



Nathaniel R. Olsen

Dynamic Autonomous Robotics (DARC) Lab,
Robotics Center and
Department of Mechanical Engineering,
University of Utah,
Salt Lake City, UT 84112

Sasha M. McKee

Dynamic Autonomous Robotics (DARC) Lab,
Robotics Center and
Department of Mechanical Engineering,
University of Utah,
Salt Lake City, UT 84112

Osama S. Haddadin

L3-Harris Technologies,
Salt Lake City, UT 84116

Scott M. Lyon

L3-Harris Technologies,
Salt Lake City, UT 84116

Jared E. Campbell

L3-Harris Technologies,
Salt Lake City, UT 84116

Kam K. Leang¹

Fellow ASME
Dynamic Autonomous Robotics (DARC) Lab,
Robotics Center and
Department of Mechanical Engineering,
University of Utah,
Salt Lake City, UT 84112
e-mail: kam.k.leang@utah.edu

Information-Theoretic Bayesian Inference for Multi-Agent Localization and Tracking of an RF Target with Unknown Waveform

Information-theoretic motion planning and machine learning through Bayesian inference are exploited to localize and track a dynamic radio frequency (RF) emitter with unknown waveform (uncooperative target). A target-state estimator handles non-Gaussian distributions, while mutual information is utilized to coordinate the motion control of a network of mobile sensors (agents) to minimize measurement uncertainty. The mutual information is computed for pairs of sensors through a four-permutation-with-replacement process. The information surfaces are combined to create a composite map, which is then used by agents to plan their motion for more efficient and effective target estimation and tracking. Simulations and physical experiments involving micro-aerial vehicles with time difference of arrival (TDOA) measurements are performed to evaluate the performance of the algorithm. Results show that when two or three agents are used, the algorithm outperforms state-of-the-art methods. Results also show that for four or more agents, the performance is as competitive as an idealized static sensor network.

[DOI: 10.1115/1.4065592]

Keywords: Bayesian estimation, mutual information, source localization, target tracking

1 Introduction

The target localization and tracking problem involves estimating the current and future states of a target using measurement data and a measurement model [1,2]. Applications include space exploration [3], transportation [4], military-threat monitoring [5] (see Fig. 1), search and rescue [6], and finding gas leaks [7,8]. Recently, increased attention on the use of unmanned autonomous systems, such as unmanned aerial vehicles, has drawn more interest in the target localization and tracking process [9].

This paper develops and evaluates the performance of an information-based technique to coordinate a network of mobile sensors (agents) to efficiently localize and track an uncooperative, dynamic target. Utilizing machine learning through Bayesian inference in the form of a particle filter and leveraging state

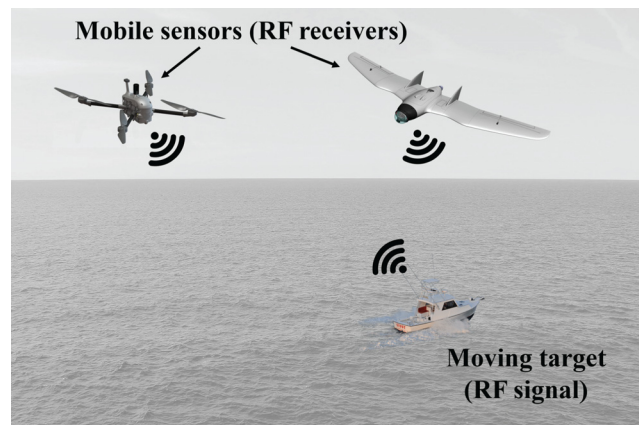


Fig. 1 Localization and tracking of an uncooperative target using RF receivers

¹Corresponding author.

Contributed by the Dynamic Systems Division of ASME for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received September 12, 2023; final manuscript received March 19, 2024; published online July 25, 2024. Assoc. Editor: Vaibhav Srivastava.

estimates, an information-theoretic motion planner automatically configures the sensor network to maximize information gain and thus minimizes measurement uncertainty. Few, if any, information-based motion planners have been developed for measured waveform parameters such as time difference of arrival (TDOA), frequency difference of arrival (FDOA), or differential received signal strength (DRSS). One most-related approach assumes that one of the two sensor locations is fixed [10]. The major challenge in 2D-location estimation using either TDOA or FDOA measurements is the sensor measurements are ambiguous (i.e., there is no unique answer); however, mobility provides more measurements with spatial variance to help eliminate incorrect estimates.

A novel approach is presented whereby the motion planner considers the two-locations-one-measurement case to calculate the mutual information (MI). Specifically, the MI is computed for pairs of sensors through a four-permutation-with-replacement process. The resulting MI surfaces are combined to create a composite map, which is then used to determine the optimal locations for any number of agents in a prescribed search space. The approach takes advantage of the geometry of the problem to reduce computation. Detailed simulation and experimental studies are performed to validate the method. The main contributions are:

- (1) Developing a mutual information-based algorithm that autonomously configures a team of mobile sensors for effective target estimation and localization;
- (2) Validating the algorithm through simulations for two to eight mobile robotic sensors;
- (3) Validating the algorithm through physical experiments with two and three micro-aerial vehicles with TDOA measurements tracking a moving mobile ground target; and
- (4) Quantifying the performance of the algorithms.

2 Prior Related Work

2.1 Cooperative and Uncooperative Targets. Localization and tracking can be applied to two types of targets: cooperative [11] and uncooperative [12]. In the cooperative case, the target shares information with agents to aid state estimation. Examples of signals that are often used are time of arrival and received signal strength measurements, where the initial transmission time and the initial signal strength are shared with the agents, respectively. These measurements, combined with the information shared by the target, allow the target's distance to be determined. On the other hand, an uncooperative target does not share information with agent(s). In this case, the target may not be aware of an agent tracking its location. The location of uncooperative targets can be inferred by (1) reflecting a signal off the target to make the initial transmission time or signal strength available (such as in radar, sonar, and LiDAR) or (2) introducing another sensor to acquire additional information so that relative measurements can be made (such as using TDOA, FDOA, and DRSS). An additional measurement enables alternative means of calculating possible target positions and is described later in Sec. 3.2.

2.2 State-of-the-Art Approaches. Algorithms have been developed for target localization and tracking using TDOA sensor measurements. Deterministic approaches that utilize explicit equations [13–15], least squares [16–18], Taylor series [19,20], and numerical algebraic geometry [21] have been investigated. These approaches work well when a sufficient number of TDOA measurements are available; however, when only one TDOA measurement is available, there is no unique solution. When there is a solution, it does not characterize the distribution of other likely target locations due to noise in the measurements.

Probabilistic methods include maximum likelihood estimators [22,23], genetic algorithms [24], Gaussian mixture models [25,26], bank of maximum a posteriori [27], Kalman filters [28–31], and particle filters [32–34]. Kalman [35] and particle filters [32] are particularly useful for recursive state estimation that leverage prior

knowledge of states in combination with current state measurements. One advantage of particle filters is they can handle non-Gaussian distributions. Extended and unscented Kalman filters have been developed to expand the usefulness of the Kalman filter and to deal with nonlinearity [36,37]. Overall, these probabilistic solutions make available an idea of the distribution of possible target locations, such as the covariance matrix in Kalman filters and the particle distribution in particle filters. The information can be updated over time to gain a reliable estimate of the target position, regardless of the number of measurements.

2.2.1 Static Sensor Networks. In Ref. [38], a stationary distributed network of sensors uses TDOA (and FDOA) measurements with an extended Kalman filter to track a constant-velocity target. This is taken a step further for tracking a dynamically-moving target, where a stationary sensor network is used, with an extended Kalman filter embedded in an interactive multiple model architecture [31]. The stationary receiver case was expanded in Ref. [30], where TDOA and FDOA measurements were used with an extended Kalman filter to track a target traveling in a circular trajectory. In Ref. [39], a particle filter outperformed an extended Kalman filter in a static-sensor scenario with the target moving in a straight line. The effect of using more than just TDOA measurements was investigated in Ref. [40] for a target that moved dynamically in 3D. In practice, static sensor networks can be costly to implement and maintain. Their detection range is also confined to a certain area and they are only useful if the target ventures into the area covered by the network.

2.2.2 Mobile-Sensor Networks. A network of mobile TDOA sensors is described in Ref. [28], where extended and unscented Kalman filters localized a constant-velocity target. The movement of the two agents followed predetermined paths. In a similar problem, a Gaussian mixture model was included where the agents again traveled in predefined trajectories [26]. An unscented Kalman filter was considered for FDOA in addition to TDOA with a constant-velocity target and predetermined paths for the agents in Ref. [29]. Mobile-sensor networks are more flexible to implement and deploy, and they are not limited to tracking a target within an established area.

2.3 Advancements. While probabilistic solutions can characterize the distribution of possible target locations and help inform future estimates, state-of-the-art target localization and tracking methods do not use this characterization of the possible target locations to guide the motion of or reconfigure the sensors. Instead, the sensor network remains static and is unable to perform as well when few sensors are available.

To advance the state-of-the-art, Bayesian estimation is combined with mutual information motion planning to optimally configure a network of mobile sensors. Specifically, MI is used to autonomously guide the motion of each mobile sensor to maximize information gain and minimize uncertainty for localization and tracking of an uncooperative radio frequency (RF) target. Recently, MI has been applied to problems such as search and rescue [41], emergency response [42], chemical-gas plume monitoring [8], and indoor target tracking [43]. The formulation described herein focuses on the use of TDOA measurements.

Computing MI is computationally demanding due to high-dimension integration. As the number of sensor nodes increases, the computation time increases exponentially. As such, prior MI work for mobile sensors often focuses on approximating the MI to reduce complexity and/or evaluating a single node and making pairwise comparisons [41]. The pairwise comparison process increases coordination between agents. In Ref. [44], the submodularity of MI is exploited. In Ref. [42], a decentralized method of calculating the MI is described. Each agent maintains its own belief of the environment and uses MI to position the sensor node. In Ref. [43], MI is approximated by reducing the number of particles. A

simplified method is used to calculate the gradient of MI in Ref. [45]; however, it is best applied to a single sensor.

The use of MI to coordinate the motion of a network of TDOA sensors for uncooperative target localization and tracking has not been well studied. The most-related effort assumes one fixed node [10,46]. Other works have focused on the placement of TDOA sensors using the Fisher information matrix [47–49].

3 Technical Approach

3.1 Notation. Lower-case letters are scalars, for example, $a \in \mathbb{R}$, where \mathbb{R} is the set of real numbers. Bold lower-case letters are vectors, for example, $\mathbf{a} \in \mathbb{R}^n$, where \mathbb{R}^n represents the n th-dimensional vector space over the reals. The variables for continuous-time and discrete-time are $t \in \mathbb{R}$ and $k \in \mathbb{Z}^+$, respectively, where \mathbb{Z}^+ is the set of positive integers. A continuous-time function is represented by $f(t) : \mathbb{R} \rightarrow \mathbb{R}$ and the discrete-time function by $z[k] : \mathbb{Z}^+ \rightarrow \mathbb{R}$. The sample time is assumed constant and denoted by $\Delta t \in \mathbb{R}$.

3.2 Time Difference of Arrival Measurement Model. In practice, TDOA measurements are computed by taking the difference of signal arrival times for two agents with synchronized clocks. The likely locations of the target can then be determined by using the time and range difference of arrival (RDOA) equations. The RDOA is found by the difference in the relative ranges

$$\text{RDOA} = \sqrt{(x_1 - x_T)^2 + (y_1 - y_T)^2} - \sqrt{(x_2 - x_T)^2 + (y_2 - y_T)^2} \quad (1)$$

where x_1 and y_1 are the x and y coordinates of one agent, respectively; x_2 and y_2 are the x and y coordinates of the second agent, respectively; and x_T and y_T are the x and y coordinates of the target, respectively. The TDOA can then be calculated as

$$\text{TDOA} = \frac{\text{RDOA}}{c} \quad (2)$$

where c is the speed of light.

Figures 2(a) and 2(b) show examples of target location distributions for two TDOA measurements. The measurements produce a non-Gaussian distribution of the target location for a given measurement. Figures 2(c) and 2(d) show that increasing the number of agents can decrease ambiguity in the location of the target. Depending on the configuration of the agents and target, a network of three agents is sufficient for a unique solution. The explicit solution for the intersection point(s) is nonlinear and the topic of several research papers cited in Sec. 2.

3.3 Problem Statement and Hypothesis. Given a network of mobile TDOA sensors (two or more), the goal is to maneuver the sensors to quickly and effectively localize an uncooperative target. An estimate of the target position is obtained using a machine learning process through Bayesian estimation in the form of a particle filter. The target states are estimated via a constant-velocity Kalman filter. Next, an MI-based motion planner [41] coordinates the motion of the sensors to reduce the uncertainty of the posterior and refine the target state estimation. This is done using a four-dimensional (4D) MI matrix which is computed for a specified grid space using a four-permutation-with-replacement process. The mutual information matrix is described in Sec. 3.7.

For this problem, the following assumptions are made:

- (1) Only one target exists;
- (2) The target altitude is known;
- (3) The maximum speed of the target is known;

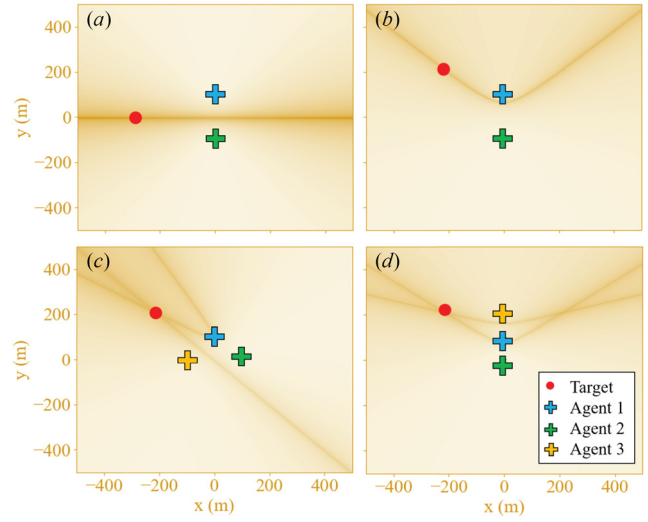


Fig. 2 Examples of different TDOA curves, where each two-sensor configuration creates a curve with points that have the same TDOA value: (a) zero TDOA and (b) less than the maximum TDOA with agent 1 receiving the signal first. When the number of agents increases, the intersection of the TDOA curves indicates the possible locations of the target. In configuration (c), the number of agents is sufficient for a unique solution, whereas in configuration (d) either an additional agent or an alternative configuration would be necessary for a unique solution.

- (4) The target moves around inside of a specified area (not necessarily the search area available to the agents);
- (5) Noise can be present in the TDOA measurement; and
- (6) Robot dynamics are ignored.

The following hypothesis is made: The MI-based motion planners will outperform the five uninformed methods. Specifically, the average percent root-mean-square error for the estimated target location will be lower for the MI-based motion planners. This hypothesis will be tested in simulations and physical experiments.

3.4 Overview of Algorithms for Comparison. Figure 3 shows the block diagram for the information theory 1 (IT1) algorithm. There are four main components to IT1: (1) Bayesian estimation in the form of a particle filter, (2) position and velocity state estimation through a Kalman filter, (3) an information-based motion planner, and (4) a mobile-sensor motion controller.

Let the search domain for the target state-space be given by \mathcal{A} . First, TDOA measurements are fed into the particle filter which provides estimated states and weights of the true target location, $\hat{\alpha}[k]$. For the i th particle, $\alpha^i = [x_T, y_T] \in \mathcal{A} \subset \mathbb{R}^2$ is the set of x and y coordinates for a single target and $w^i \in [0, 1]$ is the importance weight for each particle. The mean of the particles from the particle filter serves as an estimate only of the position of the target and produces noisy measurements since no motion model is considered. To filter this output and estimate the velocity of the target state ψ , a constant-velocity Kalman filter is used. The information-based motion planner uses a subset of 500 particles and the corresponding

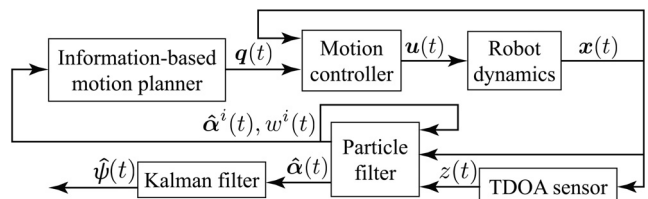


Fig. 3 Block diagram of the IT1 algorithm

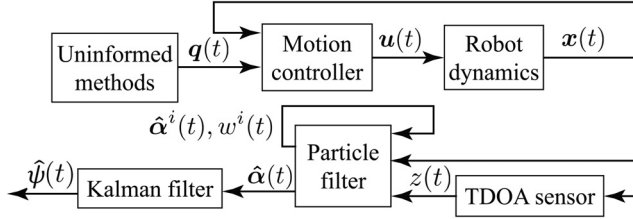


Fig. 4 Block diagram of the algorithm without information theory guided motion

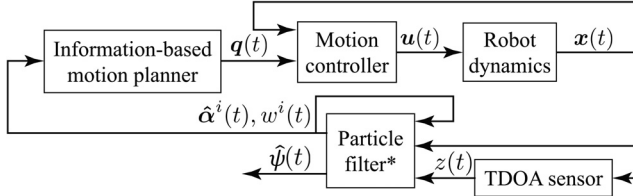


Fig. 5 Block diagram of the IT2 algorithm, which excludes the Kalman filter and uses a particle filter (denoted by “*”) with 40,000 particles

weights from the particle filter to generate information surfaces for every combination of two agents in a grid that spans the search space. The subset of particles is chosen to be 500 because it provides a good approximation of the target state at a much lower computational cost to create the MI surfaces. These surfaces are used to generate the best waypoints for the agents represented as $q(t)$. Finally, the motion controller enables the agents to travel to their respective waypoints with the command signal $u(t)$.

To compare the impact of the feedback in guiding the motion of the agents, the IT1 algorithm is compared to a general version of the same algorithm. The general algorithm is identical to IT1 except that it lacks feedback from the information-theoretic motion planner and is pictured in Fig. 4.

An additional algorithm, information theory 2 (IT2), is also investigated and the structure is depicted in Fig. 5. The differences between IT1 and IT2 are the number of particles used (8000 versus 40,000, respectively) and the lack of a Kalman filter on the output of IT2’s particle filter. Because of this, more particles are used for the IT2 algorithm to produce a better approximation; however, this comes with an increase in computational cost. Due to the lack of the Kalman filter, the velocity estimate is calculated by the derivative of the position estimates from the particle filter. The same information-based motion planner is used here as well that only considers a subset of 500 particles.

3.5 Particle Filter. Particle filters can handle non-Gaussian processes and multimodal distributions [8,32]. A Bayesian-based particle filter is implemented in the discrete-time domain where k denotes the discrete-time instant. Using the current measurement and the previous measurement, $z[k]$ and $z[k-1]$, respectively, in addition to the measurement history given by $z[k_s : k]$, the posterior probability distribution $p(\alpha[k]|z[k_s : k])$ can be found from Bayes’ rule as

$$p(\alpha[k]|z[k_s : k]) = \int_{\alpha[k-1]} p(\alpha[k]|\alpha[k-1]) \times p(\alpha[k-1]|z[k_s : k-1]) d\alpha[k-1] \quad (3)$$

$$p(\alpha[k]|z[k_s : k]) = \frac{p(z[k]|\alpha[k])p(\alpha[k]|z[k_s : k-1])}{p(z[k]|z[k_s : k-1])} \quad (4)$$

This integral equation is complex and computationally demanding; thus, the following nonparametric approximation is used:

$$p(\alpha[k]|z[k_s : k]) \approx \sum_{i=1}^{\eta} w^i[k] \delta(\alpha[k] - \alpha^i[k]) \quad (5)$$

for the i th particle. In Eq. (5), η denotes the finite number of particles used, and $\delta(\cdot)$ is the Dirac Delta function. Importance sampling is used to update the weights of particles, i.e.,

$$w^i[k] \propto w^i[k-1] p(z[k]|\alpha^i[k]) \quad (6)$$

The minimum mean-squared estimate of the posterior is given by [8]

$$\hat{\alpha}[k] = \int_{\alpha} \alpha[k] p(\alpha[k]) d\alpha[k] \approx \sum_{i=1}^{\eta} w^i[k] \alpha^i[k] \quad (7)$$

where “ \wedge ” indicates the approximated location of the target. The likelihood model is used to calculate the probability of a measurement for a particle target state estimate $\alpha^i[k]$ and is chosen to be

$$p(z[k]|\alpha^i[k]) = \mathcal{N}(z[k], \hat{z}^i[k]) \quad (8)$$

where \mathcal{N} is the normal distribution, and $\hat{z}^i[k]$ is the estimated measurement value calculated by Eqs. (1) and (2) with the current sensor configuration at time instant k and the target state given by the i th particle state estimation.

3.6 Kalman Filter. Kalman filters are recursive filters that can estimate states given noisy measurements [35]. A Kalman filter consists of three major steps: state prediction, Kalman gain update, and measurement update.

3.6.1 State Prediction

$$\hat{\psi}_{k|k-1} = F_k \hat{\psi}_{k-1|k-1} \quad (9)$$

$$\hat{P}_{k|k-1} = F_k \hat{P}_{k-1|k-1} F_k^T + Q_{k-1} \quad (10)$$

3.6.2 Kalman Gain Update

$$S_k = H_k \hat{P}_{k|k-1} H_k^T + R_k \quad (11)$$

$$K_k = \hat{P}_{k|k-1} H_k^T S_k^{-1} \quad (12)$$

3.6.3 Measurement Update

$$\tilde{y}_k = z_k - H_k \hat{\psi}_{k|k-1} \quad (13)$$

$$\hat{\psi}_{k|k} = \hat{\psi}_{k|k-1} + K_k \tilde{y}_k \quad (14)$$

$$\hat{P}_{k|k} = \hat{P}_{k|k-1} - K_k S_k K_k^T \quad (15)$$

For the Kalman filter equations, the state estimate $\hat{\psi}$ has the form $[x_T, y_T, \dot{x}_T, \dot{y}_T]^T$, where x_T and y_T are the target position estimates and \dot{x}_T and \dot{y}_T are the target velocity estimates. The initial state estimate $\hat{\psi}_{k-1|k-1}$ is set equal to the mean of the x and y particles from the particle filter $\hat{\alpha}_k$. Furthermore, z_k is the received measurement at time-step k . The initial covariance matrix $\hat{P}_{k-1|k-1}$ is $[2, 0, 0, 0; 0, 2, 0, 0; 0, 0, 0.5, 0; 0, 0, 0, 0.5]$. The state transition matrix is defined as $F_k = [1, 0, \Delta t, 0; 0, 1, 0, \Delta t; 0, 0, 1, 0; 0, 0, 0, 1]$. Since the particle filter in the previous step converts the TDOA measurement(s) into a position measurement, the observation matrix is defined as $H_k = [1, 0, 0, 0; 0, 1, 0, 0]$. The process noise covariance is constant and defined as $Q_{k-1} = \Delta t^*$

[1/4, 0, 1/2, 0; 0, 1/4, 0, 1/2; 1/2, 0, 1, 0; 0, 1/2, 0, 1], where Δt is based on an update rate of 100 Hz and the error matrix is constant and defined as $\mathbf{R}_k = [0.1, 0; 0, 0.1]$. Although a constant-velocity Kalman filter is used to track the dynamically-moving target, the fast sampling rate used allows for more robust tracking and the process noise does not play as big of a role as the measurement noise.

3.7 Information-Based Motion Planner. The goal of the motion planner is to ensure the agents autonomously navigate to the most informative configuration as the target location is estimated over time. To achieve this, information-theoretic techniques are used to determine which configurations maximize information gain and minimize the uncertainty in the estimation process. It is assumed that the target is not aware of the previous measurements that the agents received. The following cost function is considered:

$$J(\boldsymbol{\theta}, p(\boldsymbol{\alpha}[k]|z[k_s : k])) = -I(z[k]; \boldsymbol{\alpha}[k]|z[k_s : k-1]) \quad (16)$$

where $\boldsymbol{\theta}$ is a vector of $x_1, y_1, x_2,$ and y_2 which represents the locations of two agents in x and y coordinates. The mutual information I is expressed as

$$I(z[k]; \boldsymbol{\alpha}[k]|z[k_s : k-1]) = I(z[k]; \boldsymbol{\alpha}[k]) \\ = H(z[k]) - H(z[k]|\boldsymbol{\alpha}[k]) \quad (17)$$

Therefore, to maximize mutual information, the difference between $H(z[k])$ and $H(z[k]|\boldsymbol{\alpha}[k])$ must be maximized, where

$$H(z[k]) = -\int_{z[k] \in \mathcal{Z}} p(z[k]) \log(p(z[k])) dz[k] \quad (18)$$

and

$$H(z[k]|\boldsymbol{\alpha}[k]) = -\int_{z[k] \in \mathcal{Z}} \int_{\boldsymbol{\alpha}[k] \in \mathcal{A}} p(z[k], \boldsymbol{\alpha}[k]) \\ \times \log(p(z[k]|\boldsymbol{\alpha}[k])) dz[k] d\boldsymbol{\alpha}[k] \quad (19)$$

Equations (18) and (19) can be approximated for a particle filter using

$$H(z[k]) \approx -\int_{z[k] \in \mathcal{Z}} \sum_{i=1}^n w_i^i p(z[k]|\boldsymbol{\alpha}[k] = \boldsymbol{\alpha}[k]^i) \\ \times \log \left(\sum_{i=1}^n w_i^i p(z[k]|\boldsymbol{\alpha}[k] = \boldsymbol{\alpha}[k]^i) \right) dz[k] \quad (20)$$

$$H(z[k]|\boldsymbol{\alpha}[k]) \approx -\int_{z[k] \in \mathcal{Z}} \sum_{i=1}^n w_i^i p(z[k]|\boldsymbol{\alpha}[k] = \boldsymbol{\alpha}[k]^i) \\ \times \log(p(z[k]|\boldsymbol{\alpha}[k] = \boldsymbol{\alpha}[k]^i)) dz[k] \quad (21)$$

respectively. These equations are evaluated using numerical integration techniques.

3.7.1 Mutual Information Matrix. Equations (16)–(21) evaluate the mutual information given a sensor pair, particles, and particle weights and will be referred to as $\mathcal{H}(\boldsymbol{\theta}, \hat{\boldsymbol{\alpha}}^i, \mathbf{w}^i)$. Since the mutual information is dependent upon a sensor pair, every quartet of possible locations for the agents ($x_1, y_1, x_2,$ and y_2) must be evaluated in \mathcal{H} and stored in a four-dimensional mutual information matrix

$$\mathcal{I}[i, j, k, l] = \mathcal{H}([\mathbf{x}_g[i], \mathbf{y}_g[j], \mathbf{x}_g[k], \mathbf{y}_g[l]], \hat{\boldsymbol{\alpha}}^i, \mathbf{w}^i) \quad (22)$$

where $i, j, k,$ and l are indices that correspond to TDOA sensor pair configurations in a grid. By using a four-permutation-with-replacement process, all possible sensor pair configurations are evaluated. The vectors \mathbf{x}_g and \mathbf{y}_g contain the possible x and y locations for the

agents in the grid. The pseudo-code is depicted in Algorithm 1. The maximum of \mathcal{I} corresponds to which quartet provides the most information, or the best locations to place two agents to maximize information gain and minimize uncertainty about the target states. The argmax function is defined to output the indices of the maximum value in a matrix.

Algorithm 1 Process to compute the information theory matrix, where \mathbf{x}_g and \mathbf{y}_g correspond to vectors of potential x and y locations, respectively

```

1 for  $i=0$  to length( $\mathbf{x}_g$ ) do
2    $\theta[0] = \mathbf{x}_g[i]$ 
3   for  $j=0$  to length( $\mathbf{y}_g$ ) do
4      $\theta[1] \leftarrow \mathbf{y}_g[j]$ 
5     for  $k=0$  to length( $\mathbf{x}_g$ ) do
6        $\theta[2] = \mathbf{x}_g[k]$ 
7       for  $l=0$  to length( $\mathbf{y}_g$ ) do
8          $\theta[3] \leftarrow \mathbf{y}_g[l]$ 
9          $\mathcal{I}[i, j, k, l] \leftarrow \mathcal{H}(\boldsymbol{\theta}, \hat{\boldsymbol{\alpha}}^i, \mathbf{w}^i)$ 

```

Figure 6 provides examples of two-dimensional subsets of \mathcal{I} . In Fig. 6(a), the information surface when agent 1 is fixed in a corner is depicted. The surface shows the amount of information available for possible locations of agent 2. From this surface, it can be determined that the best location to place agent 2 is in the location with the maximum information or the opposite corner from agent 1. Conversely, as seen in Fig. 6(b), if agent 2 is fixed at the location determined from Fig. 6(a), the information surface will show the best location to place agent 1, which clearly will be the fixed location described in Fig. 6(a). It is also important to recognize that the information surface Fig. 6(a) is the same even if agent 2 is fixed at the location of agent 1 and the information is calculated for various locations of agent 1. Finally, it is important to note that if agent 2 is placed in the same position as agent 1, zero information will result. These facts allow for a reduction in computational time when generating the 4D matrix of information.

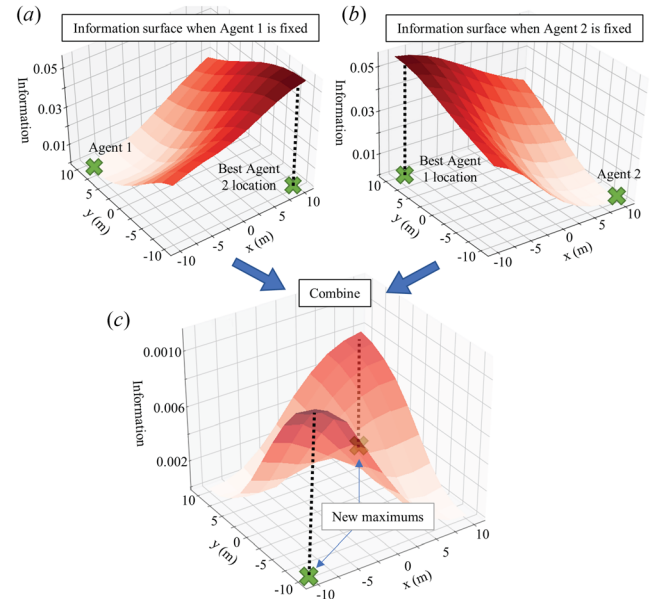


Fig. 6 The surface with the maximum information available is multiplied by the opposing surface generated by the same sensor pair. This creates a new surface that has zero information at the locations of the agents. The argmax of the new surface corresponds with where the next agent should be placed.

3.7.2 *Placement of Agents for Time Difference of Arrival Measurements.* Once computed, \mathcal{I} is used to place as many agents as desired in the order of maximum information gain. Algorithm 2 and Fig. 6 depict the process of placing the agents. First, the argmax of \mathcal{I} is determined, and the information surfaces corresponding to it are obtained as in Figs. 6(a) and 6(b). These surfaces correspond to where to place the first two agents. Next, the surfaces Figs. 6(a) and 6(b) are combined via multiplication to produce a new surface Fig. 6(c), referred to as \mathcal{M}_2 . The argmax of \mathcal{M}_2 is used to determine where to place the third agent. To continue placing more agents, the information surface for the fixed third agent is obtained from \mathcal{I} and combined with \mathcal{M}_2 Fig. 6(c) to create surface \mathcal{M}_3 . The argmax is taken of \mathcal{M}_3 to place the fourth agent and the process continues for placing as many agents as desired. The x and y locations to place the agents are stored in vectors $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$.

Algorithm 2 Process of combining surfaces to determine the best locations to place the agents, where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ correspond to vectors of where to place the agents in x and y coordinates, respectively, and the number of agents is a

```

1   $i, j, k, l = \text{argmax}(\mathcal{I})$ 
2   $\mathcal{M}_2 = \mathcal{I}[i, j, :, :] * \mathcal{I}[:, :, k, l]$ 
3   $\tilde{\mathbf{x}} = [\mathbf{x}_g[i], \mathbf{x}_g[k]]$ 
4   $\tilde{\mathbf{y}} = [\mathbf{y}_g[j], \mathbf{y}_g[l]]$ 
5  if  $a \geq 3$  then
6    for  $n = 3$  to  $a$  do
7       $i, j = \text{argmax}(\mathcal{M}_{n-1})$ 
8       $\mathcal{M}_n = \mathcal{M}_{n-1} * \mathcal{I}[i, j, :, :]$ 
9       $\tilde{\mathbf{x}}.append(\mathbf{x}_g[i])$ 
10      $\tilde{\mathbf{y}}.append(\mathbf{y}_g[j])$ 

```

3.8 Error Metric. The root-mean-square-error (RMSE), \bar{e} , is used to compare the proposed algorithms to other common approaches. The RMSE is defined as

$$\bar{e} = \sqrt{\frac{\sum_{i=1}^n (m_i - \hat{m}_i)^2}{n}} \quad (23)$$

where n is the number of observations, m_i is the observed value, and \hat{m}_i is the estimated value.

4 Experimental Design

Experiments compared the IT1, IT2, and uninformed methods. Figure 4 shows the uninformed algorithm without the feedback from the particle filter. It is also important to note that the algorithms depicted in Figs. 4 and 3 are identical except for the lack of this feedback mechanism. Instead, waypoints for the agents in the algorithm depicted in Fig. 4 are predefined and not based on the current information. Comparing these scenarios helps to determine how useful this feedback is in practice. The goal of the experiment is to track the target as closely as possible for the duration of the experiment.

The sampling frequency of the hardware (control and sensing) is 40 Hz; however, in the IT1 and IT2 cases, the agents are given new waypoints each second. The target is given the same path for all scenarios but slight deviations from the path in the physical experiments are possible though not significant enough to impact the outcome. The size of the physical platform (4.2 m \times 2.6 m) is also used as a constraint for simulations to allow for better cross-comparison between simulations and physical experiments. The agent search area is limited to a 1.5 m \times 1.5 m area, where the x -direction search range spans 0.65–2.15 m and the y -direction search range spans 0.35–1.85 m. Figure 7 shows the area of interest and the path for which all simulations and physical experiments are

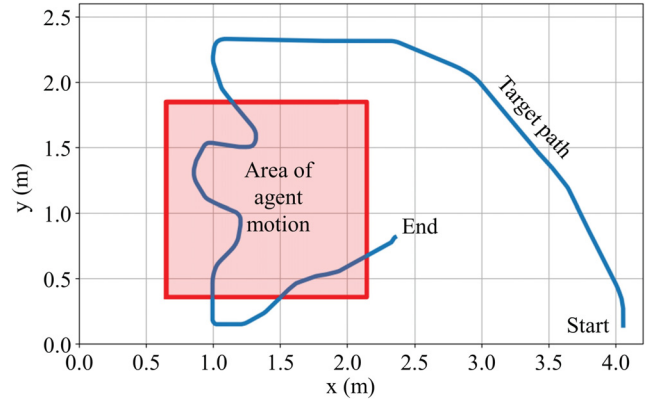


Fig. 7 Map of the experimental layout. This layout and its parameters are consistent across both simulation and physical experiments.

performed. The maximum speed of the agents in simulation and physical experiments is set at 0.15 m/s. The speed of the target is constant and equal to that of the agents. However, in cases where there are only two agents, the speed of the target is half at 0.075 m/s.

4.1 Simulation Setup. Simulations are created using PYTHON (version 3.8.3) on a Windows 10 machine. A TDOA measurement error of 1% is used. Simulations are completed for two to eight agents using the following seven different methods (100 each, 4900 total):

- (M1) *Static border:* The agents are placed evenly spaced around the border and do not move. Agent 1 is placed in the bottom left-hand corner, and the remaining agents are placed evenly in a counterclockwise fashion.
- (M2) *Static uniform:* The agents are placed randomly, in a uniform fashion throughout the area and do not move.
- (M3) *Rotation:* The agents are placed as described in M1, but move in a counterclockwise fashion on the border.
- (M4) *Raster:* The agents are placed as described in M2, but then move in a raster-scan pattern.
- (M5) *Random:* The agents are placed as described in M2, but then move randomly within the search space.
- (M6) *IT1:* The agents are placed as described in M2, but then follow the motion trajectory generated by the information theory algorithm described in Fig. 3.
- (M7) *IT2:* The agents are placed as described in M2, but then follow the motion trajectory generated by the information theory algorithm described in Fig. 5.

Methods M1–M5 are the uninformed methods mentioned in Fig. 4. Examples of methods M1–M5 are illustrated in Fig. 8. For the two-agent case, the speed of the target is half that of the agents' speed.

4.2 Physical Experiment Setup. Physical experiments are performed to test the efficacy of the particle filter and MI-based motion planners (IT1 and IT2). Figure 9 summarizes the experimental system and setup. The experiments are performed on a desktop computer running LINUX UBUNTU 18.04 and the ROBOT OPERATING SYSTEM (ROS) MELODIC. The uncooperative target is a custom-designed omnidirectional ground robot equipped with an Odroid XU4 single board computer interfaced with a Robotis U2D2 that controls the Dynamixel servomotors (Fig. 9(a)). The single board computer runs the ROS environment for vehicle control and navigation after receiving waypoint commands from the ground station (GS) through a 2.4/5 GHz WiFi module. The target robot is also equipped with an RF emitter, loco positioning system (LPS) deck, which operates similarly to the LPS node without the added computing power of the STM32F072 MCU. The RF-sensing aerial

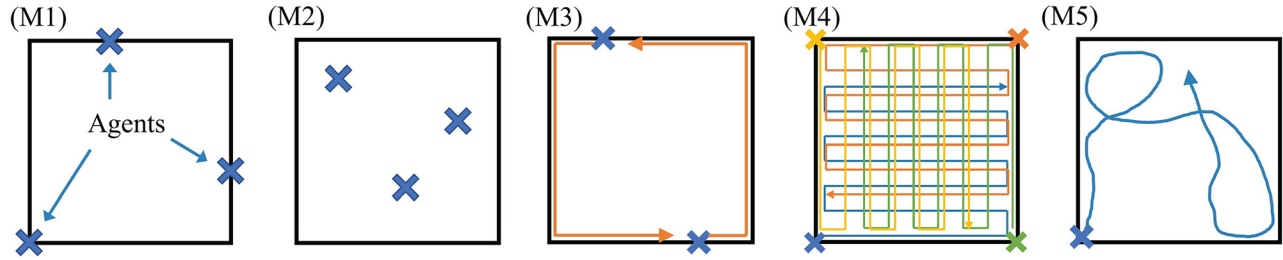


Fig. 8 Examples of the five uninformed methods used for comparison: (M1) three static agents evenly spaced on the border, (M2) three static agents randomly distributed with uniform placement, (M3) two agents evenly spaced and orbiting along the border, (M4) four agents scanning in a raster pattern, and (M5) one agent moving randomly. For each of the methods, simulations with two to eight agents are conducted.

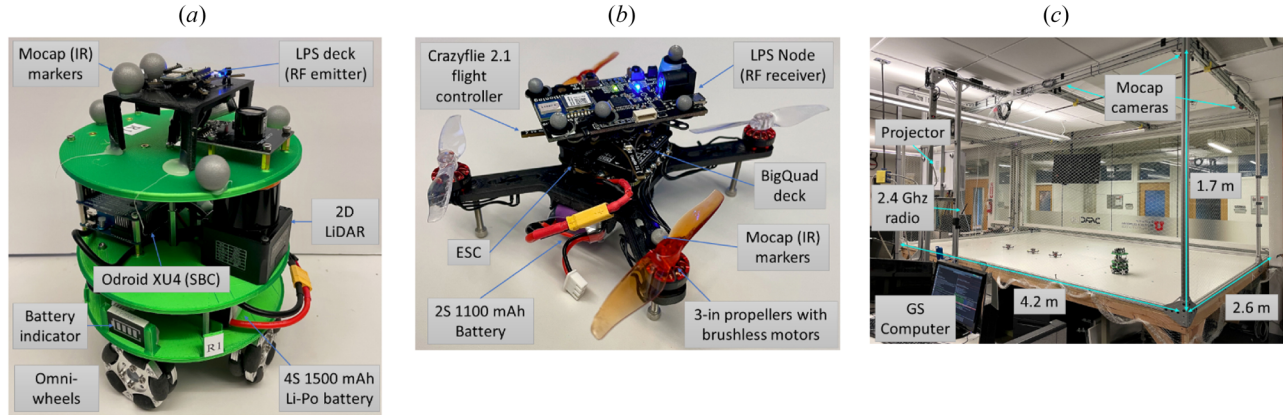


Fig. 9 Physical experiment system and setup: (a) a custom omnidirectional ground robot acting as the target, (b) custom quadcopters (DARCFlys) as agents, and (c) an experimental platform equipped with an Optitrack motion capture system

vehicles (small custom-built quadcopters named DARCFly) are designed and developed based on the CRAZYFLIE 2.1 (Bitcraze, Malmö, Sweden) platform. By incorporating an electronic speed controller and BigQuad deck, brushless motors are used and powered by a 2S 1100 mAh battery (Fig. 9(b)). The upgraded motors and battery allow for greater payload capacity and longer flight times. Furthermore, each robot carries a LPS node, also by Bitcraze, which uses a Decawave DWM1000 module for RF sensing and STM32F07 MCU for additional on-board computation. The GS is equipped with a 2.4 GHz industrial, scientific and medical band radio that communicates with the DARCFly and sends flight commands. The GS computes all necessary calculations, including the particle filter, MI, and position controller for the robots. All experiments are conducted on a 4.2 m × 2.6 m × 1.7 m platform with ten Flex 13 Optitrack motion capture (Mocap) cameras connected to a GS computer (Fig. 9(c)).

4.2.1 Sensor Characterization. To detect true TDOA measurements, the above-mentioned LPS node and LPS deck are used. These are equipped with a DWM1000 module, a single, low-power chip equipped with all necessary RF circuitry, i.e., antenna, RF transceiver, power management, ultrawideband, and clock circuit. The module is used for two-way ranging or TDOA location systems and deals with all signal processing to report a TDOA measurement. The onboard clock manages clock synchronization across sensors and is equipped with a 38.4-MHz reference crystal which is trimmed during production to get a frequency calibration offset of 2 ppm. TDOA measurements are dependent upon the speed of light. This fact means that for submeter accuracy in small spaces, the time of arrival of the signal needs to be accurate to the nanosecond—a high expectation for a low-cost sensor. The inexpensive sensor is characterized to determine how accurate and reliable it performs in physical experiments. Various tests are conducted to determine

the cause of sensor measurement errors. Tests reveal that the angle of the antenna impacts the performance of the sensor. This behavior, however, is nonlinear and cannot practically be inversely modeled. One simplifying assumption that appears to make the sensor behave more consistently is to direct the sensor antenna at the closest agent (see Fig. 10). Therefore, for the duration of the experiments, the target robot rotates to point in the direction of the closest agent. Figure 11 shows the percent error for two TDOA sensor measurements when a target follows the path in Fig. 7 and the agents are in the static border configuration.

4.2.2 Physical Experiments. Physical experiments are conducted only for the two and three agents' scenarios. This limits the number of agents that can fly in the small 1.5 m × 1.5 m volume. The IT1 and IT2 methods are both tested; however, to prevent collisions between the agents, the only uninformed methods used for comparison are the static border (M1) and rotation (M3) cases.

From the sensor characterization step, physical experiments are conducted in two batches. The first physical experiments use a simulated sensor with the same assumed 1% error as in the simulated experiments. The second set of physical experiments use the real sensor with an assumed 10% error.

5 Results

The RMSE error is given by

$$\bar{e} = \sqrt{\frac{\sum_{n=1}^k (\psi[n] - \hat{\psi}[n])^2}{k}} \quad (24)$$

and the error is calculated for each of the various experiments over the entire path. The mean RMSE is then computed using all of the

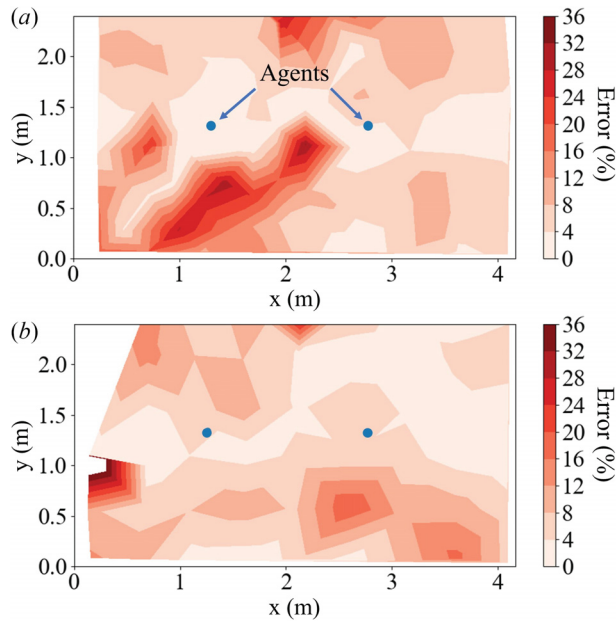


Fig. 10 The percent error of the TDOA measurement when the target is at different locations in space. Two different scenarios are presented: (a) the percent error when the target antenna is directed perpendicular to the x -axis and (b) the percent error when the target antenna is directed at the closest agent.

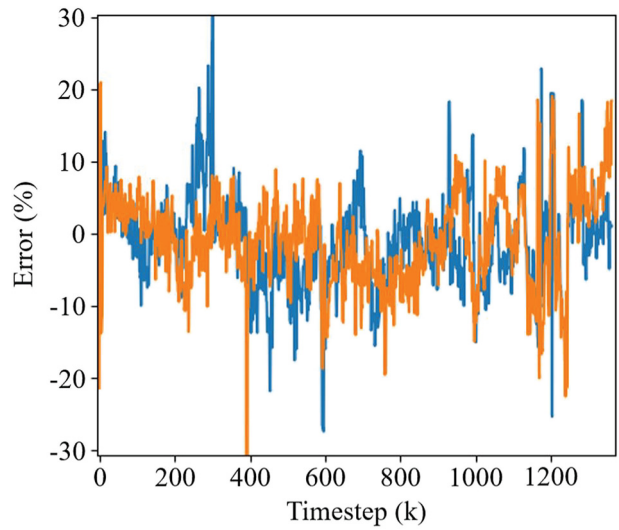


Fig. 11 The percent error of two measurements when the target follows the prescribed path and directs itself at the closest agent (sampling rate of 20 Hz). The agents are in the static border configuration.

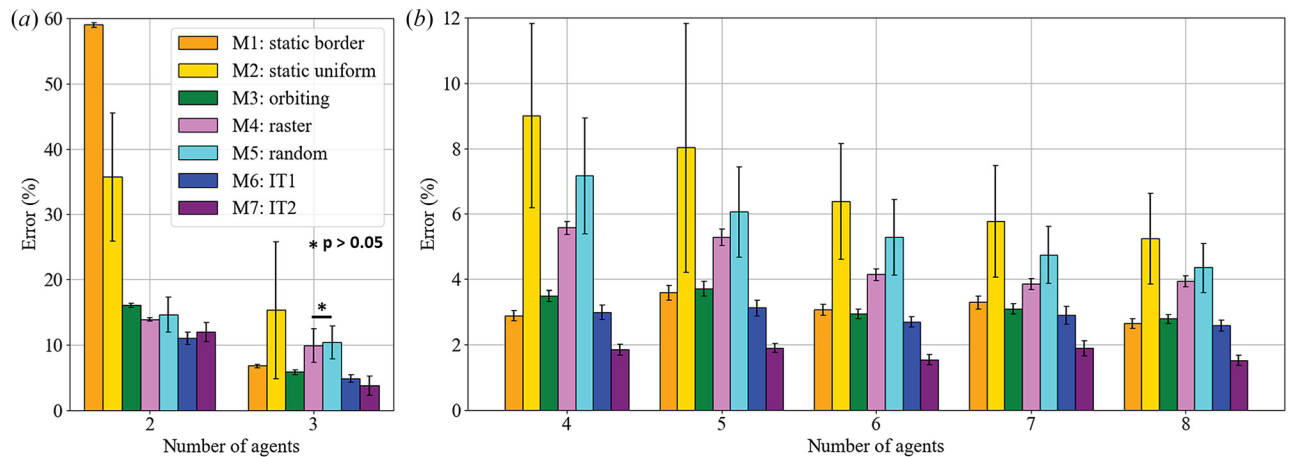


Fig. 12 The percent error for 100 simulations across all methods studied, for two to eight agents. The two and three-agent cases are shown in (a), where methods M4 and M5 are not significantly different when three agents are used. The four through eight agent scenarios are shown in (b).

Table 1 The mean and standard deviation of the normalized percent RMSE values for 100 simulations for all methods (M1–M7) and agent (two to eight) combinations

Methods	Number of agents							
	2	3	4	5	6	7	8	
M1: static border	59.04/0.35	6.76/0.28	2.89/0.16	3.59/0.23	3.08/0.17	3.30/0.20	2.65/0.15	
M2: static uniform	35.76/9.84	15.31/10.52	9.01/2.82	8.03/3.81	6.39/1.77	5.77/1.71	5.24/1.39	
M3: orbiting	16.07/0.33	5.81/0.38	3.49/0.17	3.71/0.22	2.95/0.15	3.10/0.16	2.79/0.13	
M4: raster	13.91/0.25	9.92/2.56	5.58/0.20	5.29/0.25	4.14/0.18	3.85/0.17	3.93/0.17	
M5: random	14.66/2.70	10.42/2.55	7.17/1.77	6.06/1.39	5.29/1.16	4.75/0.87	4.35/0.75	
M6: IT1	11.02/0.96	4.85/0.62	2.99/0.22	3.13/0.24	2.70/0.16	2.90/0.27	2.58/0.17	
M7: IT2	12.00/1.46	3.73/1.49	1.85/0.17	1.90/0.13	1.54/0.15	1.89/0.24	1.52/0.15	

The mean is reported first followed by the standard deviation.

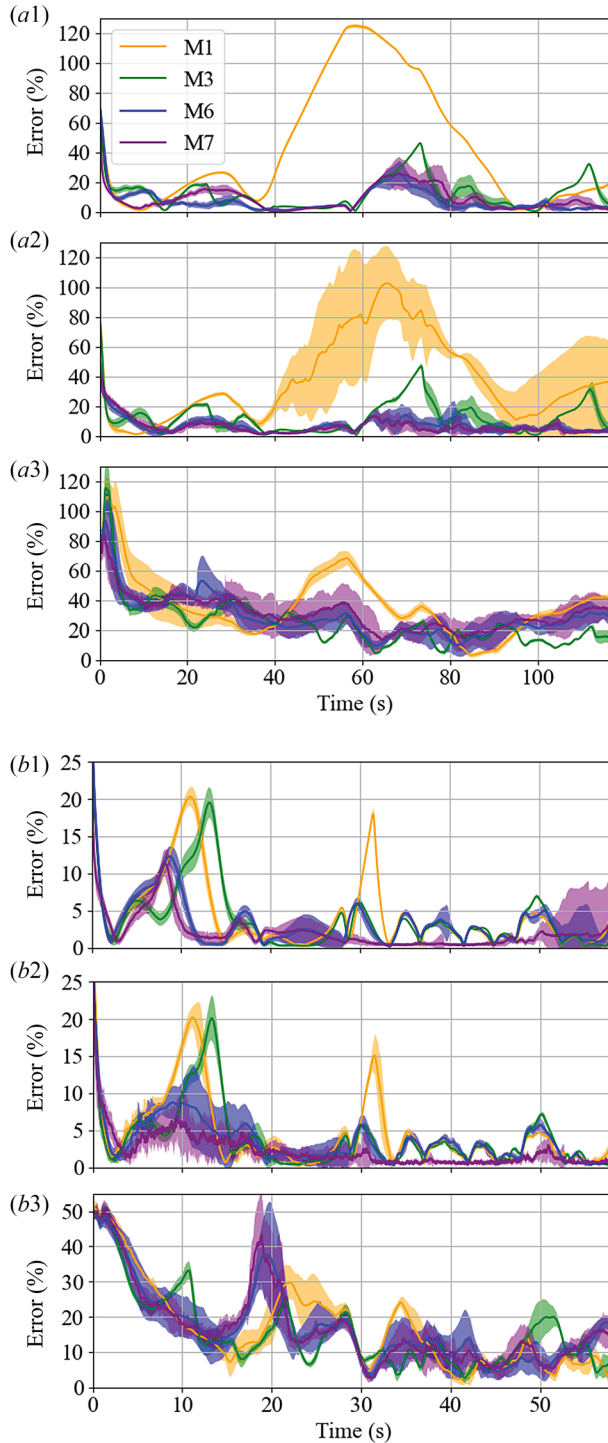


Fig. 13 The average percent error and standard deviation as it evolves in time for the various target tracking algorithms. The (a1)–(a3) and (b1)–(b3) plots correspond to the performance of the algorithm using two and three agents, respectively. (a1) and (b1) is from simulations. (a2) and (b2) are from physical experiments with a simulated sensor. (a3) and (b3) are from physical experiments with a real sensor.

tests for each experimental condition (100 tests for simulations and ten tests for physical experiments). The mean RMSE values are then converted to percent error. This is done by normalizing the data with respect to the maximum Euclidean distance between two agents in the search space as follows:

$$\bar{e}_p = \frac{\bar{e}}{l} \times 100\% \quad (25)$$

where $l = \sqrt{1.5^2 + 1.5^2}$ m is the maximum distance between the agents.

5.1 Simulation Results. Simulations are conducted for two to eight agents for all seven methods (M1–M7). For each agent/method combination, 100 simulations are run. The mean and standard deviation for all cases are illustrated in the bar graphs in Fig. 12. No significant difference is found between methods M4 and M5 when three agents are used ($p=0.17$). Table 1 reports the mean and standard deviation values from Fig. 12.

An example of how the average percent error and standard deviation evolve over time for method M1, M3, M6, and M7 for two and three agents are shown in Figs. 13(a1) and 13(b1), respectively.

Specific simulation examples for two and three agents are illustrated in Figs. 14 and 15, respectively, where (a1) is method M1, (a2) is method M3, (a3) is method M6, and (a4) is method M7.

5.2 Physical Experiment With Simulated Sensor. Physical experiments are conducted with two and three agents using a simulated sensor with a 1% error. The methods utilized are M1, M3, M6, and M7. For each of these methods, ten physical experiments are conducted. The mean and standard deviation for the experiments are illustrated in the bar graph in Fig. 16. When two agents are used, no significant difference is found between the M6 and M7 cases ($p=0.087$). When three agents are used, no significant difference is found between the M3 and M6 cases ($p=0.63$). Table 2 reports the mean and standard deviation values from the bar graph. Figure 17 shows three instances in time of an experiment with three agents using the IT2 algorithm (method M7).

In Fig. 13, plots (a2) and (b2) show the average percent error and standard deviation as they evolve over time for the ten physical experiments for two and three agents, respectively.

Specific physical experiment examples using a simulated sensor with 1% sensor error for both two and three agents are shown in Figs. 14 and 15, respectively. Four different methods are compared: (b1) is method M1, (b2) is method M3, (b3) is method M6, and (b4) is method M7.

5.3 Physical Experiment With Real Sensor. Physical experiments are conducted with two and three agents using a real sensor with a high degree of error (see Fig. 11). The methods investigated are M1, M3, M6, and M7. For each of these methods, ten physical experiments are conducted. The mean and standard deviation for the experiments are illustrated in Fig. 16. When using two agents, no significant difference is found between M6 and M7 ($p=0.53$). When using three agents, no significant difference is found for three comparisons: M1 and M6 ($p=0.41$), M1 and M7 ($p=0.97$), and M6 and M7 ($p=0.51$). Table 3 reports the mean and standard deviation values from Fig. 16.

In Figs. 13(a3) and 13(b3) show the average percent error and standard deviation as they evolve over time for the ten physical experiments for two and three agents, respectively.

Specific physical experiment examples using the real sensor with an assumed 10% sensor error for both two and three agents are shown in Figs. 14 and 15, respectively. Four different methods are compared: (c1) is method M1, (c2) is method M3, (c3) is method M6, and (c4) is method M7 (Fig. 18).

6 Discussion

This study introduced a novel method of autonomously positioning a network of agents using mutual information for the localization and tracking of a dynamically moving uncooperative target. The performance of the information-based algorithm and uninformed methods were evaluated using Monte Carlo simulations and

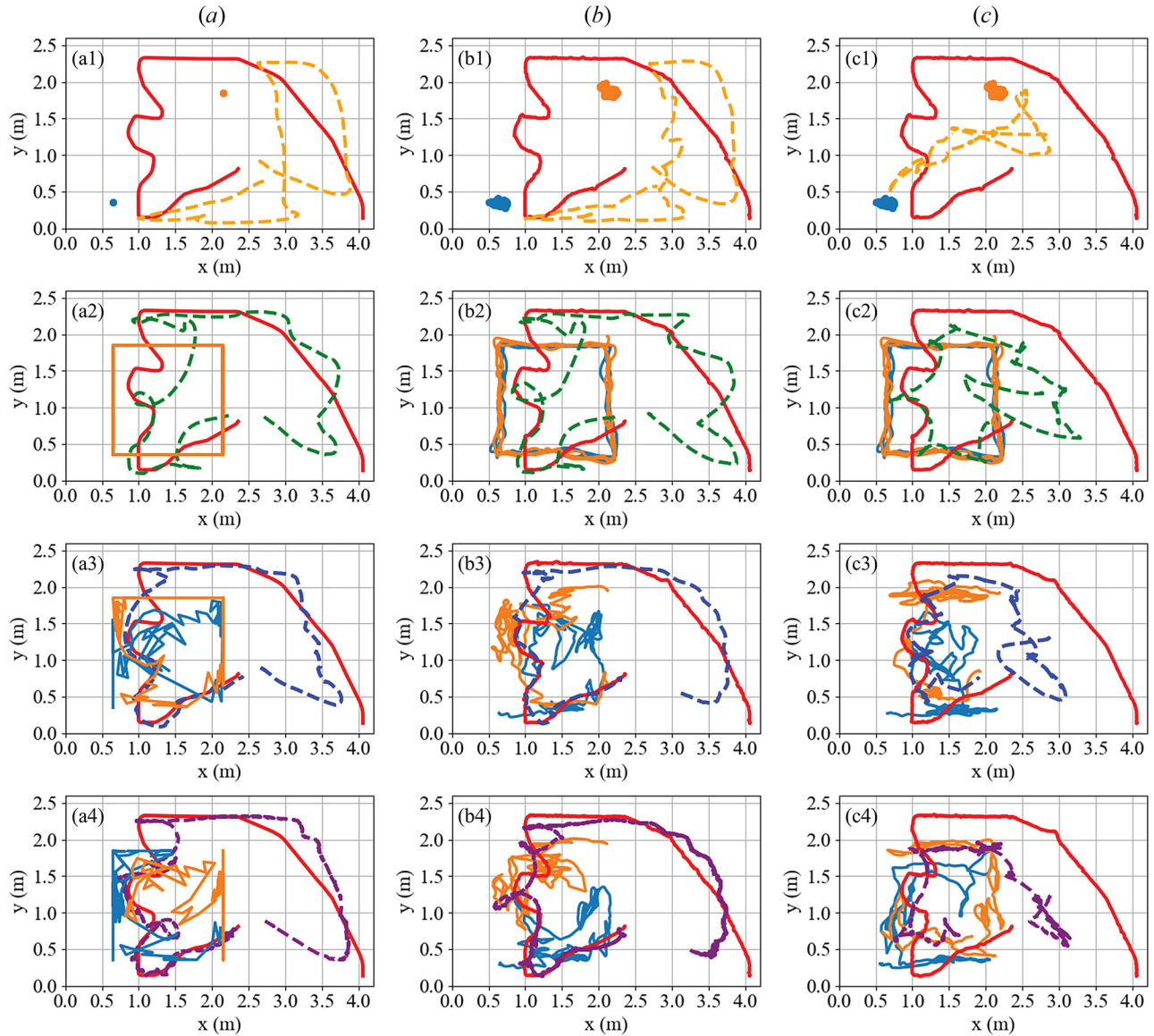


Fig. 14 Example of methods M1, M3, M6, and M7 (rows 1–4, respectively) with two agents. Agent paths stay within the square-shaped constraint, the true target path is in the solid line, and the estimated target path is the dashed line. Simulations are in column (a). Physical experiments with a simulated sensor are in column (b). Physical experiments with a real sensor are in column (c).

physical experiments. All tests are conducted on the same dynamic path.

An important observation is the difference in the estimation process between the first six methods and IT2 (M7). The estimation process for the first six methods is the same; they each use the same particle filter and constant-velocity Kalman filter. The only difference between the first six methods is the type of uninformed method used or the presence of the particle filter feedback in the motion planner for the IT1 case. IT2 is different because it does not utilize a Kalman filter and the particle filter uses 40,000 particles instead of 8000 particles (see Figs. 3–5).

It is clear from Fig. 12 that the performance of the IT2 algorithm is superior to all other methods when there are three or more agents. This result is expected since there is no constant-velocity Kalman filter. Between the 30-s and 50-s time instance in Figs. 13(b1) and 13(b2), several peaks of error are observed for methods M1, M3, and M6. These peaks are notably absent for the IT2 algorithm, and they correspond to the rapid changes in the target velocity seen in Figs. 15(a1)–15(a4). It can be seen in Fig. 15(a4) that the target estimate stays on track as the target moves. On the other hand, Figs. 15(a1)–15(a3) show significant overshoot due to the

anticipation of a constant-velocity by the Kalman filter. The drawback of using the IT2 algorithm instead of the IT1 algorithm, despite the position estimate being more accurate, is the velocity estimate has much higher error for the IT2 algorithm (see Fig. 19).

6.1 Simulations Results. As shown in Fig. 12 and Table 1, the IT1 algorithm performs better than the next closest uninformed method, by 2.89% error when two agents are used. It is emphasized that in the two-agent case, the target and agents do not move at the same speed, rather, the agents are capable of moving at twice the speed of the target. Figure 2 visualizes the importance of more than one TDOA measurement for localization. Allowing the agents to move up to twice the speed of the target allows for pseudo-intersections of TDOA measurements to occur. This is one reason the static methods (M1 and M2) yield poor performance. This two-agent result is significant because it shows that when one is limited to just two sensors, the IT1 algorithm outperforms state-of-the-art methods.

As illustrated in Fig. 12 and Table 1, the hypothesis that the MI-based motion planner would better localize and track a dynamic

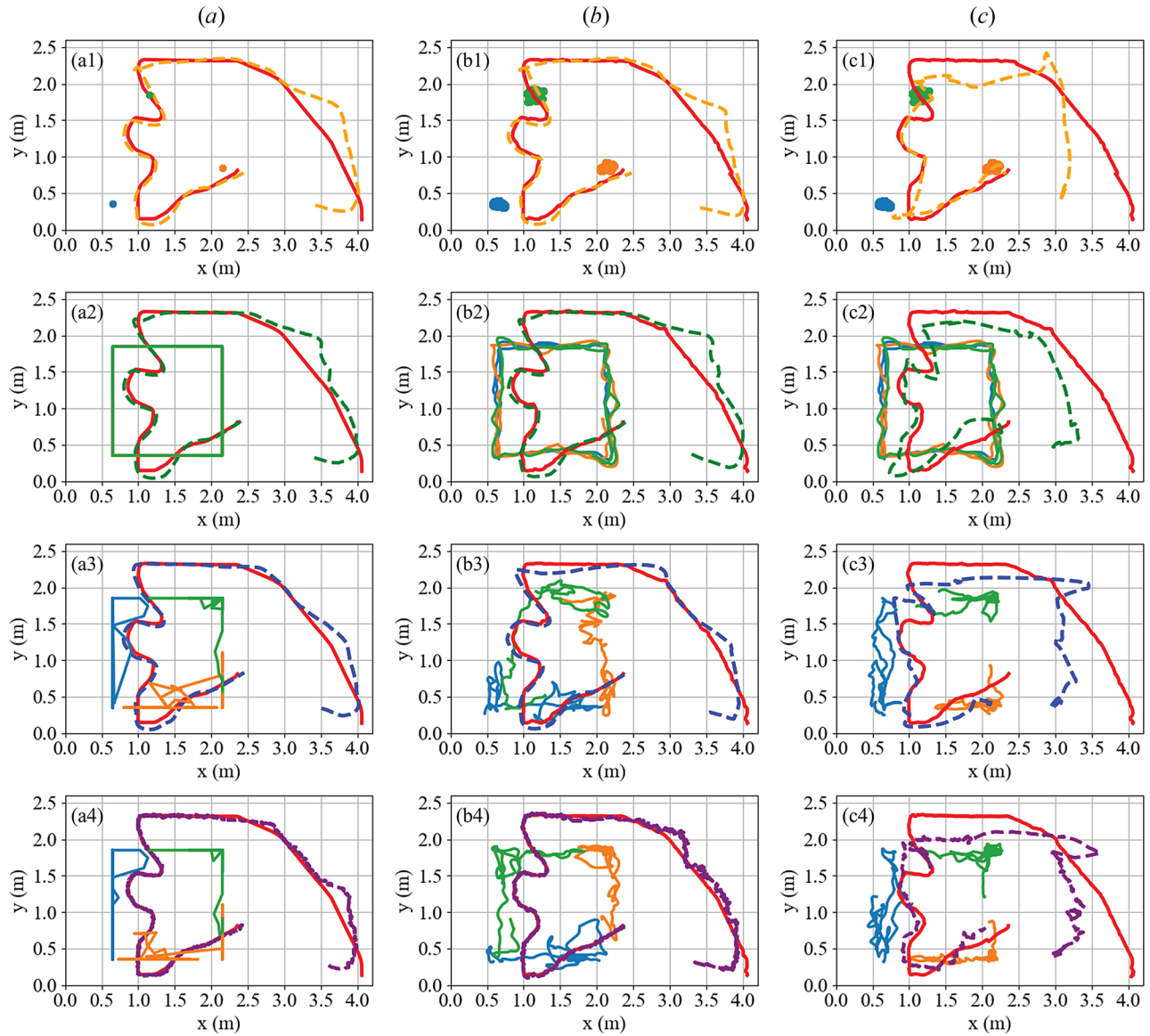


Fig. 15 Example of methods M1, M3, M6, and M7 (1–4, respectively) with three agents. Agent paths stay within the square-shaped constraint, the true target path is in the solid line, and the estimated target path is the dashed line. Simulations are in column (a). Physical experiments with a simulated sensor are in column (b). Physical experiments with a real sensor are in column (c).

target compared to the other five uninformed methods is confirmed in all simulation cases, except for when there are four agents. In this case, the static border case (M1) performed marginally better by 0.1% error. In this configuration, the static agents are located in the four corners of the search area, with maximum separation. In the information surface shown in Fig. 6(c), it is better to select agent combinations that are most separated. The static border (M1) and rotation (M3) methods maintain a large separation distance between the agents throughout the duration of the simulations. Agents stay on the edges of the search area with some degree of uniform distance between them. The performance of the IT1 algorithm (M6) compared to the static border method, in this case, is very promising. It illustrates that the IT1 algorithm can perform as well as an ideal scenario containing four static sensors. This means that if an ideal static sensor network does not exist, this algorithm autonomously realizes this configuration and yields behavior consistent with an idealized static sensor network. This can be contrasted with the static uniform, random, and raster uninformed methods which perform consistently worse. These uninformed methods do not encode this

notion of maximum separation between agents for optimum performance.

One seemingly strange behavior observed in the results is the increase in the normalized percent RMSE values for methods M1, M3, M6, and M7 when going from four to five agents and six to seven agents. This is related to the initial starting locations of the agents. The importance of the corners in measurement was discussed earlier. When transitioning from four to five agents, the starting configuration for these four methods changes the number of corners being occupied from four to one. When transitioning from six to seven agents, the starting configuration changes from two corners being occupied to one. The importance of these corner positions can be easily observed in Table 1, where the static border case for four agents (all of which reside in corners) performs better than five, six, and even seven agents. The impact of starting in a less ideal starting configuration, therefore, propagates into the normalized percent RMSE. It is also noted that this phenomenon does not occur in the M2, M4, and M5 cases of the uninformed methods. This is due to the fact that the initial positions and motion of each of these

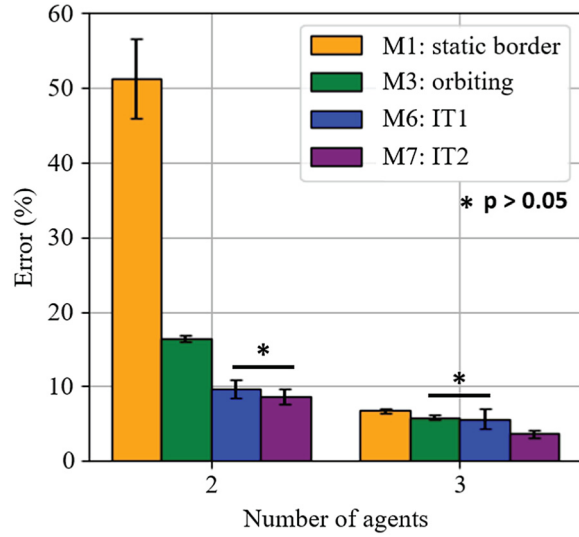


Fig. 16 The percent error for ten physical experiments when a simulated sensor with one percent error is used. Physical experiments are limited to two and three agents and methods M1, M3, M6, and M7.

Table 2 The mean and standard deviation of the normalized percent RMSE values for ten physical experiments with a simulated sensor and 1% error

Methods	Number of agents	
	2	3
M1: static border	51.15/5.29	6.75/0.30
M3: orbiting	16.38/0.37	5.85/0.29
M6: IT1	9.62/1.22	5.62/1.34
M7: IT2	8.65/1.07	3.69/0.49

Methods M1, M3, M6, and M7 are used in combination with two and three agents. The mean is reported first followed by the standard deviation.

agents are not tied to one another, or the corners in the way unlike the M1, M3, M6, and M7 cases. Finally, as expected, the IT2 algorithm performed better than all other methods for position estimation.

6.2 Physical Experiments With Simulated Sensor. The static border (M1) method performs practically identical to simulations in the three-agent case but slightly better than simulations in the two-agent case. This performance difference is due to the ability of the physical vehicle being able to hover in place with slight variations in its location as seen in Fig. 14(b1).

As seen in Fig. 16 and Table 2, in the two-agent case, the IT1 and IT2 algorithms performed exceptionally well compared to the static border and rotation (orbiting) cases. The performance of both IT1 and IT2 in this scenario is better than the simulations. In the three-agent case, the IT1 algorithm still performed better than the static border and rotation cases but not by as much as in the simulations; the IT2 algorithm has slightly better performance. These discrepancies are likely due to the dynamics of the aerial robots. In the rotation (M3) method, the path of the agents, as well as the estimated path taken by the target, is almost identical to simulations (see (a2) and (b2) in Figs. 14 and 15), and the agents do not have any dramatic changes in the direction of motion. In the IT1 and IT2 cases, however, dramatic changes in the direction of motion are common which cause the path taken by the agents to differ from simulations. This in turn causes the estimated target path to be different (see (a3) and (a4) versus (b3) and (b4) in Figs. 14 and 15).

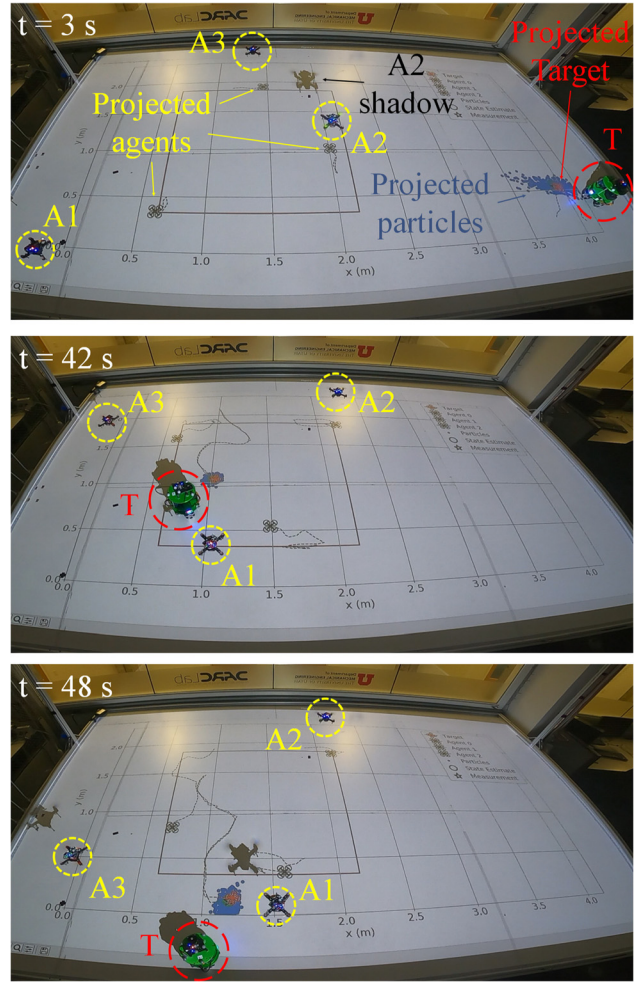


Fig. 17 Physical experiment for three instances in time using the IT2 algorithm with a simulated sensor. An image that shows the agents, the target, and the distribution of the particles is projected onto the platform surface.

Table 3 The mean and standard deviation of the normalized percent RMSE values are reported in the table below for physical experiments with a real sensor

Methods	Number of agents	
	2	3
M1: static border	42.35/3.60	20.63/0.59
M3: orbiting	30.03/1.49	18.88/0.67
M6: IT1	33.16/2.91	21.10/1.58
M7: IT2	34.12/3.52	20.64/1.32

The mean is reported first followed by the standard deviation.

6.3 Physical Experiments With Real Sensor. As expected, the physical experiments using a real sensor did not perform nearly as well as the simulated sensor. This was due to the nonlinear behavior and uncertainty of measurement of the real sensor as seen in Figs. 10 and 11. In Figs. 14(c1) and 14(c4), the two-agent case performed very poorly. For the three-agent case, the results depicted in Figs. 15(c1) and 15(c4), the algorithm had much more success. As shown in Figs. 10 and 11, the sensor error is not consistent, and therefore it is possible the poor performance of IT1 and IT2 compared to the rotation (M3) method is due to differing sensor

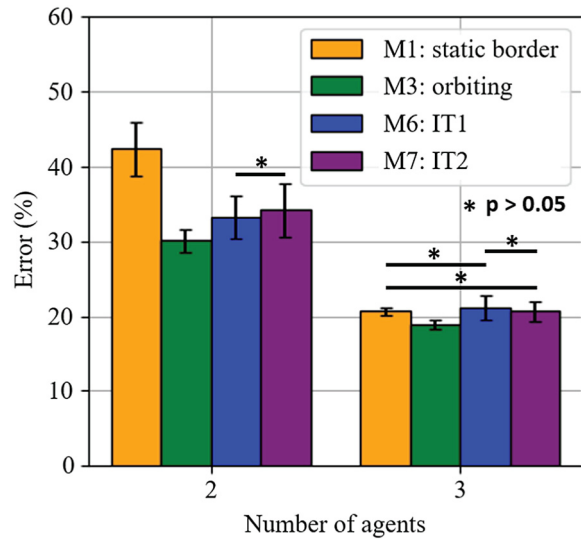


Fig. 18 The percent error for ten physical experiments when a real sensor with high error (assumed in algorithms to be 10%) is used. Physical experiments are limited to two and three agents and methods M1, M3, M6, and M7.

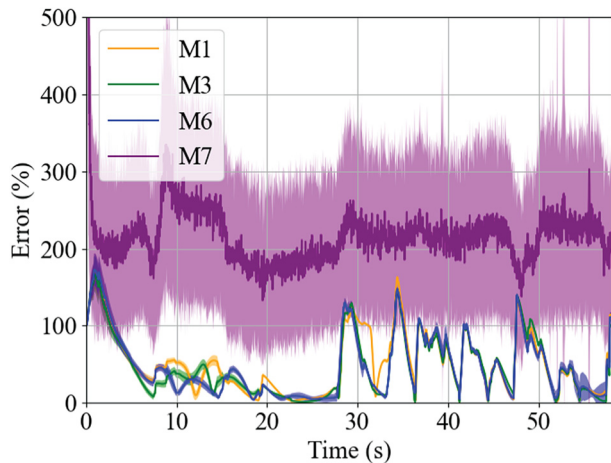


Fig. 19 Velocity percent error for the three agents in simulation. Despite M7 being the best position estimate method, the velocity percent error is far larger when the derivative of the position measurements is used to calculate the velocity for that method.

error. In the three-agent case, the only method that was significantly different than the others is the rotation method.

7 Conclusions

A novel method of uncooperative target localization and tracking using TDOA measurements with a particle filter and MI to autonomously guide the motion of the agents was developed, described, simulated, and validated in physical experiments. The algorithm performs better than state-of-the-art methods for localization and tracking using TDOA measurements for the two-agent case. The algorithm also performs as well as an idealized static sensor network, which would allow it to be deployed when no existing architecture is present. The algorithm has applications in search and rescue, surveillance, and military threat monitoring. Future work will (1) explore using the algorithm to enable a swarm of agents to autonomously follow/chase a dynamically moving target instead of being confined to a search area, (2) investigate the computational demands of the algorithm as it relates to the number

of agents being placed, and (3) incorporate other measurements such as FDOA or DRSS, and expand the scale of the experiments.

Acknowledgment

The paper/data have been reviewed in accordance with the International Traffic in Arms Regulations (ITAR), 22 CFR PART 120.11, and the Export Admin. Regulations (EAR), 15 CFR 734(3) (b)(3), and may be released without export restrictions.

Data Availability Statement

The authors attest that all data for this study are included in the paper.

References

- [1] MacDonald, V. H., and Schultheiss, P. M., 1969, "Optimum Passive Bearing Estimation in a Spatially Incoherent Noise Environment," *J. Acoust. Soc. Am.*, **46** (1A), pp. 37–43.
- [2] Ancker, C. J., 1958, "Airborne Direction Finding—The Theory of Navigation Errors," *IRE Trans. Aeronaut. Navig. Electron.*, ANE-5(4), pp. 199–210.
- [3] Crassidis, J. L., and Markley, F. L., 2003, "Unscented Filtering for Spacecraft Attitude Estimation," *J. Guid., Control, Dyn.*, **26**(4), pp. 536–542.
- [4] Choi, J., Ulbrich, S., Lichte, B., and Maurer, M., 2013, "Multi-Target Tracking Using a 3D-Lidar Sensor for Autonomous Vehicles," *16th International IEEE Conference on Intelligent Transportation Systems*, The Hague, Netherlands, Oct. 6–9, pp. 881–886.
- [5] Luo, R. C., Yih, C.-C., and Su, K. L., 2002, "Multisensor Fusion and Integration: Approaches, Applications, and Future Research Directions," *IEEE Sens. J.*, **2**(2), pp. 107–119.
- [6] Bourne, J. R., and Leang, K. K., 2019, "Bayesian Estimation of Snow-Avalanche Victim Pose: A Method to Assist Human and/or Robot First Responders to Quickly Locate a Buried Victim," *ASME Paper No. DSCC2019-8946*.
- [7] Bourne, J. R., Pardyjak, E., and Leang, K. K., 2019, "Coordinated Bayesian-Based Bio-Inspired Plume Source Term Estimation and Source Seeking for Mobile Robots," *IEEE Trans. Rob.*, **35**(4), pp. 967–986.
- [8] Bourne, J. R., Goodell, M., He, X., Steiner, J., and Leang, K. K., 2020, "Decentralized Multi-Agent Information-Theoretic Control for Target Estimation and Localization: Finding Chemical Leaks," *Int. J. Rob. Res.*, **39**(13), pp. 1525–1548.
- [9] Smith, D., and Singh, S., 2006, "Approaches to Multisensor Data Fusion in Target Tracking: A Survey," *IEEE Trans. Knowl. Data Eng.*, **18**(12), pp. 1696–1710.
- [10] Wang, H., Yao, K., and Estrin, D., 2005, "Information-Theoretic Approaches for Sensor Selection and Placement in Sensor Networks for Target Localization and Tracking," *J. Commun. Networks*, **7**(4), pp. 438–449.
- [11] Wang, Z., and Gu, D., 2012, "Cooperative Target Tracking Control of Multiple Robots," *IEEE Trans. Ind. Electron.*, **59**(8), pp. 3232–3240.
- [12] Hedley, M., and Zhai, Q., 2014, "Wireless Sensor Network Using Hybrid TDOA/RSS Tracking of Uncooperative Targets," *2014 International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Sydney, NSW, Sept. 7–10, pp. 385–390.
- [13] Carter, G., 1981, "Time Delay Estimation for Passive Sonar Signal Processing," *IEEE Trans. Acoust., Speech, Signal Process.*, **29**(3), pp. 463–470.
- [14] Chan, Y.-T., and Ho, K., 1994, "A Simple and Efficient Estimator for Hyperbolic Location," *IEEE Trans. Signal Process.*, **42**(8), pp. 1905–1915.
- [15] Fang, B. T., 1990, "Simple Solutions for Hyperbolic and Related Position Fixes," *IEEE Trans. Aerosp. Electron. Syst.*, **26**(5), pp. 748–753.
- [16] Ho, K., and Xu, W., 2004, "An Accurate Algebraic Solution for Moving Source Location Using TDOA and FDOA Measurements," *IEEE Trans. Signal Process.*, **52**(9), pp. 2453–2463.
- [17] Wang, G., Cai, S., Li, Y., and Ansari, N., 2016, "A Bias-Reduced Nonlinear WLS Method for TDOA/FDOA-Based Source Localization," *IEEE Trans. Veh. Technol.*, **65**(10), pp. 8603–8615.
- [18] Jin, B., Xu, X., and Zhang, T., 2018, "Robust Time-Difference-of-Arrival (TDOA) Localization Using Weighted Least Squares With Cone Tangent Plane Constraint," *Sensors*, **18**(3), p. 778.
- [19] Kovavisaruch, L., and Ho, K. C., 2005, "Modified Taylor-Series Method for Source and Receiver Localization Using TDOA Measurements With Erroneous Receiver Positions," *2005 IEEE International Symposium on Circuits and Systems (ISCAS)*, Kobe, Japan, May 23–26, pp. 2295–2298.
- [20] Lu, X., and Ho, K. C., 2006, "Taylor-Series Technique for Moving Source Localization in the Presence of Sensor Location Errors," *2006 IEEE International Symposium on Circuits and Systems (ISCAS)*, Kos, Greece, May 21–24, pp. 1075–1078.
- [21] Cameron, K. J., and Bates, D. J., 2018, "Geolocation With FDOA Measurements Via Polynomial Systems and RANSAC," *2018 IEEE Radar Conference (RadarConf18)*, Oklahoma City, OK, Apr. 23–27, pp. 0676–0681.
- [22] Chan, Y.-T., Hang, H. Y. C., and Ching, P.-C., 2006, "Exact and Approximate Maximum Likelihood Localization Algorithms," *IEEE Trans. Veh. Technol.*, **55**(1), pp. 10–16.
- [23] Drake, S. R., and Dogancay, K., 2004, "Geolocation by Time Difference of Arrival Using Hyperbolic Asymptotes," *2004 IEEE International Conference on*

- Acoustics, Speech, and Signal Processing, Montreal, QC, Canada, May 17–21, pp. 361–364.
- [24] Díez-González, J., Álvarez, R., Sánchez-González, L., Fernández-Robles, L., Pérez, H., and Castejón-Limas, M., 2019, “3D TDOA Problem Solution With Four Receiving Nodes,” *Sensors*, **19**(13), p. 2892.
- [25] Musicki, D., and Koch, W., 2008, “Geolocation Using TDOA and FDOA Measurements,” *2008 11th International Conference on Information Fusion*, Cologne, Germany, June 30–July 3, pp. 1–8.
- [26] Okello, N., Fletcher, F., Musicki, D., and Ristic, B., 2011, “Comparison of Recursive Algorithms for Emitter Localisation Using TDOA Measurements From a Pair of UAVs,” *IEEE Trans. Aerosp. Electron. Syst.*, **47**(3), pp. 1723–1732.
- [27] Huang, G. P., Zhou, K. X., Trawny, N., and Roumeliotis, S. I., 2010, “A Bank of Maximum a Posteriori Estimators for Single-Sensor Range-Only Target Tracking,” Proceedings of the 2010 American Control Conference (ACC), Baltimore, MD, June 30–July 2, pp. 6974–6980.
- [28] Fletcher, F., Ristic, B., and Musicki, D., 2007, “Recursive Estimation of Emitter Location Using TDOA Measurements From Two UAVs,” *2007 10th International Conference on Information Fusion*, Quebec, QC, Canada, July 9–12, pp. 1–8.
- [29] Li, Y., Hao, C., Li, M., He, L., Li, P., and Wan, Q., 2019, “Moving Target Tracking Using TDOA and FDOA Measurements From Two UAVs With Varying Baseline,” *J. Phys.: Conf. Ser.*, **1169**(1), p. 012013.
- [30] Li, Y., Wang, X., and Lu, X., 2020, “Combined Carrier Phase and Code Phase Passive Radiation Source Localisation Method,” *IET Radar, Sonar Navig.*, **14**(1), pp. 147–155.
- [31] Wu, P., Guo, Q., Zhang, X., and Bo, Y., 2014, “Maneuvering Target Tracking Using Passive TDOA Measurements,” *Proceedings of the 33rd Chinese Control Conference*, Nanjing, China, July 28–30, pp. 758–762.
- [32] Gustafsson, F., Gunnarsson, F., Bergman, N., Forsslund, U., Jansson, J., Karlsson, R., and Nordlund, P.-J., 2002, “Particle Filters for Positioning, Navigation, and Tracking,” *IEEE Trans. Signal Process.*, **50**(2), pp. 425–437.
- [33] Nordlund, P.-J., Gunnarsson, F., and Gustafsson, F., 2002, “Particle Filters for Positioning in Wireless Networks,” European Signal Processing Conference (EUSIPCO), Toulouse, France, Sept. 3–6, pp. 311–314.
- [34] Zhong, X., Tay, W. P., Leng, M., Razul, S. G., and See, C. M. S., 2016, “TDOA-FDOA Based Multiple Target Detection and Tracking in the Presence of Measurement Errors and Biases,” 2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Edinburgh, UK, July 3–6, pp. 1–6.
- [35] Kalman, R. E., and Bertram, J. E., 1960, “Control System Analysis and Design Via the “Second Method” of Lyapunov: II Discrete-Time Systems,” *ASME Trans. ASME J. Basic Eng.*, **82**(2), pp. 394–400.
- [36] Julier, S. J., and Uhlmann, J. K., 1997, “New Extension of the Kalman Filter to Nonlinear Systems,” *Signal Processing, Sensor Fusion, and Target Recognition VI*, Orlando, FL, Apr. 21–25, pp. 182–193.
- [37] Julier, S. J., and Uhlmann, J. K., 2004, “Unscented Filtering and Nonlinear Estimation,” *Proc. IEEE*, **92**(3), pp. 401–422.
- [38] Takabayashi, Y., Matsuzaki, T., Kameda, H., and Ito, M., 2008, “Target Tracking Using TDOA/FDOA Measurements in the Distributed Sensor Network,” *2008 SICE Annual Conference*, Chofu, Japan, Aug. 20–22, pp. 3441–3446.
- [39] Cho, J. A., Na, H., Kim, S., and Ahn, C., 2012, “Moving-Target Tracking Based on Particle Filter With TDOA/FDOA Measurements,” *Etri J.*, **34**(2), pp. 260–263.
- [40] Seo, H., Kim, H., Kang, J., Jeong, I., Ahn, W., and Kim, S., 2019, “3D Moving Target Tracking With Measurement Fusion of TDOA/FDOA/AOA,” *ICT Express*, **5**(2), pp. 115–119.
- [41] Hoffmann, G. M., and Tomlin, C. J., 2010, “Mobile Sensor Network Control Using Mutual Information Methods and Particle Filters,” *IEEE Trans. Autom. Control*, **55**(1), pp. 32–47.
- [42] Julian, B. J., Angermann, M., Schwager, M., and Rus, D., 2012, “Distributed Robotic Sensor Networks: An Information-Theoretic Approach,” *Int. J. Rob. Res.*, **31**(10), pp. 1134–1154.
- [43] Charrow, B., Kumar, V., and Michael, N., 2014, “Approximate Representations for Multi-Robot Control Policies That Maximize Mutual Information,” *Auton. Robots*, **37**(4), pp. 383–400.
- [44] Atanasov, N. A., 2015, *Active Information Acquisition With Mobile Robots*, Ph. D. dissertation, University of Pennsylvania, Philadelphia, PA.
- [45] Ramirez-Paredes, J.-P., Doucette, E. A., Curtis, J. W., and Gans, N. R., 2016, “Optimal Placement for a Limited-Support Binary Sensor,” *IEEE Rob. Autom. Lett.*, **1**(1), pp. 439–446.
- [46] Wang, H., Yao, K., Pottie, G., and Estrin, D., 2004, “Entropy-Based Sensor Selection Heuristic for Target Localization,” *3rd International Symposium on Information Processing in Sensor Networks, Association for Computing Machinery*, Berkeley, CA, Apr. 27, pp. 36–45.
- [47] Dogancay, K., 2012, “UAV Path Planning for Passive Emitter Localization,” *IEEE Trans. Aerosp. Electron. Systems*, **48**(2), pp. 1150–1166.
- [48] Hamdollahzadeh, M., Adelipour, S., and Behnia, F., 2016, “Optimal Sensor Configuration for Two Dimensional Source Localization Based on TDOA/FDOA Measurements,” 2016 17th International Radar Symposium (IRS), Krakow, Poland, May 10–12, pp. 1–6.
- [49] Zhao, Y., Li, Z., Hao, B., and Shi, J., 2019, “Sensor Selection for TDOA-Based Localization in Wireless Sensor Networks With Non-Line-of-Sight Condition,” *IEEE Trans. Veh. Technol.*, **68**(10), pp. 9935–9950.