Potential Game-Based Non-Myopic Sensor Network Planning for Multi-Target Tracking

**SU-JIN LEE[1,2], (Member, IEEE), SOON-SEO PARK[1], (Student Member, IEEE),
AND HAN-LIM CHOI [1], (Senior Member, IEEE)**

[1]Department of Aerospace Engineering and KI for Robotics, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
[2]Institute of Defense Advanced Technology of Research, Agency for Defense Development, Daejeon 34186, South Korea

Corresponding author: Han-Lim Choi (hamlimc@kaist.ac.kr)

ABSTRACT This paper presents a potential game-based method for the non-myopic planning of mobile sensor networks in the context of target tracking. The planning objective is to select the sequence of sensing points over more than one future time step to maximize information about the target states. This multi-step lookahead scheme aims to overcome getting trapped at local information maximum when there are gaps in the sensing coverage due to constraints of the sensor platform mobility or limitations in sensing capabilities. However, long-term planning becomes computationally intractable as the length of planning horizon increases. This paper develops a game-theoretic approach to address the computational challenges. The main contributions of this paper are twofold: 1) to formulate a non-myopic planning problem for tracking multiple targets in a potential game, the size of which increases linearly as the number of planning steps and 2) to design a learning algorithm exploiting the joint strategy fictitious play and dynamic programming, which overcomes the gaps in sensing coverage. The numerical examples of multi-target tracking demonstrate that the proposed method gives a better estimation performance than myopic planning and is computationally tractable.

INDEX TERMS Potential game, sensor network management, mutual information, non-myopic planning, multi-target tracking, wireless sensor networks, path planning.

I. INTRODUCTION

Mobile sensor networks have been successfully used to acquire information about the quantities of interest spread over large areas, including applications such as, monitoring spatial phenomena, mapping, and tracking targets [1]–[6]. The mobility of sensor platforms certainly contributes to expanding the sensing coverage and mission areas, but the associated constraints in mobility and sensor modalities necessitate the resolution of crucial decision making around where and when to sense taking into account the constraints. The waypoints of the sensor platforms need to be chosen to maximize information gain while satisfying all of the associated resource constraints.

Sensor planning schemes may be myopic or non-myopic in terms of time domain. When we consider a set of sensing locations for the next time step only, this approach is referred to as "greedy" or "myopic". This short-term management yields good performance in tasks such as sensor placements

for monitoring spatial phenomena [7], mobile sensor targeting for weather forecasting [2], [3], exploration path generation for simultaneous localization and mapping [8], [9], and sensor management for target tracking [5], [10]–[13]. However, there are some situations in which non-myopic strategies give poorer performance in the next step, but better estimation accuracy at the end of the planning horizon. In [6], [14], and [15], multi-step lookahead sensor planning was shown to significantly improve target tracking performance. Typically, when a sensor network contains "sensor holes", a greedy algorithm results in poor performance [14]. The sensor holes are gaps in the sensing coverage where the sensor network cannot make observations of the targets. These invisible areas are caused by the limited field of view of sensors or by obstacles existing in environments, such as elevation difference of surveillance regions or by physical constraints on the sensor platform motion. This is the case for tracking moving objects with such sensors of restricted

capability. Non-myopic planning can address the issue of sensor holes by looking multiple steps ahead in the future to extend candidate sensing regions.

This paper considers non-myopic planning for multi-target tracking. Previous work on sensor planning for target tracking can be found in [1] and [16]–[18], and for resource management for multi-function radars in [11] and [19]. Non-myopic sensor planning is challenging because the number of possible sensing sequences increases exponentially with the length of the planning horizon. Exponential explosion in computational time and memory usage follow for finding the optimal sensing sequence. Several planning algorithms have been proposed to reduce computational costs. A simple but sub-optimal strategy is a greedy method that determines sensing points in sequence, choosing the next location that provides the maximum information gain conditioned on the preceding sensing decisions [1]. Approaches to finding optimal solutions can be found in [16], [17], and [20]. The methods used approximate cost functions and reduce the computational burden by using pruning algorithms. Since sensor planning has the uncertainty of the underlying states, adaptive path planning problems have been formulated as a Partially Observable Markov Decision Process (POMDP) [21]. Reference [22] designed guidance algorithms for controlling unmanned aerial vehicles (UAVs) by solving a POMDP approximately. However, these more complex algorithms still suffer from computation cost, and thus can deal only with problems of limited size.

This work addresses computational challenge by formulating a non-myopic planning problem as a potential game. The method is extended from the potential game formulation that was proposed in the authors' earlier work [23], [24] for efficient selection of sensing points for the next time step only. To formulate an optimization problem into a potential game, we need to specify players, their respective actions, their local objective/utility functions, and the learning rules for the players [25]. In the game framework of [24], each sensor in a sensor network represents a player trying to maximize its utility function defined by the conditional mutual information. This local utility design leads to a potential game, with a global objective being the mutual information between the target states and the measurement variables to be taken at the next time step. In addition to our earlier work, more works exists, which use game theory for distributed sensor network management. For example, track selection for multi-target tracking in a multifunction radar network has been made through an anti-coordination game [11]. A radar in a network tries to maximize the overall tracking accuracy criterion and a best response based dynamics is presented to find an equilibrium. In [26], the sensor management assigns moving sensors to targets through a distributed sensor-based negotiation game. Each sensor makes a decision locally that maximizes its utility by negotiating with neighboring sensors. In [27], clusters of multistatic radars in a network select the optimal waveforms maximizing the signal-to-disturbance ratio through a potential game. The authors proved the

uniqueness of the Nash equilibrium using the discrete concavity property of the proposed game. However, such game-theoretic approaches also have the same computational cost issue when considering long-term planning, thus the studies handled short-term management only.

To yield an efficient potential game method for informative non-myopic planning, we modify the game framework found in our earlier work and propose a learning algorithm. In formulating a sensor network planning problem into a potential game, we take three steps: (i) Specifying the global objective function for multi-target tracking, (ii) formulating the optimization problem as a potential game, and (iii) designing a learning algorithm to find a Nash equilibrium of the potential game. First, mutual information is used as a global objective of a planning problem. The quantities of interest are set to the target states at the last time of the planning horizon to reduce the computational burden. Then, a potential game is formulated by defining a player as a sensor at each time step. A local utility is defined by using the marginal contribution rule [24], [28], and it is shown to lead to a potential game with the mutual information. Lastly, we propose a learning method to find a solution of the designed game, which is extended from the joint strategy fictitious play [29]. Dynamic programming is used in the first run of the learning algorithm in order to fill the gaps in sensing coverage caused by the limited sensing capabilities and constraints of platform mobility. Numerical studies on UAV coordination for optimal ground target tracking are presented to demonstrate that the game-theoretic mechanisms provide computational efficiency in sensor planning over multiple time steps.

II. PROBLEM FORMULATION

While the game-theoretic method of cooperative planning applies to many different applications with sensor networks, here we illustrate sensor planning in the context of multi-target tracking.

In this section, we define a multi-target tracking problem with a mobile sensor network. We then formulate a planning problem to find out the sensing locations to enhance tracking performance. First, the models of components comprising the target tracking problem are defined, and then a data fusion method for planning is described briefly to represent the conditional probability density function (pdf) of the target state conditioned on the measurements.

The goal of a target tracking problem is to estimate the kinematic states of targets with a finite set of measurements. In general, the state estimation is performed via Bayesian filtering [30], which is a recursive algorithm consisting of two processes: (i) Predicting the prior distribution of the target states using the dynamic model of the target and (ii) updating the distribution with the sensor measurement model and taken measurements. Therefore, specific models of target dynamics and sensor measurement are presented for the sake of concreteness. It should be noted, however, that the planning algorithms we introduce can be readily applied to other problems

in which sensors have a finite set of actions (i.e., sensing locations, sensor modes) to select.

A. TARGET AND SENSOR MODELS

We consider a number of unmanned aerial vehicles (UAVs) carrying sensors to track a group of targets on the ground, as shown in Fig. 1. To estimate the target states with measurements, a sensor network holds information about the targets using a motion model and prior distribution of the states.
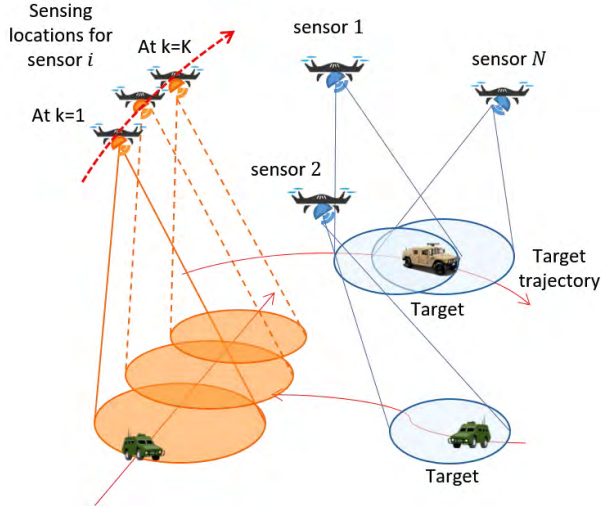

FIGURE 1. A sensor network planning problem for multi-target tracking: A number of targets moving on the ground are tracked by a sensor network. Every UAV (unmanned aerial vehicle) is carrying a sensor that observes the targets at a prespecified height.

Let a set of targets moving on the ground be $\mathcal{T} = \{1, 2, \ldots, M\}$. Here, we assume that the number of targets are known and constant, M. The state of target $j \in \mathcal{T}$ at time k is denoted by $\mathbf{x}_k^{(j)} = [x_k^{(j)}, y_k^{(j)}, \dot{x}_k^{(j)}, \dot{y}_k^{(j)}]^T$, where $x_k^{(j)}$ and $y_k^{(j)}$ represent the target position in two dimensional Cartesian coordinates, and $\dot{x}_k^{(j)}$ and $\dot{y}_k^{(j)}$ are the corresponding velocity components. We assume that each target follows continuous white noise acceleration models and moves independently from others. That way it is possible to ignore the joint distribution and use one tracking filter for each target [31].

$$\mathbf{x}_{k+1}^{(j)} = F_k \mathbf{x}_k^{(j)} + w_k^{(j)} \tag{1}$$

where $w_k^{(j)} \sim \mathcal{N}(0, Q_k^{(j)})$ is a white Gaussian process noise, independent of the other targets and measurements. F_k and Q_k are system transition and process noise covariance matrices, respectively. For the simulations in this paper, we use a continuous white noise acceleration model.

$$F_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$Q_k = \begin{bmatrix} \dfrac{\Delta t^3}{3} & 0 & \dfrac{\Delta t^2}{2} & 0 \\ 0 & \dfrac{\Delta t^3}{3} & 0 & \dfrac{\Delta t^2}{2} \\ \dfrac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \dfrac{\Delta t^2}{2} & 0 & \Delta t \end{bmatrix} q \tag{2}$$

where ΔT is the time between two successive measurements at kth step to $(k+1)$th step, and q is the process noise intensity, representing the strength of the deviations from predicted motion by the dynamic model [32]. When q is small, this model represents a nearly constant velocity. Note however, the actual maneuver of the targets can be different from the dynamics predicted by the sensor network. In the simulations, we will show the effect of incorrectly fitted models on the performance of planning algorithms.

A sensor network is represented by a set of vehicles $\mathcal{N} = \{1, 2, \ldots, N\}$, and each vehicle is assumed to be equipped with one sensor for simplicity.[1] For each sensor $i \in \mathcal{N}$, the measurement taken at time k is denoted in a general nonlinear form as

$$z_k^{(i)} = h_k^{(i)}(\mathbf{x}_k) + v_k^{(i)} \tag{3}$$

where $v_k^{(i)} \sim \mathcal{N}(0, R_k^i)$ is a white Gaussian noise process, independent of the other measurement noises and of process noise $w_k^{(j)}$, $\forall k$, $\forall j \in \mathcal{T}$. $\mathbf{x}_k$ is the set of states of targets and a sensor network. More specifically, in a target tracking problem, sensors usually measure the kinematic information about the target relative to the sensor, itself. Thus the state of the i-th sensor in $\mathbf{x}_k$ includes position, orientation, and velocity of the agent. Denoting the pose of i-th sensing agent at time k as $\mathbf{x}_k^{(i)_s}$, the measurement model can be rewritten to express the relative kinematic state of the target as

$$z_k^{(i)} = h(\mathbf{x}_k^{(j)}, \mathbf{x}_k^{(i)_s}) + v_k^{(i)} \tag{4}$$

For the simulations in this paper, we set the measurement model to a *radar-like measurement*, consisting of range and azimuth to a target. At each time k, sensor i obtains the positional measurement $\mathbf{z}_k^{(i,j)} = [r_k^{(i,j)}, \phi_k^{(i,j)}]^T$ for one of the targets, $j \in \mathcal{T}$:

$$\begin{bmatrix} r_k^{(i,j)} \\ \phi_k^{(i,j)} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + z_i^2} \\ \tan^{-1}((y_j - y_i)/(x_j - x_i)) \end{bmatrix}. \tag{5}$$

This is illustrated in Fig 2. For notational simplicity, the discrete time index k can be dropped later. The superscripts i and j represent the index of target and sensor, respectively, in the rest of this paper. As shown in Fig. 2, sensors have directivity, thus only the target inside the sensing region can be detected, and the sensor can obtain information about the location of the target within that area.

---

[1]In this paper, we use a sensing agent, a sensor, and a sensing node interchangeably. For simplicity, we assume that each vehicle has one sensor on it. In general, some platforms can carry more than one sensor. In that case, the sensing locations for the sensors equipped on the same vehicle should match. Then the decision variables are for the vehicles not for the sensors.
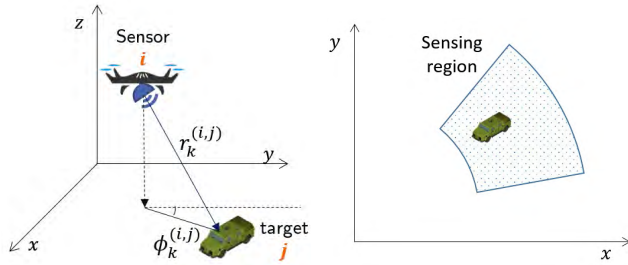
FIGURE 2. Radar-like measurement model for target tracking. (left) A sensor is mounted on a UAV and takes measurements consisting of range and azimuth to the target. (right) Sensors have a limited sensing region, in which a target can be detected and the relative position of the target can be measured.

In radars it is shown that the signal to noise ratio (SNR) dependent error is a major factor of the measurement error [33]. An azimuth angle error can be modeled as a white Gaussian noise with zero mean and standard deviation given by

$$\sigma_\phi \approx \frac{\theta_{BW} / \cos \psi}{k_m \sqrt{2SNR}} \qquad (6)$$

where θ_{BW} is the 3dB beamwidth in the angular coordinate and k_m is the monopulse pattern difference slope. In addition to SNR, angular measurement error increases when targets are located offset from the beam center due to a loss of gain [32], which is reflected in the term $\cos \psi$. ψ is the beam scan angle off-broadside. Similarly, a range error can be represented as zero-mean Gaussian noise with standard deviation given by

$$\sigma_R \approx \frac{\Delta R}{\sqrt{2SNR}} \qquad (7)$$

where ΔR is radar range resolution.

To make the problem simple, the three dimensional position and heading angle $\mathbf{x}^{(i)_s} = [x_i, y_i, z_i, \theta_i]^T$ of each sensing agent $i \in \mathcal{N}$ are assumed to be known. There is no uncertainty about the state of sensing locations. Thus, the information about the target state can be obtained by considering the probability distribution of the target and the induced distribution of measurement variables.[2]

B. ESTIMATION
Estimation of multi-target tracking is much harder than single target tracking, because the number of targets varies over time and the measurements need to be assigned to tracks. These issues have their own importance in multi-target tracking problems. However, in this paper, we aim at proposing a planning algorithm to maximize information about the states of moving targets. To focus on the algorithm that decides

the next sensing positions we assume that data association is known perfectly and that the number of targets to track is known and fixed. With the assumption that each target moves independently from the other targets, the single filter for each target is sufficient to develop the planning method.

The planning algorithm we propose can be applied with various estimation filters. Here, for each target j, the tracking process is performed using an extended Kalman filter (EKF). In the case of well defined transition models, the EKF has been considered to be a practical means of nonlinear state estimation [34]. Two types of targets are simulated for the simulation in this chapter: A nearly constant velocity model and a Dubins vehicle. When the sampling time is sufficiently small, Dubins car models fit into the target dynamics in (2) [15].

The EKF accomplishes the sequential estimation of the mean and covariance through two stages: Prediction and update, as follows [32]

- Prediction
  State estimate: $\mathbf{x}_{k|k-1} = F_{k-1}\mathbf{x}_{k-1|k-1}$
  State covariance: $P_{k|k-1} = F_{k-1}P_{k-1|k-1}F'_{k-1} + Q_{k-1}$
  Measurement: $\bar{\mathbf{z}}_{k|k-1} = h(\mathbf{x}_{k|k-1}, \mathbf{x}^s_k)$
  Innovation covariance: $S_k = H_k P_{k|k-1} H'_k + R_k$
- Update
  Innovation: $\tilde{\mathbf{z}}_k = \mathbf{z}_k - \bar{\mathbf{z}}_{k|k-1}$
  Filter gain: $W_k = P_{k|k-1} H'_k S_k^{-1}$
  State estimate: $\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + W_k \tilde{\mathbf{z}}_k$
  State covariance: $P_{k|k} = P_{k|k-1} - W_k S_k W'_k$

where $P_{k|k-1}$ and $P_{k|k}$ are the prior and posterior error covariance, respectively. $H_k = \frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_{k|k-1}}$ is the linearized measurement matrix evaluated at the predicted state $\mathbf{x}_{k|k-1}$. Since the state variables are assumed to follow the normal distribution, the induced variables from the dynamic model and the measurement model also follow normal distributions and the joint probability distribution of all the variables including the target states and measurements are denoted by multivariate Gaussian distributions. Therefore, the covariance update formula is a representation of the conditional covariance matrix given the measurement variables. In the next section, we will use the modified equation of covariance matrix considering the time step to calculate the amount of information contained in the measurement variables.

III. NON-MYOPIC SENSOR NETWORK PLANNING
The variables of interest in target tracking are the whole states of targets denoted by $X^{\mathcal{T}} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}\}$. The goal of sensor network planning is to find out the sequences of sensing locations for a sensor network such that information about the states of all targets is maximized. Therefore, the objective function can be the quantity of information about the targets' kinematics at the instant of interest. Sensor planning is called myopic when only the states at the next step are considered. While myopic planning has low computational costs and provides good performance in many examples [5], [7], [35], it performs worse than non-myopic scheduling in some cases.

---

[2]If the sensor state is represented by a probability distribution over the possible sensing locations, the distribution of the measurement variable can be obtained from the observation model. The distribution of the measurement variable usually spreads more widely than when the sensor state has no uncertainty, thus less information about the target states is contained in the measurement.

Reference [14] shows that a simple scenario with sensor holes can cause the performance degradation in myopic planning. To overcome the problem of greedy algorithm, a potential game based method that considers the change over a multi-step lookahead horizon will be provided.

A. INFORMATION-MAXIMIZING PLANNING FRAMEWORK

The goal of sensor planning for a target tracking problem is to select the sequences of sensing decisions for the next time steps that hold the most information about the target states. The information can be quantified by mutual information, which is the difference between the entropy of the target state and its conditional entropy conditioned on the measurement variables [36]. Entropy is the measure of uncertainty in a random entity. More specifically, (differential) entropy of a continuous random variable with probability density function $f_{\mathbf{x}}(x)$ is defined as

$$\mathcal{H}(\mathbf{x}) \triangleq -\mathbb{E}_{\mathbf{x}}[\log f_{\mathbf{x}}(x)] \tag{8}$$

where $\mathbb{E}_{\mathbf{x}}[\cdot]$ denotes expectations over random entity $\mathbf{x}$. The conditional entropy is defined as

$$\mathcal{H}(\mathbf{x}|\mathbf{z}) = \mathbb{E}_{\mathbf{z}}[\mathcal{H}(\mathbf{x}|\mathbf{z} = z)] \tag{9}$$

$\mathcal{H}(\mathbf{x}|\mathbf{z})$ is an expected entropy of the conditional distribution taken over all possible values of random variable $\mathbf{z}$. It represents a measure of uncertainty that will remain in $\mathbf{x}$ on the average before the specific value of $\mathbf{z}$ is given. From the definition of the conditional entropy we can compute the expected reduction in uncertainty of the target states before the new measurements are taken.

$$\mathcal{I}(\mathbf{x}; \mathbf{z}) = \mathcal{H}(\mathbf{x}) - \mathcal{H}(\mathbf{x}|\mathbf{z}) \tag{10}$$

As shown in Fig. 1, there are N sensors in a sensor network and M targets to track. We seek to maximize the information about the states of targets at K steps later by selecting the most informative sequences of sensing decisions over the next K time steps. The global objective of a sensor network can be represented by

$$\max_{Z_{1:K}^{\mathcal{N}}} \mathcal{I}(X_K^{\mathcal{T}}; Z_{1:K}^{\mathcal{N}}) \tag{11}$$

where $X_K^{\mathcal{T}} = \{\mathbf{x}_K^{(1)}, \ldots, \mathbf{x}_K^{(M)}\}$ is a set of random variables representing all of the multi-target states at the last time step of the planning horizon. $Z_{1:K}^{\mathcal{N}} = \{\mathbf{z}_{1:K}^{(1)}, \ldots, \mathbf{z}_{1:K}^{(N)}\}$ represents a sequence of predicted measurements to be taken by a sensor network $\mathcal{N}$ over a K-step horizon. All the variables here follow the probability distribution after updating with the previous measurement history, and the time step k of the subscript represents the relative time from the planning time. Thus, the random variables change with the time at which the planning decisions are made. Note that the variables of interest is the states at the final time of the planning horizon, not the sum of rewards resulting from each decision stage. References [16] and [17] adopted the additive cost also considering the cost at intermediate points in time. Even though

they showed good performance for tracking a single target, the summation could be computationally expensive for multi-target tracking. Therefore, we consider only the terminal reward of a sensor sequence as our objective to reduce the number of variables included in computing mutual information. Additionally, since the final expected target states reflect the intermediate condition of the targets, it is sufficient to consider the mutual information about the final states only.

The quantity of information that each measurement has about the targets depends on the sensing position and also the heading angle due to the directivity of sensors. Since a sensor can take one measurement at a time, we should decide which target is to be observed. Therefore, there are three decision variables for each measurement $\mathbf{z}_k^{(i)}$, $i \in \mathcal{N}$, $k \in \{1, 2, \ldots, K\}$, which are represented by a tuple: $a_k^{(i)} \triangleq [x, y, \theta, j]_k^{(i)}$. Here, $(x, y)_k^{(i)}$ denotes the sensing location for sensor $i \in \mathcal{N}$ at time step k. As shown in Fig. 3, the number of possible sensing locations at each time step is finite and the set of sensing candidates is denoted by $S_k^{(i)}$. For example, in the figure $|S_k^{(i)}| = 17$: stay at the previous location or move one of 2 distances in one of 8 directions. In the scenario for simulation, all of the sensors are supposed to fly at different heights and maintain their heights to prevent collision between agents, thus the z-coordinate is not a decision variable. For the heading angle of a sensor, there is a limit to the heading angle that can be rotated in one time step, and due to the limited field of view of sensors the heading angle is important for the success of taking measurements. When it is determined which target is observed, the best heading angle to the target can be decided from the current pose so that it does not exceed the limit of rotation. The heading angle is considered as a function of the current sensor's pose, next sensing the candidate location, and selecting the target's estimated position. Therefore, the pose of a sensor can be dropped out of the decision variables. The optimization problem (11) can be rewritten with the modified decision variables as

$$a_{1:K}^{\mathcal{N}\,*} = \arg\max_{a_{1:K}^{\mathcal{N}}} \mathcal{I}(X_K^{\mathcal{T}}; Z_{1:K}^{\mathcal{N}}) \tag{12}$$

where $a_{1:K}^{\mathcal{N}}$ is the set of $a_k^{(i)}$ for all $i \in \mathcal{N}$, $k \in \{1, 2, \ldots, K\}$. Each action for sensor i at time step k is denoted by $a_k^{(i)} = [(x, y), j]$ where $j \in \mathcal{T}$ and $(x, y) \in S_k^{(i)}$.
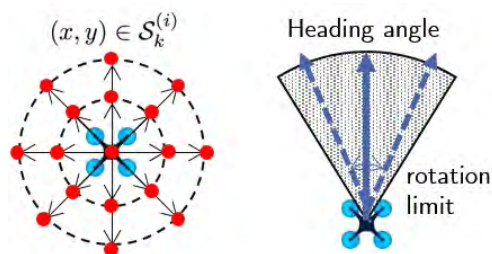

FIGURE 3. Possible actions for a sensor: Locations(left) and heading angle(right); the shaded area in the right figure is the field of view, and the two arrows are the limit of rotation of a UAV in one time step.

The above objective function for a multi-target tracking problem can be simplified when the targets are assumed to move independently of each other, because the state of each target becomes independent of other targets.

Lemma 1: If each target moves independently of each other, the objective function (11) can be rewritten as a sum of information between each target and the measurements that are taken from that target,

$$\mathcal{I}(X_K^\mathcal{T}; Z_{1:K}^\mathcal{N}) = \sum_{j \in \mathcal{T}} \mathcal{I}(\mathbf{x}_K^{(j)}; Z_1^{N_{j,1}}, \ldots, Z_K^{N_{j,K}}) \quad (13)$$

where $N_{j,k} \subset \mathcal{N}$ is a subset of sensors that decide to observe target j at time step k. $Z_1^{N_{j,1}}, \ldots, Z_K^{N_{j,K}}$ are the set of measurement variables that are taken for target j over the planning horizon of length K. Proof: *The above results can be easily induced by applying chain rule of mutual information [36].*

$$\mathcal{I}(X_K^\mathcal{T}; Z_{1:K}^\mathcal{N}) = \sum_{j \in \mathcal{T}} \mathcal{I}(\mathbf{x}_K^{(j)}; Z_{1:K}^\mathcal{N} | \mathbf{x}_K^{(1)}, \ldots, \mathbf{x}_K^{(j-1)})$$
$$= \sum_{j \in \mathcal{T}} \mathcal{I}(\mathbf{x}_K^{(j)}; Z_{1:K}^\mathcal{N})$$

Applying the chain rule to the measurement variables, the mutual information for each target becomes

$$\mathcal{I}(\mathbf{x}_K^{(j)}; Z_{1:K}^\mathcal{N}) = \mathcal{I}(\mathbf{x}_K^{(j)}; Z_1^{N_{j,1}}, \ldots, Z_K^{N_{j,K}})$$

This follows directly from the mutual information between independent variables being zero and the measurement variables from other targets also being independent. □

From the above lemma, the global objective can be obtained by considering information terms separately for each target.

B. POTENTIAL GAME FORMULATION

In [24], we presented a potential game approach for selecting the informative sensing points over the next time step and showed that the proposed method can achieve close-to-optimal solution quality by exploiting a systematic decision update procedure of a potential game. In this section, we will extend the game-theoretic framework for cooperative selection to non-myopic sensor planning that decides sensing points over several time steps. Before stating the potential-game formulation for non-myopic planning, a brief introduction to the basics of a potential game will be given first.

A general strategic form of a game consists of a finite set of players $\mathcal{P} = \{1, 2, \ldots, L\}$; each player $i \in \mathcal{P}$ has a finite action set $\mathcal{A}_i$ that the player can select, and has a preference structure over the actions according to its utility (payoff) function $u_i : \mathcal{A} \to \mathbb{R}$. $\mathcal{A} = \prod_{i \in \mathcal{P}} \mathcal{A}_i$ is a set of all possible combinations of actions for all players to choose at a time. $a = (a_1, a_2, \ldots, a_L) \in \mathcal{A}$ is the collection of strategies of all players, called a strategy profile, where $a_i \in \mathcal{A}_i$ denotes the strategy chosen by player $i \in \mathcal{P}$. For notational convenience, $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_L)$ denotes the collection

of actions of players other than player *i*. With this notation, a strategy profile is expressed as $a = (a_i, a_{-i})$.

A potential game is a non-cooperative game in which the incentive of the players changing their actions can be expressed by a single function, called the potential function. That is, that the player tries to maximize its utility is equivalent to maximizing the global objective [37].

Definition 1 (Potential Games): A finite non-cooperative game $\mathcal{G} = \langle \mathcal{P}, \{\mathcal{A}_i\}_{i \in \mathcal{P}}, \{u_i\}_{i \in \mathcal{P}} \rangle$ is a potential game if there exists a scalar function $\phi : \mathcal{A} \to \mathbb{R}$ such that

$$u_i(a_i', a_{-i}) - u_i(a_i'', a_{-i}) = \phi(a_i', a_{-i}) - \phi(a_i'', a_{-i}) \quad (14)$$

for every $i \in \mathcal{P}$, $a_i', a_i'' \in \mathcal{A}_i$, $a_{-i} \in \mathcal{A}_{-i}$. The function ϕ is referred to as a potential function of the game $\mathcal{G}$.

The property of a potential game in (14) is called perfect alignment between the potential function and the player's local utility functions. In other words, if a player changes its action unilaterally, the amount of the change in its utility is equal to the change in the potential function.

Potential games have two important properties due to the utility alignment [38]. The first one is that the existence of pure strategy Nash equilibrium is guaranteed. Since in a potential game the joint strategy space is finite, there always exists at least one maximum value of the potential function. This strategy profile maximizing the potential function locally or globally is a pure Nash equilibrium. Hence, every potential game possesses at least one pure Nash equilibrium.

Definition 2 (Nash Equilibrium): A strategy profile $a^* \in \mathcal{A}$ is called a (pure) Nash equilibrium if

$$u_i(a_i^*, a_{-i}^*) \geq u_i(a_i, a_{-i}^*) \quad (15)$$

for every $a_i \in \mathcal{A}_i$ and every player $i \in \mathcal{P}$.

The second important property concerns the dynamics of a game. A learning algorithm involves a process of finding out a Nash equilibrium by repeating a game. Many learning algorithms for potential games are established and have been proven to have guaranteed asymptotic convergence to a Nash equilibrium [39].

A potential game approach for sensor planning problems is to design the components of a game while satisfying perfect alignment [39]. In [24], we set each sensing agent as a player participating in a game and adopted a conditional mutual information as a local utility function. When the same player concept is applied to non-myopic planning game, it causes a degree of computational burden. In the case of L sensing allowable options for each step, there are a total of L^K distinct sensing sequences of length K, resulting in a total of $L^K \times M^K$ actions for each player. A participant then has an exponentially increased number of action strategies with the length of planning horizon. Thus, it is computationally intractable to calculate the utilities for all the different sensing selections. To address this problem, we set a sensor at each time step as a player, yielding a total of $N \times K$ players each having a strategy space $S_k^{(i)} \times \mathcal{T}$. If all the sensing agents have the same possible movement options, the number of actions for each
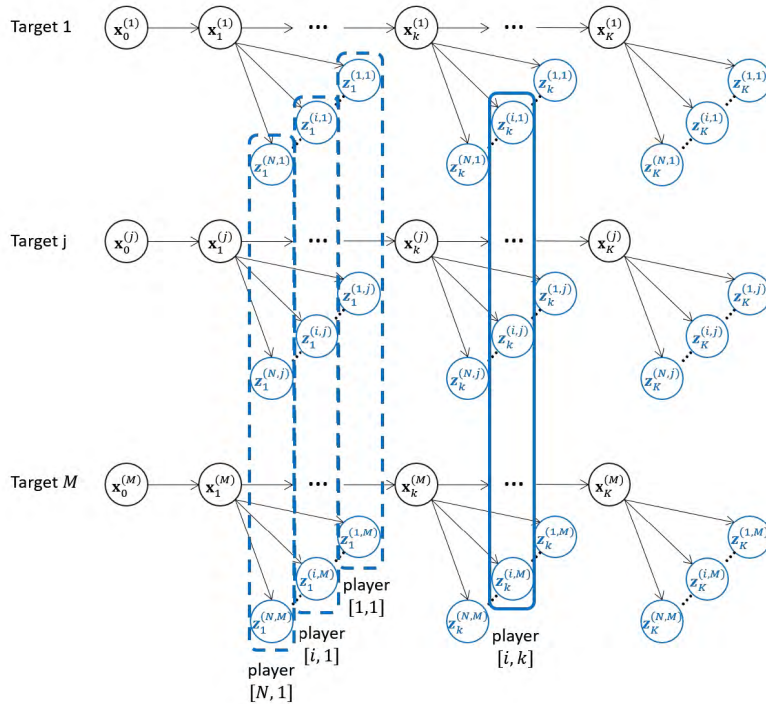
FIGURE 4. A graphical model with a proposed game design of a multi-target tracking problem in which each target moves independent of each of the targets. A rectangle represent a player in the potential game formulated for cooperative sensor network planning.

player to consider is constant with the number of time steps the algorithm looks ahead. Instead of an exponential increase in the number of actions for each player, this specification of players linearly increases the number of participants of a game. In summary,

- The players are the sensors at each time step represented by the set $\mathcal{P} \triangleq \mathcal{N} \times \mathcal{K}$, where $\mathcal{K} = \{1, \ldots, K\}$ is the set of time steps that the planning algorithm looks ahead. Each player is denoted by a 2-tuple $[i, k]$, where $i \in \mathcal{N}, k \in \mathcal{K}$
- The strategy space of each player is represented by the set $\mathcal{A}_k^{(i)} \triangleq S_k^{(i)} \times \mathcal{T}$ for each player $(i, k) \in \mathcal{P}$. The action which player $[i, k]$ can select is denoted by the tuple $a_k^{(i)} \triangleq [(x, y), j]$, where $(x, y) \in S_k^{(i)}$ is one of the possible sensing locations to which sensor i can move at time step k. The candidate positions are constrained by the sensor's moving capability such as maximum speed. The action set for player $[i, k]$ corresponds to the decision variables for measurement $\mathbf{z}_k^{(i)}$ as explained in the previous section.

A graphical model of the state and measurement variables and the players for a potential game formulation is depicted in Fig. 4.

As proposed in [24], the conditional mutual information conditioned on other players' decisions is considered as a

local utility for each agent

$$u_k^{(i)} = \mathcal{I}(X_K^{\mathcal{T}}; \mathbf{z}_k^{(i)} | Z_{-[i,k]}) \qquad (16)$$

where $Z_{-[i,k]}$ is the collection of measurements of players other than player $[i, k]$. As independence between the target states and the measurements from other targets simplifies the mutual information defined as a global objective, the local utility can be rewritten in a simpler form.

$$u_k^{(i)} = \sum_{j \in \mathcal{T}} \mathcal{I}(\mathbf{x}_K^{(j)}; \mathbf{z}_k^{(i)} | Z_{-[i,k]}) \qquad (17)$$

Since each sensor take measurements of one target at a time, the above expression for a local utility becomes

$$u_k^{(i,j)}(a_k^{(i)}, a_{-[i,k]}) = \mathcal{I}(\mathbf{x}_K^{(j)}; \mathbf{z}_k^{(i,j)} | Z_{-[i,k]}^{(j)}) \qquad (18)$$

which represents utility value when player $[i, k]$ selects target j for the next measurements. $Z_{-[i,k]}^{(j)}$ represents the measurement variables that are taken for target j among the collection of measurements, $Z_{-[i,k]}$. Using zero mutual information between two independent variables, the above expression can be easily derived. Therefore, it is sufficient to consider the measurements of sensors that select the same target when each player computes its utility values for a specific target selection over possible sensing locations.

Lemma 2: With local utility function defined as (17) the distributed procedure leads to a potential game with a global

objective (potential function) as

$$\phi(a_k^{(i)},\ a_{-[i,k]}) = \mathcal{I}(X_K^{\mathcal{T}}; Z_{1:K}^{\mathcal{N}})$$

Proof: Follows directly from [24, Lemma 1 and Proof of Lemma 1]. □

IV. LEARNING ALGORITHM FOR NON-MYOPIC SENSOR PLANNING

With the proposed utility function, the designed potential game is solved by a process called a *repeated game*, in which the same set of games is played over and over again while adapting the decisions of players to updated sets of information. At each play in a repeated game, every agent chooses an action according to a specific rule (such as fictitious play, better/best response and log-linear learning [40]), which is generally represented as a probability distribution over the player's actions set. In [24], we adopted the joint strategy fictitious play (JSFP) for finding the Nash equilibrium in a myopic sensor network planning problem formulated as a potential game. In this section, we will present learning dynamics that exploits the same learning framework as in our previous work. First, we review the basic concept of JSFP, summarizing the description of [24]. Next, we introduce the extended learning algorithm to non-myopic planning and analyze the computation complexity to perform the procedure.

A. JOINT STRATEGY FICTITIOUS PLAY

In the fictitious play based algorithms, each player presumes that opponents choose an action according to the empirical frequency of play. Particularly in JSFP, a player keeps track of the joint actions of others, and computes an expected utility based on the joint empirical frequency [29]. Let $f_{-i}(a_{-i}; t)$ be the frequency with which all players but i have selected joint action profile a_{-i} up to the t−th play. Then the expected utility for action $a_i \in \mathcal{A}_i$ is given by

$$u_i(a_i; t) = \mathbb{E}_{f_{-i}(t)}[u_i(a_i, a_{-i})]$$
$$= \frac{1}{t}\sum_{\tau=1}^{t-1} u_i(a_i, a_{-i}(\tau))$$

where $u_i(a_i, a_{-i}(\tau))$ is the utility computed for $a_i \in \mathcal{A}_i$ and the joint actions of others $a_{-i}(\tau)$ selected at the previous play τ. Marden *et al.* [29] gave a the simple expression for the expectation $u_i(a_i; t)$ as

$$u_i(a_i; t) = \frac{t-1}{t} u_i(a_i; t-1) + \frac{1}{t} u_i(a_i, a_{-i}(t-1))$$

This recursion form reduces computation complexity significantly compared to the traditional fictitious play based on empirical frequencies of marginal actions [24]. Also, with some notion of inertia, that is, a probabilistic reluctance to change actions, JSFP was shown to converge to a pure strategy Nash equilibrium [29, Th. 2.1].

B. LEARNING ALGORITHM WITH JSFP

In [24], we applied the JSFP algorithm to obtain a solution of a myopic sensor planning problem. The previously proposed JSFP method was simple in that the method does not need to consider a kinematic constraint of a mobile sensor over multiple time steps. It is sufficient to consider the set of next possible sensing locations only, which is unchanged during a repeated game. However, when considering the path over more than one time step, the k-th possible sensing locations and heading angles are subject to the previous and next pose of the sensing platform. Therefore, we should reflect changing reachable candidates at each time step in a learning algorithm.

Another challenge in non-myopic planning is sensor holes. For example, when a sensor has a limited field of view and all targets are located out of view at the planning time and in one time step later, a greedy method can get trapped at the local maximum [14]. Extending the candidate region to the K-step reachable region, this sensor hole problem can be addressed. Since a mobile platform can reach any location inside the farthest points reachable by K movements, it is enough to define the set of possible sensor configuration options for the Kth step candidates as the region including the farthest points and inside that boundary.

A non-myopic learning algorithm to be presented starts a repeating game from the last time step K. Player $[i, K]$ for each $i \in \mathcal{N}$ first selects the optimal position as an initial action within a K-step reachable region by solving the local greedy method. This initialization scheme continues to $(K-1)$th step with conditioning pdf of variables conditioned on the Kth decisions. After converging to the solution for each time step, the algorithm goes back to the previous time step until the one step lookahead decision. This initialization planning method is similar to dynamic programming.

In Algorithm 1, the learning algorithm for non-myopic planning is summarized. There are two `while` loops: Outer and inner. The inner loop corresponds to JSFP for a potential game considering k-th step only. The inner loop terminates according to the following rules:

- If JSFP converges to a Nash equilibrium, the decisions for k-th step are made with those optimal sensing locations.
- Otherwise, the loop terminates with probability $1 - \beta$, $\beta \in (0, 1)$ (See line 12, where rand() generates a random number from the uniform distribution on the interval $(0,1)$).

where β represents the willingness to optimize at time step k. According to these rules, players try to optimize their decisions with probability β, and will stay with the previous action with probability $1 - \beta$ at each repeated game. This nonzero inertia term is required to avoid cycling in local searching [29]. After the inner loop terminates, the selected decisions are stored in $N_{j,k}$'s, which are used for updating the covariance matrix. The outer loop is performed recursively until all of the decisions are not changed over a repeated

---

Algorithm 1 JSFP-Based Double Loop Learning for Non-Myopic Sensor Network Planning

---

INPUT: target states: $\{\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(M)}\}$, state covariances: $\{P_0^{(1)}, \ldots, P_0^{(M)}\}$, sensors' pose: $\{\mathbf{x}_0^{(1)s}, \ldots, \mathbf{x}_0^{(N)s}\}$, $\alpha \in (0, 1)$

1: Predict target states and covariances $\{\mathbf{x}_{1:K|0}^{(1)}, \ldots, \mathbf{x}_{1:K|0}^{(M)}\}, \{P_{1:K|0}^{(1)}, \ldots, P_{1:K|0}^{(M)}\}$
2: Initialize $N_{j,k} = \emptyset, \forall j \in \mathcal{T}$, for $k \in \mathcal{K}$
3: Convergedout = FALSE
4: **while** $\neg$ Convergedout **do**
5:   **for** $k \in \{K, \ldots, 1\}$ **do**
6:     $S_k^{(i)}$ = generate sensing locations at time step k, for each $i \in \mathcal{N}$
7:     Update covariance matrix $\{P_{K|N_{1,-k}}^{(1)}, \ldots, P_{K|N_{M,-k}}^{(M)}\}$
8:     Initialize $\bar{u}_k^{(i,j)}$ to zero vector
9:     Exitin = FALSE
10:    $t := 0$
11:    **while** $\neg$ Exitin **do**
12:     **if** rand() $> \beta$ **then**
13:       Exitin = TRUE
14:     **end if**
15:     $t := t + 1$
16:     **for** $i \in \mathcal{N}$ **do**
17:       $u_k^{(i,j)}$ = compute utility values at time step k, for each $(x, y) \in S_k^{(i)}$ and $j \in \mathcal{T}$
18:       $\bar{u}_k^{(i,j)} := \frac{t-1}{t}\bar{u}_k^{(i,j)} + \frac{1}{t}u_k^{(i,j)}$
19:     **end for**
20:     **for** $i \in \mathcal{N}$ **do**
21:       $[(x, y), j]_{(i,k)}^* = \arg\max_{(x,y)\in S_k^{(i)}, j\in\mathcal{T}} \bar{u}_k^{(i,j)}$
22:       **if** rand() $> \alpha \wedge \neg$ Exitin **then**
23:         $a_k^{(i)} = [(x, y), j]_{(i,k)}^*$ update strategy to the optimal decision.
24:       **end if**
25:     **end for**
26:     Exitin = check convergence of inner game
27:    **end while**
28:    Update $N_{j,k}$ from $a_k^{(1)}, \ldots, a_k^{(N)}$
29:   **end for**
30:   Convergedout = check convergence of outer game
31: **end while**

---

game. Since the algorithm is a variant of JSFP, it can be proved that the algorithm with inertia almost surely converges to a pure strategy Nash equilibrium.

Theorem 1: Algorithm 1 with $\alpha \in (0, 1)$ and $\beta \in (0, 1)$ almost surely converges to a pure strategy Nash equilibrium of the potential game in Lemma 2, with consistent tie-breaking in all the arg max operations involved in the process.

We provide a proof of Theorem 1 in the Appendix. The structure of Algorithm 1 is similar to fading memory JSFP with inertia, because the utility values are not stored through an outer loop. The utility $\bar{u}_k^{(i,j)}$ is initialized to a zero vector every time an inner loop ends. Therefore, we will prove Theorem 1 by following a structure in [29, Proof of Theorem 3.1].

C. CALCULATION OF UTILITY FUNCTIONS

This section explains how to compute the utility values in Algorithm 1 and analyzes its computation cost. In the previous section, the local utility function for a multi-target

tracking problem is defined as (17). When the dynamics and measurement models are linear and Gaussian, the mutual information objective of a normal distribution relates to its covariance matrix only.

$$
\begin{aligned}
\mathcal{I}(\mathbf{x}_K^{(j)}; \mathbf{z}_k^{(i,j)}|Z_{-[i,k]}^{(j)}) \\
= \mathcal{H}(\mathbf{z}_k^{(i,j)}|Z_{-[i,k]}^{(j)}) - \mathcal{H}(\mathbf{z}_k^{(i,j)}|\mathbf{x}_K^{(j)}, Z_{-[i,k]}^{(j)}) \\
= \frac{1}{2}\log\left(\left|P(\mathbf{z}_k^{(i,j)}|Z_{-[i,k]}^{(j)})\right|\right) \\
- \frac{1}{2}\log\left(\left|P(\mathbf{z}_k^{(i,j)}|\mathbf{x}_K^{(j)}, Z_{-[i,k]}^{(j)})\right|\right)
\end{aligned}
\tag{19}
$$

where $P(\mathbf{z}_k^{(i,j)}|Z_{-[i,k]}^{(j)})$ and $P(\mathbf{z}_k^{(i,j)}|\mathbf{x}_K^{(j)}, Z_{-[i,k]}^{(j)})$ represent the conditional covariance matrices of $\mathbf{z}_k^{(i,j)}$ conditioned on $Z_{-[i,k]}^{(j)}$ and additional $\mathbf{x}_K^{(j)}$, respectively. When $P(\mathbf{x}, \mathbf{z})$ represents the covariance between $\mathbf{x}$ and $\mathbf{z}$, the conditional covariance matrix for a Gaussian can be computed as

$$
P(\mathbf{x}|\mathbf{z}) = P(\mathbf{x}) - P(\mathbf{x}, \mathbf{z})P^{-1}(\mathbf{x})P(\mathbf{z}, \mathbf{x}).
\tag{20}
$$

From the above equation, we need the covariance matrix relating three variables $\mathbf{x}_K^{(j)}$, $\mathbf{z}_k^{(i,j)}$ and $Z_{-[i,k]}^{(j)}$ to compute the utility (19). Here, $\mathbf{x}_K^{(j)}$ represents the *a priori* state estimate predicted at the planning time, then

$$\mathbf{x}_K^{(j)} = \mathbf{x}_{K|0}^{(j)} = F_K^{(j)} \mathbf{x}_0^{(j)}$$

For notational simplicity, we will omit the superscript (*j*) specifying the target in this section, then all the variables here are for the single target *j*. Likewise, the measurement variables to be taken at time step *k* are induced from the a priori estimate $\mathbf{x}_{k|0}^{(j)}$. From target dynamics (1) and measurement model

$$\mathbf{x}_k = \mathbf{x}_{k|0} = F_k \mathbf{x}_0 + w_k$$
$$\mathbf{z}_k^{(i)} = h(\mathbf{x}_{k|0}, \mathbf{x}_k^{(i)}) + v_k^{(i)}$$

The measurement noise $v_k^{(i)}$ is a zero mean uncorrelated Gaussian process with covariance $R_k^{(i)}$ and is uncorrelated with any other sensor noise as well as the state variables and the process noise. The covariance matrices are computed as follows,

$$P(\mathbf{x}_k, \mathbf{x}_k) = F_k P_0 F_k' + Q_k \quad \text{for } k \in \mathcal{K}$$
$$P(\mathbf{z}_k^{(i)}, \mathbf{z}_k^{(i)}) = H_k^{(i)} P(\mathbf{x}_k, \mathbf{x}_k) H_k^{(i)'} + R_k^{(i)} \quad \text{for } k \in \mathcal{K}$$
$$P(\mathbf{z}_k^{(i)}, \mathbf{x}_K) = H_k^{(i)} P(\mathbf{x}_k, \mathbf{x}_k) F_{K-k}'$$
$$P(\mathbf{z}_k^{(i)}, \mathbf{z}_l^{(i)}) = H_k^{(i)} P(\mathbf{x}_k, \mathbf{x}_k) F_{l-k}' H_l^{(i)'} \quad \text{for } l < k$$

where F_{l-k} is the system transition matrix from time step *k* to *l* as defined in (2) with appropriate time interval between two time steps. The elements in the covariance matrix related to the target states $P(\mathbf{x}_k, \mathbf{x}_k)$ can be obtained prior to executing the learning algorithm for solving the potential game. Predicting the covariance matrices of the target states from $k = 1$ to $k = K$ requires $\mathcal{O}(K)$. There are two types of the measurement variables in (19). While $\mathbf{z}_k^{(i,j)}$ is the measurements for a sensing candidate at the current game, $Z_{-[i,k]}^{(j)}$ are the measurement variables selected in the previous game. Before starting each game in a repeated game, the terms related to the measurement variables can be computed and the resulting conditional covariance matrix $P(\mathbf{z}_k^{(i,j)} | Z_{-[i,k]}^{(j)})$ and $P(\mathbf{z}_k^{(i,j)} | \mathbf{x}_K^{(j)}, Z_{-[i,k]}^{(j)})$ are obtained. The maximum size of $Z_{-[i,k]}^{(j)}$ is $(N-1)(K-1)$. Since inversion of a $n \times n$ symmetric positive matrix requires $\mathcal{O}(n^3)$ flops and a determinant computation using Cholesky factorization needs $\mathcal{O}(n^3)$ flops [41], the resulting computation time for obtaining a set of utility values of one agent is $\mathcal{O}(N^3 K^3) + \mathcal{O}(L)$. After obtaining the conditional matrix, computing each utility value requires constant time, resulting in $\mathcal{O}(L)$ because the size of $\mathbf{z}_k^{(i,j)}$ is constant (2 for this scenario). In a repeated game, the prediction of the target states is done once, on the other hand, the conditional covariance matrix is computed for every game step. Therefore, the overall computing time is approximated as $\mathcal{O}(N^3 K^3)$ if *L* is fixed. Compared with myopic planning ($K = 1$), the calculation time increases polynomially to the length of the time horizon.

V. NUMERICAL SIMULATIONS

To compare the performance of a new algorithm with a greedy method, a scenario with sensor holes is presented. There are five targets moving on the ground. The map size is approximately 600×600. Three of the targets move straight with nearly constant velocity and change their speed sign when the target crosses the map boundary. The remaining two targets are modeled as Dubins vehicles that have a fixed forward velocity with a bounded turning radius. One example of this scenario is shown in Fig. 5 (red, green, blue: nearly constant velocity model. magenta, yellow: Dubins vehicles). The reason for there being two types of target movement is to see the performance difference between a well predicted model and a poorly matched model. A multi-step lookahead sensor planning is expected to outperform short-term methods in situations where the dynamics of targets are predictably changing, but to give poor performance in cases where the dynamic model is not matched to a real target dynamics. A target model applied to EKF is close to a nearly constant velocity model, while Dubins vehicles do not fit into that dynamics.
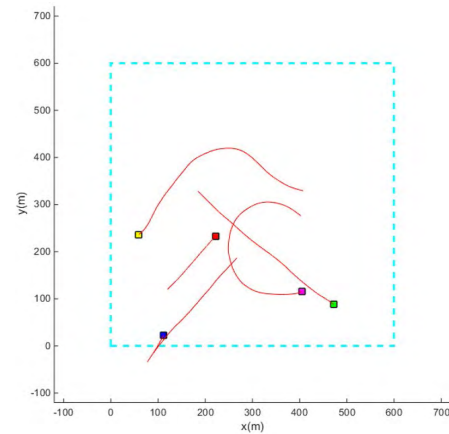

FIGURE 5. One example of a multi-target tracking scenario, in which (red, green, and blue) squares move in nearly constant velocity and the other two squares follow Dubins vehicle model.

For sensor platforms, we consider a set of quadrotors moving at different heights from each other to alleviate collision avoidance constraints. In a scenario for simulation, there are three platforms, each of which is equipped with one radar-like sensor. The sensors are mounted at a slight angle to look down. The standard deviations of measurement noise are depicted in Fig. 6, in which a sensor is located 500 high from the ground and $(x, y) = (0, 0)$. Figure 7 depicts the mutual information that can be obtained when a ground target is located with respect to the sensor. The coordinates of the sensor are $(0, 0)$, and the sensor is oriented towards the positive x-axis, as shown in Fig. 6.

The proposed non-myopic planning is realized in two ways as [16]. The first is open-loop planning (OL), in which the planning is performed after all the planned decisions are exhausted. The second is open-loop feedback (OLF), which
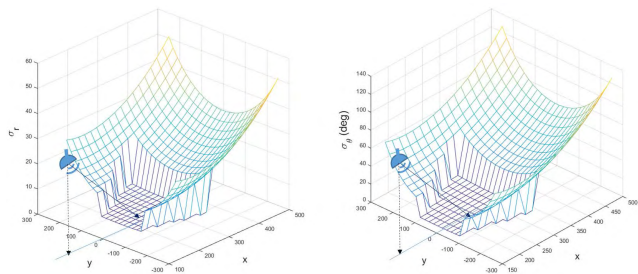
FIGURE 6. Measurement error term of a radar-like sensor. A sensor locates at (0,0, 490) and points to the positive x-axis slightly looking down.
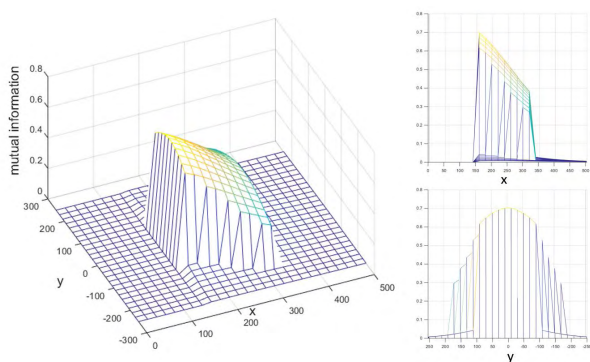

FIGURE 7. Mutual information with varying target location on the ground.

is the same as rolling horizon planning. In OLF, the forepart part of decisions is executed, and then a new plan for the following K steps is generated, having updated the covariance matrices and the state variables. To overcome the sensor hole problem and prevent the divergence of the filter, the decisions are made every two time steps and take the measurement of the same target as the previous one in the intermediate time step. Thus, $K = 2$ corresponds to myopic planning, which finds the optimal sensing actions in two steps and the same target is observed for two steps. We ran 100 simulations using both non-myopic planning schemes. For each run, the states of targets and the positions of the sensors were randomly initialized. Figures 8 and 9 present a comparison between myopic planning and two different non-myopic planning methods. As expected, when the target dynamics predicted the real target movements well, the performance
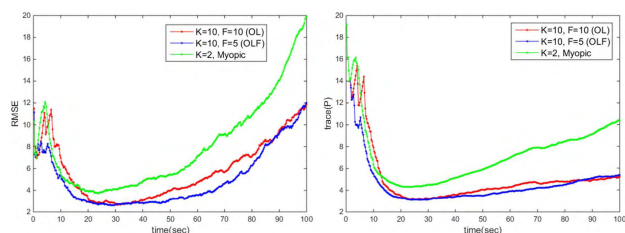

FIGURE 8. Comparison of the RMSE of the target position for nearly constant velocity target.
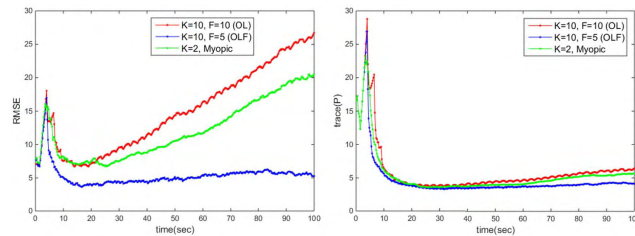

FIGURE 9. Comparison of the RMSE of the target position for Dubins vehicle target.

of non-myopic planning was better than myopic planning (See Fig. 8). On the other hand, when the dynamic model for the target state does not fit into the actual movement of the target, non-myopic planning (OL) gives poorer performance (See Fig. 9) than a greedy method. However, the open-loop feedback planning overcomes the nonlinearity of the target dynamics and gives the best performance of all the planning methods.

VI. CONCLUSIONS

In this paper, we investigated a potential game approach to long-term sensor network planning. Since a non-myopic optimization problem requires an exponentially increasing computation time as the length of the planning time horizon increases, a player for a potential game is needed to be defined differently from the myopic case in order to keep the size of a problem manageable. Accordingly, a new learning algorithm was proposed combining dynamic programming and JSFP. To demonstrate the performance of the non-myopic planning method, we considered a multi-target tracking problem involving sensor holes due to the constrained mobility of platforms and limitations on sensor visibility. The numerical simulations showed the effectiveness of the proposed approach and the conditions in which non-myopic planning gives better performance than myopic planning.

APPENDIX
PROOF OF THEOREM 1

According to [29], for the proof of convergence it is sufficient to show that there exists a non-zero probability, $\epsilon^* > 0$, such that the probability of convergence to an equilibrium by finite time t^* is at least ϵ^*. Since ϵ^* does not depend on stage t of a learning, this will imply that the action profile almost certainly converges to an equilibrium.

In [29], the proof of fading memory JSFP with inertia relies on the fact that if the current action profile is repeated a sufficient number of times (finite and independent of time) then a best response to the empirical frequencies becomes equivalent to a best response to the current action profile and hence will increase the potential provided that there is only a unique deviator. This will always happen with at least a fixed (time independent) probability because of the inertia. The outer loop of the proposed learning algorithm considers the most recent information only. That is, the outer structure of the learning rule is fading JSFP with inertia considering

the immediately previous decisions only. Thus, an outer loop with the same action profile does not need to be repeated. If there is a probability that one player changes its decision unilaterally to optimize its decision, the proof will be completed. Due to the parameter β and the inertia term α of JSFP in an inner loop, this unilateral best response will always happen with at least a fixed probability. Thus, the iterative procedure is proven to converge to a pure strategy Nash equilibrium almost surely as in [29, Proof of Theorem 3.1].

REFERENCES

[1] A. Singh, A. Krause, and W. J. Kaiser, "Nonmyopic adaptive informative path planning for multiple robots," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1–2.

[2] H.-L. Choi and J. P. How, "Efficient targeting of sensor networks for large-scale systems," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 6, pp. 1569–1677, Nov. 2011.

[3] H.-L. Choi and J. P. How, "Coordinated targeting of mobile sensor networks for ensemble forecast improvement," *IEEE Sensors J.*, vol. 11, no. 3, pp. 621–633, Mar. 2011.

[4] H. M. La, W. Sheng, and J. Chen, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 1, pp. 1–12, Jan. 2015.

[5] G. M. Hoffman and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 32–47, Jan. 2010.

[6] C. Kreucher, A. O. Hero, K. Kastella, and D. Chang, "Efficient methods of non-myopic sensor management for multitarget tracking," in *Proc. 43rd IEEE Conf. Decis. Control*, vol. 1, Dec. 2004, pp. 722–727.

[7] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, no. 2, pp. 235–284, Jun. 2008.

[8] H. Kretzschmar and C. Stachniss, "Information-theoretic compression of pose graphs for laser-based SLAM," *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1219–1230, 2012.

[9] Q. Yang and S.-J. Yoo, "Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms," *IEEE Access*, vol. 6, pp. 13671–13684, 2018.

[10] H.-L. Choi, J. P. How, and P. I. Barton, "An outer-approximation approach for information-maximizing sensor selection," *Optim. Lett.*, vol. 7, no. 4, pp. 745–764, 2013.

[11] N. Bogdanović, H. Driessen, and A. Yarovoy. (2016). "Track selection in multifunction radars for multi-target tracking: An anti-coordination game." [Online]. Available: https://arxiv.org/abs/1602.02032

[12] N. Farmani, L. Sun, and D. J. Pack, "A scalable multitarget tracking system for cooperative unmanned aerial vehicles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 4, pp. 1947–1961, Aug. 2017.

[13] H. Jiang and Y. Liang, "Online path planning of autonomous UAVs for bearing-only standoff multi-target following in threat environment," *IEEE Access*, vol. 6, pp. 22531–22544, 2018.

[14] J. Liu, D. Petrovic, and F. Zhao, "Multi-step information-directed sensor querying in distributed sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 5, Apr. 2003, p. V-145.

[15] C. Ding, A. A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, "Coordinated sensing and tracking for mobile camera platforms," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 5114–5119.

[16] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Nonmyopic sensor scheduling and its efficient implementation for target tracking applications," *EURASIP J. Adv. Signal Process.*, vol. 2006, no. 1, p. 031520, 2006.

[17] J. L. Williams, J. W. Fisher, III, and A. S. Willsky, "Approximate dynamic programming for communication-constrained sensor network management," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4300–4311, Aug. 2007.

[18] H. Oh, S. Kim, H.-S. Shin, A. Tsourdos, and B. White, "Coordinated standoff tracking of groups of moving targets using multiple UAVs," in *Proc. 21st Medit. Conf. Control Autom. (MED)*, Jun. 2013, pp. 969–977.

[19] A. Charlish, K. Woodbridge, and H. Griffiths, "Multi-target tracking control using continuous double auction parameter selection," in *Proc. 15th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2012, pp. 1269–1276.

[20] A. Logothetis and A. Isaksson, "On sensor scheduling via information theoretic criteria," in *Proc. Amer. Control Conf.*, vol. 4, Jun. 1999, pp. 2402–2406.

[21] D. A. Castanon, "Approximate dynamic programming for sensor management," in *Proc. 36th IEEE Conf. Decis. Control*, vol. 2, Dec. 1997, pp. 1202–1207.

[22] E. K. P. Chong, C. M. Kreucher, and A. O. Hero, III, "Partially observable Markov decision process approximations for adaptive sensing," *Discrete Event Dyn. Syst.*, vol. 19, no. 3, pp. 377–422, 2009.

[23] H.-L. Choi, "A potential game approach for distributed cooperative sensing for maximum mutual information," in *Proc. Amer. Control Conf. (ACC)*, Washington DC, USA, Jun. 2013, pp. 509–515.

[24] H.-L. Choi and S.-J. Lee, "A potential-game approach for information-maximizing cooperative planning of sensor networks," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2326–2335, Nov. 2015.

[25] R. Gopalakrishnan, J. R. Marden, and A. Wierman, "An architectural view of game theoretic control," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 3, pp. 31–36, 2011.

[26] D. Shen et al., "Game theoretic sensor management for target tracking," *Proc. SPIE*, vol. 7697, p. 76970C, Apr. 2010.

[27] A. Panoui, S. Lambotharan, and J. A. Chambers, "Game theoretic distributed waveform design for multistatic radar networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, pp. 1855–1865, Aug. 2016.

[28] J. R. Marden and A. Wierman, "Distributed welfare games," *Oper. Res.*, vol. 61, no. 1, pp. 155–168, 2013.

[29] J. R. Marden, G. Arslan, and J. S. Shamma, "Joint strategy fictitious play with inertia for potential games," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 208–220, Feb. 2009.

[30] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[31] P. Skoglar, *UAV Path Sensor Planning Methods for Multiple Ground Target Search Tracking—A Literature Survey*. Linköping, Sweden: Linköping Univ. Electronic Press, 2007.

[32] A. B. Charlish, "Autonomous agents for multi-function radar resource management," Ph.D. dissertation, Dept. Electron. Elect. Eng., Univ. College London, London, U.K., 2011.

[33] S. H. Kim, "Multiple-model based methods for target tracking using AESA radars," M.S. thesis, Dept. Aerosp. Eng., KAIST, Daejeon, South Korea, 2013.

[34] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Association*. Storrs, CT, USA: YBS Publishing, 2011.

[35] H.-L. Choi, J. P. How, and J. A. Hansen, "Ensemble-based adaptive targeting of mobile sensor networks," in *Proc. Amer. Control Conf.*, New York, NY, USA, Jul. 2007, pp. 2393–2398.

[36] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1991.

[37] D. Monderer and L. S. Shapley, "Potential games," *Games Econ. Behav.*, vol. 14, no. 1, pp. 124–143, 1996.

[38] O. Candogan, I. Menache, A. Ozdaglar, and P. A. Parrilo, "Flows and decompositions of games: Harmonic and potential games," *Math. Oper. Res.*, vol. 36, no. 3, pp. 474–503, 2011.

[39] J. R. Marden, G. Arslan, and J. S. Shamma, "Cooperative control and potential games," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1393–1407, Dec. 2009.

[40] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games*, vol. 2. Cambridge, MA, USA: MIT Press, 1998.

[41] B. S. Andersen, J. A. Gunnels, F. Gustavson, and J. Waśniewski, "A recursive formulation of the inversion of symmetric positive definite matrices in packed storage data format," in *Applied Parallel Computing* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2002, pp. 287–296.

SU-JIN LEE received the Ph.D. degree in aerospace engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2017. From 2004 to 2011, she was a Research Engineer with the Samsung Advanced Institute of Technology. She is currently a Senior Researcher with the Agency for Defense Development. Her research interests are in the areas of distributed optimization, game theory, and their applications in sensor networks.
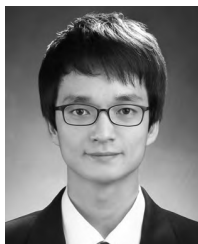
SOON-SEO PARK received the B.S. degree in aerospace engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2014, and the M.S. degree in aerospace engineering from Korea Aerospace University, Goyang, South Korea, in 2016. He is currently pursuing the Ph.D. degree in aerospace engineering with KAIST. His current research interests include path planning for information gathering and model-based reinforcement learning.

HAN-LIM CHOI received the B.S. and M.S. degrees in aerospace engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2000 and 2002, respectively, and the Ph.D. degree in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2009. He then studied for one-and-a-half years at MIT as a Post-Doctoral Associate, until 2010. In 2010, he joined KAIST, where he is currently an Associate Professor of aerospace engineering. His current research interests include estimation and control for sensor networks and decision-making for multi-agent systems. He, together with Dr. Jonathan P. How, was recipient of the Automatica Applications Prize, in 2011.

● ● ●