundary of the convex environment ∂Q and the position (or its estimate) of the target are known. In addition, assume that the robot has a maximum speed v_{\max} with which it can move on ∂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 ω_{\max} into three separate cases (see Fig. 15).

In all cases, let \mathcal{E}^i be the edge on which the robot is located before moving a distance of d_{\max} . Let \mathcal{L}^i be the line supporting the edge. In cases 1 and 2, we compute ω_{\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 \mathcal{L}^i lies within \mathcal{E}^i .

$\omega_{\max}^{1, \mathcal{E}^i}$ 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. $\omega_{\max}^{1, \mathcal{E}^i}$ can be calculated as $\omega_{\max}^{1, \mathcal{E}^i} = \omega_{\max}^{1, \mathcal{E}^i} / \Delta t$ giving $\omega_{\max}^{1, \mathcal{E}^i} = \min_{\mathcal{E}^i \in \mathcal{E}} \{\omega_{\max}^{1, \mathcal{E}^i}\}$. Here, $\theta_{\max}^{1, \mathcal{E}^i}$ is the angle shown in Fig. 15. Since we assume ∂Q and the target's position (or its estimate) are known, we can calculate the length of perpendicular bisector $|p_1^h o|$. Then, $\theta_{\max}^{1, \mathcal{E}^i}$ can be computed by applying Pythagorean theorem for $|p_1^h o|$ and $d_{\max}/2$.

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

Similar to case 1, ω_{\max} can be computed as $\omega_{\max}^{2, \mathcal{E}^i} = \omega_{\max}^{2, \mathcal{E}^i} / \Delta t$, where $\omega_{\max}^{2, \mathcal{E}^i} = \min_{\mathcal{E}^i \in \mathcal{E}} \{\omega_{\max}^{2, \mathcal{E}^i}\}$. Here, $\theta_{\max}^{2, \mathcal{E}^i}$ is the 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. As one example in Fig. 15, robot starts from one vertex of the ∂Q , \mathcal{V}_2 , and travels d_{\max} distance until the ending point p_2^e . Since ∂Q is known, we know the position of its vertex \mathcal{V}_2 and can compute the position of ending point p_2^e by knowing $|\mathcal{V}_2 p_2^e| = d_{\max}$ and ∂Q . Then, we can compute $|p_2^e o|$ and $|\mathcal{V}_2 o|$. By using the law of cosines, we can compute $\theta_{\max}^{2, \mathcal{E}^i}$ and then obtain $\omega_{\max}^{2, \mathcal{E}^i}$.

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

We assume that within one time step Δt , the robot moves $d_1^{\mathcal{V}_i}$ on one edge and $d_2^{\mathcal{V}_i}$ on another edge. Since the robot must spend some time at the vertex turning in place, we have $d_1^{\mathcal{V}_i} + d_2^{\mathcal{V}_i} < d_{\max}$. We calculate d_2 by

$$\frac{(d_1 + d_2)}{v_{\max}} + \frac{\theta_{\text{ro}}^{\mathcal{V}_i}}{\omega_{\text{ro}}} = \Delta t$$

where $\theta_{\text{ro}}^{\mathcal{V}_i}$ and ω_{ro} denote the rotation angle at the vertex \mathcal{V}_i and rotational speed of the robot, which are known. Then, we show the calculation of $\theta_{\max}^{3, \mathcal{V}_i}$ by an example in Fig. 15, where robot starts from p_3^s , crosses the vertex by rotating θ_{ro} , and ends at p_3^e . Once we know d_1 , d_2 , and θ_{ro} , we can use the law of cosines to calculate $|p_3^s p_3^e|$. Then, by applying cosine law again to $|p_3^s p_3^e|$, $|p_3^s o|$, and $|p_3^e o|$, we can compute $\theta_{\max}^{3, \mathcal{V}_i}$. We use this procedure to calculate $\theta_{\max}^{3, \mathcal{V}_i}$ at the vertex \mathcal{V}_i . Thus, the $\omega_{\max}^{3, \mathcal{V}_i}$ can be calculated as

$$\omega_{\max}^{3, \mathcal{V}_i} = \frac{\theta_{\max}^{3, \mathcal{V}_i}}{\Delta t}.$$

Then, ω_{\max}^3 can be specified as

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

where $\mathcal{V}_i \in \mathcal{V}$ and $0 \leq d_1 \leq \Delta t - (\theta_{\text{ro}}^{\mathcal{V}_i} / \omega_{\text{ro}})$.

Finally, ω_{\max} can be computed as

$$\omega_{\max} = \min \{\omega_{\max}^1, \omega_{\max}^2, \omega_{\max}^3\}. \quad (15)$$

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

⁴<https://youtu.be/UcsRCc9cfns>

REFERENCES

- [1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2004.
- [2] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *Int. J. Robot. Res.*, vol. 12, no. 1, pp. 20–44, 1993.
- [3] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proc. IEEE Wireless Commun. Netw.*, vol. 3, Mar. 2003, pp. 1609–1614.
- [4] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 16–25, Sep. 2006.
- [5] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 3067–3072.
- [6] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *Int. J. Robot. Res.*, vol. 36, nos. 13–14, pp. 1540–1553, 2017.
- [7] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 24–39, Mar. 2012.
- [8] P. Tokekar, E. Branson, J. V. Hook, and V. Isler, "Tracking aquatic invaders: Autonomous robots for monitoring invasive fish," *IEEE Robot. Automat. Mag.*, vol. 20, no. 3, pp. 33–41, Sep. 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] A. Pierson, A. Ataei, I. C. Paschalidis, and M. Schwager, "Cooperative multi-quadrotor pursuit of an evader in an environment with no-fly zones," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 320–326.
- [11] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, no. 6, pp. 1077–1087, 2012.
- [12] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE 51st Annu. Conf. Decis. Control (CDC)*, Dec. 2012, pp. 3270–3285.
- [13] L. Zhou and P. Tokekar, "Active target tracking with self-triggered communications," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2017, pp. 2117–2123.
- [14] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: Taxonomy and survey," *Auton. Robots*, vol. 40, no. 4, pp. 729–760, 2016.
- [15] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 187–198, Jan. 2018.
- [16] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. S. Sukhatme, "Cooperative multi-robot control for target tracking with onboard sensing," *Int. J. Robot. Res.*, vol. 34, no. 13, pp. 1660–1677, 2015.
- [17] A. Dias, J. Capitan, L. Merino, J. Almeida, P. Lima, and E. Silva, "Decentralized target tracking based on multi-robot cooperative triangulation," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 3449–3455.
- [18] A. Franchi, P. Stegagno, and G. Oriolo, "Decentralized multi-robot encirclement of a 3D target with guaranteed collision avoidance," *Auton. Robots*, vol. 40, no. 2, pp. 245–265, 2016.
- [19] Y. Sung, A. K. Budhiraja, R. K. Williams, and P. Tokekar. (2017). "Distributed simultaneous action and target assignment for multi-robot multi-target tracking." [Online]. Available: <https://arxiv.org/abs/1706.02245>
- [20] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.
- [21] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, May 2012.
- [22] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, vol. 501. Hoboken, NJ, USA: Wiley, 2009.
- [23] W. Evans and J. Sember, "Guaranteed Voronoi diagrams of uncertain sites," in *Proc. 20th Can. Conf. Comput. Geometry*, 2008, pp. 207–210.
- [24] M. Reinhardt, B. Noack, and U. D. Hanebeck, "Closed-form optimization of covariance intersection for low-dimensional matrices," in *Proc. 15th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2012, pp. 1891–1896.
- [25] N. P. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Sep. 2004, pp. 2149–2154.
- [26] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," in *Proc. 12th Int. Conf. Adv. Robot. (ICAR)*, Jul. 2005, pp. 492–497.
- [27] T. Wang, J. Qiu, H. Gao, and C. Wang, "Network-based fuzzy control for nonlinear industrial processes with predictive compensation strategy," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 8, pp. 2137–2147, Aug. 2017.
- [28] T. Wang, H. Gao, and J. Qiu, "A combined fault-tolerant and predictive control for network-based industrial processes," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2529–2536, Apr. 2016.



Lifeng Zhou (S'17) received the B.S. degree in automation from the Huazhong University of Science and Technology, Wuhan, China, in 2013, and the M.Sc. degree in automation from Shanghai Jiao Tong University, Shanghai, China, in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Virginia Tech, Blacksburg, VA, USA.

His current research interests include multi-robot coordination, event-based control, sensor assignment, and risk-averse decision making.



Pratap Tokekar (M'10) received the B.Tech. degree in electronics and telecommunication from the College of Engineering Pune, Pune, India, in 2008, and the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, MN, USA, in 2014.

He was a Post-Doctoral Researcher with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA, USA. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. His current research interests include algorithmic and field robotics and applications to precision agriculture and environmental monitoring.

Dr. Tokekar received the National Science Foundation Computer and Information Science and Engineering Research Initiation Initiative Award.