Survey: Mobile Sensor Networks for Target Searching and Tracking

Anders Albert
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
Email: anders.albert@ntnu.no

Lars Imsland
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
Email: lars.imsland@ntnu.no

***Abstract*—Mobile sensor networks can be employed in multiple applications, such as search and rescue, border patrol, battle scenarios, and environmental monitoring. In this survey, we review the literature utilizing mobile sensor networks in applications classified as target searching and/or tracking. Our contribution is threefold. First, we focus on the diverse types of filters applied to estimating the state of the targets. Second, we present the most common approaches to high-level trajectory planning for the sensors in the network to do target searching and/or tracking. Finally, we classify the literature based on the problem formulation used and solution characteristics. At the end of the survey, we discuss the current state of the literature and possible directions for future research efforts.**

I. Introduction

A. Mobile Sensor Network

A mobile sensor network (MSN) is *any group of agents where at least some have locomotion ability along with sensing and communication abilities*. The agents are normally referred to as sensor platforms or just sensors. A platform can be equipped with one or multiple sensors in addition to processing and communication units. An example can be a group of unmanned aerial vehicles (UAVs), each equipped with a camera used to monitor traffic [1]. In this paper, we will focus on controllable sensors. However, the sensor platforms do not need to be controllable. For example, humans in a city could serve as sensor platforms, and smart phones with environmental sensors could be utilized in an application to gather local weather data.

B. Target Searching and Tracking

Target searching and tracking (TST) *is the problem of estimating the location of one or multiple targets in a given area*. The area is referred to as the environment and can be an urban area, the sea surrounding a drilling platform, a forest, and so on. It can also contain obstacles, no-fly zones, hostiles, and so forth. The targets might be mobile or stationary and/or evasive or non-evasive. Examples of targets are people, icebergs, fire border (wildfire), and animals. There are numerous real-world applications, such as search and rescue, border patrol, battle scenarios, wildlife tracking, environmental monitoring, such as ice management, wildfire, and traffic, that can be classified as target searching and tracking [2, 3, 4, 5, 6, 7, 8]. The recent availability of cheap sensor platforms like Unmanned Aerial Vehicles (UAVs) make Mobile Sensor Networks relevant for these applications.

An MSN is well-suited for TST. The multiple sensor platform can spread out across the environment and move to make up for a limited field of view. A typical TST scenario is described in the 2D-plane. An example is illustrated with Figure 1, in which four sensors with communication constraints search for and track twelve moving targets.

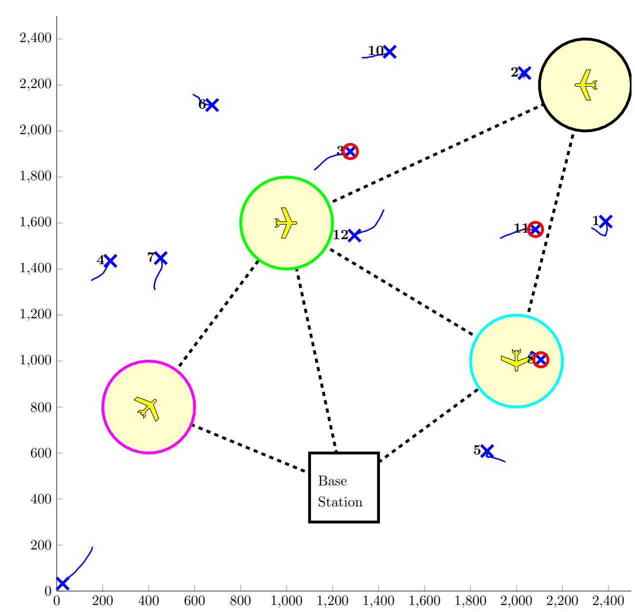

Fig. 1: Example Target Searching and Tracking scenario with four sensors and twelve targets. Each sensor is a plane-shaped yellow polygon with a light-yellow circle around to indicate its limited field of view (FOV). The color of the border of the FOV differentiates sensors from each other. The targets are marked with blue numbered X's and a short tail for recent movement. Three targets also have a red circle around them, indicating their estimated position. The base station is a black square, and the dotted black lines indicate the communication links between the sensors and base station.

C. Motivation

Sensors like UAVs are becoming cheaper and thus more available. There are already numerous companies offering UAVs for different applications. Most of these solutions require a human operator to control the UAV. However, to exploit the full potential of a UAV or multiple UAVs acting as sensors it is necessary for them to operate more autonomously. A human operator per sensor is less efficient than having the sensors report back only the interesting information to human operators. For a set of sensors to do high level task such as search for and track targets more advanced algorithms are necessary, which we will discuss in detail in this survey.

D. Contribution and literature appraisal

There are many options and challenges in using MSNs for target searching and tracking problems. Selecting an appropriate sensor, communication protocols, autopilot for each sensor platform are some of these. In most papers dealing with TSTs, the focus is usually on either high-level trajectory planning, the target state observer, or both. These are both algorithm designs and will be the focus of this survey. *Trajectory planning* is a high-level control algorithm which can be executed in a centralized or decentralized fashion, determining the path for each sensor platform. It is sometimes referred to as *path planning*. However, this term is also used for the problem of finding the path from A to B for an agent while avoiding obstacles. Another commonly used term is *Motion Planning*, but this term is even more general, which is why we will use the term trajectory planning. The *target state observer* is the estimation algorithm that uses the raw data from a sensor to estimate the state, usually the position and possibly velocity, of the targets. This is often referred to as a filter because it takes in raw measurements and aggregates them into, for example, location estimates. As far as the authors know, there have been no surveys on mobile sensor networks for the application of TST. We have limited the literature search to the combination of the phrases ”mobile sensor network” and ”target” in the title, keyword, or abstract of a paper. We have reviewed more than 300 papers, whereof approximately 100-150 have been included in this survey.

E. Previous Surveys

Several surveys have appeared with a focus on MSNs. The survey of Amundson and Koutsoukos [9] is centered around localization, which is the problem of estimating each sensor's position. Another survey is Zhu et al. [10], which focuses on communication and data management issues. The authors also discuss different applications, including target tracking. However, the discussion is limited to *single* target tracking. In this survey, we will also include multi-target tracking, as well as the issue of searching. In Ma and Tan [11], the authors discuss MSNs in an application context, and also include a section on TST. However, they mainly focus on the type of measurement device applied and separate searching and tracking into different categories. In contrast, we are also interested in the challenges that arise in the combination of searching and tracking. Another distinction is that we focus on high-level trajectory planning, and not the type of measurement devices. The survey conducted by Dagdeviren et al. [12] is on wireless sensor networks (WSN), but it also contains a section on MSN, identified as *mobile hardware agents*. Here the focus is on localization, communication, and energy harvesting.

In these surveys, the authors present taxonomies for distinct types of MSNs. In this paper, we are interested in MSNs in which we can actively control each sensor. In the terminology of the above papers, this is called *mobile actuated sensor networks* [9], *controllable movement* [10], and *controllable mobility for active sensing* [11].

The similar problem of target tracking has been well studied within the field of wireless sensor networks (WSN), and several surveys have been published [13, 14, 15, 16, 17]. Although target tracking with WSN's might have similar applications to target tracking using MSNs, this can be a very different approach. WSNs are usually decentralized, and the focus is on the communication protocol. Properties that are emphasized are energy consumption, communication range, bandwidth, limited storage and processing power. For structured WSNs sensor deployment is also a central task. All of these properties can be considered in MSN research. However, the sensor used for these networks is often expected to have more energy storage, processing power, and so on. The focus is typically on the mobility of the sensor and how to exploit this in target searching and tracking.

In addition to surveys on MSNs and WSNs there are several papers that deal with similar problems from other perspectives. From the robotics community, Robin and Lacroix [18] discuss the problem of target searching and tracking, which they call target management. The taxonomy the authors present starts by dividing searching and tracking into two different categories, as Ma and Tan has done [11]. Compared to our survey, we also consider target searching and tracking combined. In addition, Robin and Lacroix [18] is centered around the different ways TST problems are formulated, while our survey focuses on trajectory planning and target estimation. Another survey from the robotics community is Portugal and Rocha [19], in which the authors discuss the application of patrolling and take a graph theory approach. The survey is centered around control algorithms, and, in contrast to this paper, it focuses only on solving target search using graph theory. Here we will also include other approaches, such as continuous optimal control, gradient-based field, flocking control, and so on. A similar problem, pursuit-evasion, which is also from robotics, is studied by Chung et al. [20]. These are situations in which one or multiple searchers pursue one or multiple targets trying to avoid detection. The authors also discuss applications in which the targets are unaware of the sensors. For more information on literature dealing with just searching, see Stone et al. [21].

F. Organization

This survey is organized as follows. In Section II, we discuss the overall tasks of the observer and trajectory planner in search and track problems. We go into detail about the distributed and centralized architecture in Section III. Section IV contains the different filters for target state estimation, with Section V discussing trajectory planning algorithms. We classify the papers presented in the two previous sections in Section VI based on problem formulation and solutions characteristics. In Section VII, we discuss the current state of the literature and future work. A short conclusion is given in Section VIII.

G. Terminology

In the literature, there are many different terms used in the description of target searching and tracking. This partially stems from the many different approaches that are applied to these problems. Each row of Table I lists different terms used for the same concept, with the term used in this paper listed first and in bold. In addition, we will commonly use sensor to mean both the physical vehicle as well as its onboard sensor, processor, communication unit, and so on. However, in some contexts, such as in Section III, we will use sensor platform only in reference to the physical vehicle. Then, sensor refers to the unit producing measurements.

Terms
target, object
search, detection
tracking, monitoring, coverage
trajectory planning, path planning, motion planning, high-level control algorithm

TABLE I: Terms used to describe similar concepts in the literature. The term used in this paper is first and in bold.

II. Main components of Mobile Sensor Network for Target Searching and Tracking

We can divide a target search and track problem into two main tasks: observer and trajectory planning. The observer's task is to use sensor data to produce estimates of each target's state, which typically consists of position and velocity. It can also involve synchronizing estimates between multiple observers in a distributed approach, see Section III. The architecture of an observer depends on the number of targets. The most common approach is to use a filter to estimate the state of each target separately. This is illustrated in Figure 2. We will be discussing the different filters in Section IV. A problem that arises in multiple target tracking is to match measurements with the correct filter. This is called the data association problem and we will discuss it in Section IV-G. Another possibility is to combine all filters into one large filter, which is usually decoupled. This approach is rarely used, except in cases where the target's behaviors are actually coupled. Finally, the observer can also include a filter for estimating the state of the environment, called a map, which is also illustrated in Figure 2. This can include information about which areas have been explored, obstacle locations, and so on.

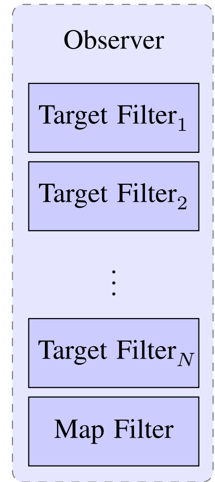

Fig. 2: Observer architecture. Typically, one filter is used for each target. In addition, sometimes a filter is used to track environmental information.

Trajectory planning is the task of utilizing the movement of an MSN for searching for and tracking targets. There are multiple factors that impact the design of a trajectory planner. First, it can be distributed or centralized. This will be discussed in more detail in Section III. Another factor is the number of targets versus the number of sensors. Here, there are typically three scenarios (we do not discuss single sensor and single target), which are all listed in Table II. When there are more sensors than targets, Scenarios 1 and 2, at least one sensor can be assigned to exclusively track each target. If there is available initial information about the location of the targets, the task is often exclusively to track the targets with the available sensors. If the positions of the targets are not known initially, but they are considered stationary, then the task is reduced to a search-only problem. However, in the case where there are fewer sensors than targets, Scenario 3, a trajectory planner has to balance the search for versus tracking of targets. In Section VI, we classify all the papers presented in this survey based on distributed/centralized approaches, the number of targets, as well as whether or not the trajectory planner considers tracking, searching, or a combination of both.

Scenario	Number of		
	targets		sensors
1	One	$<$	Multiple
2	Multiple	$<$	More
3	Multiple	$>$	Fewer

TABLE II: Different scenarios with respect to the number of targets vs sensors.

In addition, a trajectory planner must consider different types of constraints. For example, a fixed-wing, unmanned aerial vehicle cannot move sideways, which leads to sensor dynamic constraints. Furthermore, there might be obstacles in the environment, and the sensors must avoid collision, both of which are typically implemented as constraints.

III. Control Architecture

We separate between centralized and distributed approaches, which the control architecture mirrors. In a centralized approach, the sensor platforms are expected to be connected to the base station at all times. It is not necessary for each sensor platform to have much processing power as both the trajectory planning and data filtering can be done by the base station. However, in most cases the sensors platform might do preliminary aggregation of the measurements. For example, if the sensor is a camera, an onboard processor can do the image processing and send only the estimated position of the targets to the base station. Figure 3 illustrates a centralized approach.

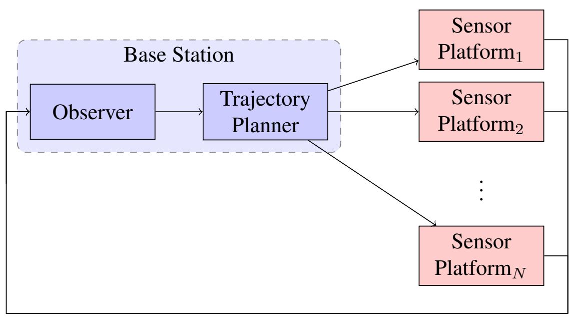

Fig. 3: Centralized architecture. The solid lines indicate constant connections. The arrows illustrate the way data travel.

In a decentralized approach, each sensor platform must perform both trajectory planning and measurement filtering onboard. In addition, they must coordinate with other platforms, while the base station tries to combine all available target observations from the sensor platforms. The focus of decentralized approaches is typically to have simple control algorithms, which do not exceed the computational power of the processing unit onboard each sensor platform. This approach is illustrated in Figure 4.

Whether to choose a centralized or decentralized architecture depends on the application. Often it is desirable to have a centralized architecture, as a decentralized architecture usually gives suboptimal solutions for the trajectory planning and makes it difficult to coordinate target measurements. In that case a decentralized might seem desirable only when no other option is available. However, there might be multiple reasons to choose a decentralized architecture. First, in the case of unreliable communication between sensors, a decentralized can take this into account and thus be more robust. Second, if there is a lag in communication between the base station and sensors, a decentralized will react faster to changes in the environment. Third, coordinating multiple sensors can become an intractable problem, depending on the planning algorithm,

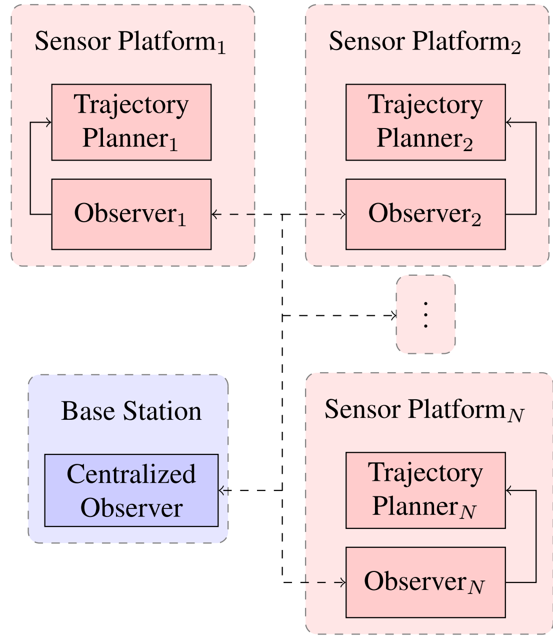

Fig. 4: Decentralized architecture. The solid lines indicate constant connections, while the dashed lines indicate partial connections. This means that sometimes they are connected and other times not. The arrows illustrate the way data travel. Notice that data travels both ways between all the observers.

and a decentralized architecture can be the only tractable option. Finally, the choice of architecture can come down to cost. Sensors with better communication range and more computing power is generally more expensive and require more power.

IV. Target State Filters

The two most popular algorithms for estimating the state of targets in mobile sensor networks are Kalman and particle filters. In this section, we will first present the basic idea for each filter. Then, we will discuss the distributed version of these filters, which are often applied in decentralized approaches. In addition, we will also briefly discuss some of the other filters applied in the literature.

A. Linear Kalman Filter

The Kalman filter is named after Rudolf E. Kálmán [22] and it is the optimal filter for linear systems with Gaussian noise [23]. The Kalman filter produces two outputs, a state estimate and an associated covariance matrix, which quantifies the uncertainty of the estimate. The filtering process contains two steps, executed recursively in time: First, in the a priori step, the previous estimate and covariance matrix propagate forward in time. In the second step, the a posteriori step, the new measurement is processed by combining it with the a priori estimate in a weighted average, depending on the uncertainty description of the new measurement and covariance matrix of the priori step. In addition, the covariance matrix is also updated.

We present the discrete version of the Kalman filter. Let x_k be the true state of a target at time k. Furthermore, let A be the transition matrix, and $w_k \sim \mathcal{N}(0, Q)$ be the process noise with zero mean and covariance $Q = Q^T \geq 0$. We combine this with the measurement, y, and its model, H, with noise $v_k \sim \mathcal{N}(0, R)$ also with zero mean and covariance $R = R^T \geq 0$, and get the following model and measurement equations

$$x_{k+1} = Ax_k + w_k \tag{1a}$$
$$y_k = Hx_k + v_k \tag{1b}$$

We use $\hat{x}$ to describe the state estimate and P the covariance matrix. To separate the a priori and posteriori step, we write $\hat{x}_{k|pri}$ and $\hat{x}_{k|post}$, and do the same for the covariance. The Kalman update equations are

$$\hat{x}_{k+1|pri} = A\hat{x}_{k|post} \tag{2a}$$
$$P_{k+1|pri} = AP_{k|post}A^T + Q \tag{2b}$$
$$\hat{x}_{k+1|post} = \hat{x}_{k+1|pri} + K_k(y_k - H\hat{x}_{k+1|pri}) \tag{2c}$$
$$P_{k+1|post} = P_{k+1|pri} - K_k H P_{k+1|pri} \tag{2d}$$

where

$$K_k = P_{k+1|pri}H^T(HP_{k+1|pri}H^T + R)^{-1}$$

Notice here that we do not include the initial conditions as these will vary for each case.

There are some centralized approaches which use the linear Kalman filter in target searching and tracking applications. Typically, one Kalman filter is used for each tracked target. For example, Cheng et al. [24], Prabhavathi and Rajeshwari [25] and Bai et al. [26] all utilize linear Kalman filters to estimate the target states. In addition, both Haugen and Imsland [27] and Albert et al. [28] use Kalman filters to estimate the state of moving icebergs with UAVs acting as a mobile sensor network.

While the covariance, R, of the measurement can usually be based on characteristics of the measurement device, the covariance, Q cannot. In all the above papers, the targets are simulated and the authors themselves selects the covariance, Q, of the process noise and assume it to be known to the filter. Usually, there is no discussion on how it should be obtained in a real-world example. Additionally, a linear Kalman filter also assumes a linear model, which might limit the applicability to a real-world example.

B. Extended Kalman Filter

The extended Kalman filter is a generalization of the non-linear target and measurement model. These models can be written as

$$x_{k+1} = f(x_k) + w_k \tag{3a}$$
$$y_k = h(x_k) + v_k \tag{3b}$$

where $f(x_k)$ is the transition function, and $h(x_k)$ is the measurement function.

This leads to a few changes is the Kalman update equations

$$\hat{x}_{k+1|pri} = f(\hat{x}_{k|post}) \tag{4a}$$
$$P_{k+1|pri} = A_k P_{k|post} A_k^T + Q \tag{4b}$$
$$\hat{x}_{k+1|post} = \hat{x}_{k+1|pri} + K_k(y_k - h(\hat{x}_{k+1|pri})) \tag{4c}$$
$$P_{k+1|post} = P_{k+1|pri} - K_k H_k P_{k+1|pri} \tag{4d}$$

where

$$K_k = P_{k+1|pri}H_k^T(H_k P_{k+1|pri}H_k^T + R)^{-1}$$
$$A_k = \frac{\partial f}{\partial x}\Big|_{\hat{x}_{k|post}} \qquad H_k = \frac{\partial h}{\partial x}\Big|_{\hat{x}_{k+1|pri}}$$

A well-known problem with the extended Kalman filter is its dependence on linearization, which can lead it to diverge if the initial state estimate is incorrect.

The extended Kalman filter is mainly used by mobile sensor networks to handle non-linear measurement models, (3b), while the target model, (3a), is usually linear. This is the case for both Ren et al. [29] (distance-to-target measurement model) and Wu et al. [30] (range-bearing sensor model). In Ren et al. [29] the authors also exploit the covariance matrix, P_k, in their control law. Martínez and Bullo [31] use a general target model in their derivation, and in simulation, use an 8-shaped movement for the target. All these papers use a centralized implementation of the extended Kalman filter.

The same critique of the linear Kalman filter applies to the extend Kalman filter, except that extended Kalman filer is able to utilize nonlinear models. In addition, the extended Kalman filter has several challenges. First, it is depended on the initial state estimate of the target for the estimate error to converge to zero. In Ren et al. [29], the authors use a sparsity decomposition scheme to initialize the targets to ensure the initial estimate error. InWu et al. [30] and Martínez and Bullo [31] there is less discussion on how to initialize the target positions. Second, in several of the mentioned papers the covariance matrix is used in the high-level trajectory planning as a measure of uncertainty of the estimate. Unfortunately, as the covariance is propagated linearly, equation (4b), in a nonlinear system this might be a poor measurement. Finally, in the extended Kalman filter the current state estimate is used to linearize the model. This means that if the current estimate is off, it can make the filter diverge.

C. Distributed Kalman Filter

In mobile sensor networks, it is often desirable to not have a centralized implementation of the observer, as this requires the network to always be fully connected. Olfati-Saber [32] has developed two distributed versions of the linear Kalman filter, which he calls the Kalman consensus filter and Kalman Information consensus filter. The first is optimal, but scales with $O(n^2)$, while the second is suboptimal and scales with $O(n)$, where n is the number of sensors.

Compared to the Kalman filter from equation (2), the a priori step stays the same, while the posteriori step changes.

Here, we only include the state update for a single sensor

$$\hat{x}_{i|post} = \hat{x}_{i|pri} + K_i(y_i - H\hat{x}_{i|pri}) + C_i \sum_{j\in N_i}(x_{j|pri} - x_{i|pri}) \tag{5}$$

We drop the subscript k, which indicates time, and instead use the subscripts i and j to note the current sensor and its neighboring sensors. This means that, for sensor i we have $\hat{x}_{i|post} = \hat{x}_{k+1|post}$ and $\hat{x}_{i|pri} = \hat{x}_{k+1|pri}$. In addition, the Kalman gain K_i of sensor i will also depend on the sensor's neighbors. Finally, we have the gain C_i which controls the trade-off between consensus of the sensors and stability of the filter. In Olfati-Saber [32], a suitable value is suggested, which provides global asymptotically properties.

Olfati-Saber applies the Kalman consensus filter to a mobile sensor network of 20 sensors in a single-target tracking problem [33]. He uses the estimation, together with a distributed flocking algorithm. This result was later extended to include multi-target tracking with a coupled estimation and flocking control algorithm [34]. Another similar distributed filter, also with a flocking algorithm, is Su et al. [35]. Here, the authors study the problem of tracking two coupled targets. The work on the distributed linear Kalman filter of Olfati-Saber has also been extended to include cubature Kalman filters and applied to mobile sensor networks for single-target tracking [36]. A cubature Kalman filter is a nonlinear filter for high-dimensional state estimation that exploits cubature points to numerically calculate multivariate moment integrals [37]. Another extension of Olfati-Saber's work is coupled estimation and flocking control for a single target with limited bandwidth in which only the position of the target is shared among the sensors [38].

Two other approaches to the distributed Kalman filter are by Giannini et al. [39] and Rigatos [40]. In Giannini et al. [39], the distributed Kalman filters use the inverse of the trace of the covariance to decide which local estimate to propagate though the sensor network. The authors of Rigatos [40] present a distributed version of both the extended and unscented Kalman filter (see Section IV-F for more about unscented Kalman filters).

The distributed versions of the Kalman and extended Kalman filters suffer from the same weaknesses as their centralized counterparts. An additional challenge is to make the local estimates to converge. As mentioned above it is possible to set the gain controlling the convergence, C_i in equation (5), such that filters converge globally asymptotically as proved by Olfati-Saber [32]. Unfortunately, this assumes an accurate knowledge of the process noise Q, which in general will not be available as discussed in Section IV-A. In the papers applying the Kalman consensus filters there is little or no discussion of how to obtain the process noise Q.

D. Particle Filter

Another popular filter in target tracking applications for mobile sensor networks is the particle filter. Unlike the Kalman filter, the particle filter does not require the process and measurement noise to be Gaussian distributed. This comes at the cost of increased computational complexity.

We will present a general version of the particle filter for a single target which will be based on Arulampalam et al. [41]. Both Kalman and particles filters are a form of Baysian filtering, but, while the Kalman filter represents the probability density function of the target state as a normal distribution, the particle filter approximates it using M number of particles. This means that the particle filter can use any form of distribution. If we let each particle be written as $\{\hat{z}^i, w^i\}$, where $\hat{z}$ and w are the state and weight, respectively, with i representing the particle number. We write the posterior probability density function for the linear Kalman filter and the particle filter together for easy comparison

$$p_{\text{kalman}}(z|y_k) = \mathcal{N}(z; \hat{x}_{k+1|post}, P_{k+1|post}) \tag{6a}$$

$$p_{\text{particle}}(z|y_k) \approx \sum_i^M w_k^i \delta(z - \hat{z}_{k+1}^i) \tag{6b}$$

$$\text{where} \quad \sum_i^M w_k^i = 1$$

where $\delta(\cdot)$ is the Dirac delta function, and z is the argument of the probability density functions for the Kalman and particle filters, which are noted as $p_{\text{kalman}}(z)$ and $p_{\text{particle}}(z)$. The notation for a normal distribution $\mathcal{N}(z; x, P)$ has argument z, mean x, and covariance matrix P.

The linear Kalman filter is updated with equation (2). When using a particle filter, each particle is sampled from an importance density function, $q(z)$, at each iteration. Normally, the importance density function is chosen to be the state transition function, equation (3a). Each particle's weight is updated by

$$w_{k+1}^i \propto \frac{p(x_{k+1})}{q(x_{k+1})} \tag{7}$$

where $p(x_{k+1}) \propto p_{\text{particle}}(z|y_k)$. The choice of importance density function is important for the performance of the algorithm.

A well-known problem with particle filters is the degeneracy problem. Eventually, all the particles will be very unlikely, which in practice means they have a very small weight. This can be measured through the variance of the weights. A normal measure of degeneracy is to calculate

$$N_{\text{eff}} = \frac{1}{\sum_i^M (w_k^i)^2} \tag{8}$$

A commonly implemented way to counteract the degeneracy problem is to resample the particles. Often a threshold is used in relation to equation (8) to decide when to resample the particles. In practice, this often means discarding the less likely and multiplying the more likely particles.

The particle filter is used both in single- and multi-target tracking by mobile sensor networks. A centralized approach for single-target tracking is implemented in Li and Djuric [42]. Here, the authors combine the particle filter with a Cramér-Rao Lower Bound to deploy mobile senors and compare it to using stationary sensors. The same authors also formulate a particle filter that does not require an assumed probability distribution for the process noise [43] for single-target tracking. This is based on the work by Míguez et al. [44], which substitutes the probability density function with a user-defined cost function that measures the quality of the state signal estimates according to the available observations. In Hoffmann and Tomlin [45], the authors also study single-target tracking. They develop a trajectory planning algorithm for minimizing the expected future uncertainty of the target state, in which they utilize the posterior probability available from the particle filter. The authors of Lu et al. [46] create a modified version of the particle filter for multi-target tracking, implemented as centralized and combined gradient-based approach for motion plans and control inputs. They use two modifications on the particle filter. First, instead of approximating the post probability distribution with a weighted sum of Dirac delta functions, they use a weighted sum of normal distributions:

$$p_{\text{particle}}(z|y_k) \approx \sum_i^M w_k^i \mathcal{N}(z; \mu_i, \sigma_i^2) \tag{9}$$

$$\text{where} \quad \sum_i^M w_k^i = 1$$

Second, they incorporate the newest measurement into the importance density function. This is done through using a target-state likelihood function, instead of a target-state transition function, equation (3a). Another similar approach for modifying the particle filter with a modified importance density function and resampling method is Juan-Yi [47]. In Hu and Tu [48], the authors use a combination of stationary and mobile sensors to track a single target. For target estimation they use a particle filter, which they modify for low energy consumption through parallel processing and better anti-noise capability.

The particle filter overcomes some of the limitations of the Kalman filters, like the assumption of Gaussian process and measurement noise, in addition to a linear model (compared to a linear Kalman filter). A draw-back is that it does not have an obvious covariance equivalent, which can be used for designing a tracking algorithm. There are a few alternatives to quanity the uncertainty of a target estimates. In Li and Djuric [42], the authors use a Cramér-Rao Lower Bound. Another option is to use the posterior probability directly like Hoffmann and Tomlin [45]. A third option is to construct a potential feedback function based on the particle filter for the controller Lu et al. [46]. Even though the particle filter can use any probability distribution, this distribution still has to be known. The authors of Li and Djuric [43] suggest a particle filter, where the process noise do not have to be known. However, this requires a user-defined risk function.

E. Distributed Particle Filter

As with the Kalman filter, it is sometimes necessary to a have a decentralized approach for the particle filter. One way to do this to organize the sensors in clusters and introduce some additional steps. After calculating the local posterior probability distribution, equation (6b), the weights, equation (7), are updated by the clusterhead (selected leader of the cluster of sensors) with

$$w_{k+1}^i = w_k^i \prod_{j \in \mathcal{N}_c} p(y_j|\hat{z}^i) \tag{10}$$

where w_{k+1}^i is the weight of particle i at time $k+1$. The set of neighbors to the clusterhead is denoted $\mathcal{N}_c$, which each have a measurement of the target y_j. The state estimate of particle i is denoted $\hat{z}^i$. Then, after normalization of the weights, the clusterheads resample and find the current estimate of the state. The result is then combined with the other clusterheads through diffusion. This is the approach used by Chen and Sezaki [49] for single-target tracking using a stationary sensor network. In Gu and Hu [50], the authors combine a flocking controller with a distributed particle filter for single-target tracking. Transporting all the particles between the nodes requires much bandwidth. Instead the authors use a Gaussian mixture model (GMM) learned from the weighted particles of all sensors through a distributed expectation maximization algorithm. This only requires that a few parameters be exchanged between the nodes. An expectation maximization (EM) is an iterative algorithm in two steps. First, a GMM is assumed and the likelihood for each particle is calculated based on the current GMM model. Second, the GMM model is updated based on the calculated particles. The first step is then repeated with the new GMM model. The algorithm stops when the GMM model has converged sufficiently. The global GMM model is found through an average consensus filter similar to the one presented in equation (5) by Olfati-Saber and Murray [51]. The algorithm by Olfati-Saber and his collaborators has also inspired the authors of Kan et al. [52] to handle single-target tracking using a mobile sensor network. They further develop the consensus algorithm for particle filters to also consider that some sensors do not observe the target. They demonstrate their algorithm using five sensors for a single target. An additional challenge is to combine distributed particle filters and multiple-target tracking. Particularly, this suffers from the curse of dimensionality, with the number of necessary particles getting multiplied by the number of targets. To meet this challenge, the authors of Beaudeau et al. [53] assume a linear state model for the targets, equation (1a), which enables them to approximate the posterior distributions with normal distributions, while they maintain a nonlinear measurement model, equation (3b). This enables them to reduce the required communication between the sensors, since each target state can be approximated by a mean vector and covariance matrix. For each time step, each sensor generates particles from a normal distribution from each of the other sensors. The newest measurement is then used to calculate each particle's weight according to equation (7). From the new posterior distribution, a new mean and covariance matrix is calculated based on the weights and particle states, which is then broadcasted to the other sensors.

The distributed particle have similar problems to distributed Kalman filters. In addition, a challenge with the particle filters is the exchange of estimates. Since each sensor approximates a probability distribution with a set number of particles, all particles must be exchanged between sensors to accurately exchange estimates. In comparison, a Kalman filter can be exchanged through one state estimate with a covariance matrix. It is natural to approximate the estimate of a particle filter like Beaudeau et al. [53]. However, in that paper the authors approximate the estimate with a Gaussian distribution, which partly make applying the particle superfluous since assuming a Gaussian distribution make the Kalman filter applicable.

F. Other Observers

In addition to Kalman and particle filters, there have been multiple other filters applied by authors using mobile sensor networks for target tracking. In this section, we will briefly present some of them.

An unscented Kalman filter is an attempt to improve the extend Kalman filter for non-linear systems. A problem with the extended Kalman filter is that the covariance matrix is propagated through a linearization, equation (4b) [54]. This is can lead to large estimation errors, especially for highly non-linear systems. The unscented Kalman filter use a set of sampling points to capture the true mean and covariance of the state estimate. This achieves improved accuracy over the extended Kalman filter, while keeping the computational complexity on the same order.Wang et al. [55] and Xie et al. [56] present a distributed version of the unscented Kalman filter for target tracking by a mobile sensor network. In addition, Wang et al. [57] present a distributed filter based on a more general unscented filter algorithm. Even though the unscented Kalman filter have a same order of magnitude as the extended Kalman filter when it comes to complexity, there is an increased complexity for the human implementing it.

Another estimation approach similar to Kalman filtering is H_∞. Here, instead of minimizing the $\mathcal{L}_2$-norm as done in the Kalman filter, the $\mathcal{L}_\infty$-norm is minimized [58]. In Nelson and Freeman [59], the authors present a distributed version of the H_∞ estimator to track a single target using an MSN. A disadvantage of the H_∞ filter, compared to the Kalman filter, is that it has more user set parameters. The same authors have also suggested a set-value estimation algorithm [60]. In contrast to the other filters, this estimator makes no assumption on the process and measurement noise except for a maximum value. Another advantage is a distributed version of the filer. It is straight-forward to use a union of sets from neighboring sensors without taking into consideration that a set should not be added multiple times (a union of a set with itself does not change the set). A problem with a set-value estimator is the increased storage capacity required for each iteration. To overcome this, the authors approximate the current set estimate with ellipsoids or parallelotopes. Another disadvantage is that since there is little assumption on the uncertainty of the estimates, there is not a covariance equivalent.

The federation filter is a distributed version of the Kalman filter designed to be computed in parallel [61] by local filters and fused by a master filter. In Jin et al. [38], each sensor acts as a master filter, with neighbors as local filters, in a single-target tracking application. This is similar to Olfati-Saber [32] and have the same problem of needing an accurate estimation of the process noise, Q.

In the case of a linear system with no disturbance, the Luenberger observer is well-suited. This works best for theoretical applications such as Wang et al. [62], in which a feedback controller for single-target tracking utilizes a Luenberger observer. However, it should not be applied in practice since it does not take into account measurement or process noise.

The authors of La and Sheng [63] present a consensus algorithm for estimating a scalar field used for navigation by a flocking algorithm. Each sensor keeps a local version of the scalar field, and the authors present an approach on how to update the field. To estimate the target's position the probability distribution is assumed known.

G. Multi-Target Tracking - Data Association

A problem which is often ignored in multiple-target tracking for MSNs, is *to match measurements to filter*. This is often referred to as the *data association problem*, but it can also be called the *multiple-target tracking problem* or *multiple sensor (data) fusion*. To avoid addressing this problem, the targets can either be assumed to be sufficiently far from each other, such that each observation can be matched with its closest filter, or that each target has a special characteristic making it trivial to match it to the correct filter. However, in general, this is not a trivial problem, and it becomes especially prevalent when we do not have continuous observations of each target.

Data association is a well-studied subject, and our goal here is not to give a comprehensive survey of the literature. Rather, it is to give a brief introduction through some surveys dealing with the problem of target tracking. The remainder of this section is based on the following surveys: Blackman [64], Pulford [65], Mallick et al. [66], and Qiu et al. [67].

Classical data association can be divided into recursive and batch approaches. A recursive approach has irreversible state updates and is typically less computationally expensive. The perhaps most straightforward approach is the nearest-neighbor algorithm. Here measurements are paired with the closest estimate, and it works well when the targets are separated in space and process noise is small. Closeness is typically measured in Euclidean distance, but the signal-to-noise ratio can also be used in cases with much clutter (false-positive measurements). An alternative approach is the all-neighbor algorithm, in which all measurements in the vicinity are used to update the target estimate. This is done by calculating the probability for each measurement and using it in a weighted average to update the target estimate. This approach is also referred to as (joint) probabilistic data association. Another algorithm is the global nearest neighbors. This is a minimal pairing with a set of measurements for a set of target estimates. It can be solved efficiently by the Hungarian algorithm [68].

Filter	Assumptions	Draw-backs	Advantages
Kalman Filter	Linear model, Gaussian noise	Restricting assumptions	Low computational complexity
Extended Kalman Filter	Nonlinear model, Gaussian noise	Can become unstable	Low computational complexity
Particle Filter	Nonlinear model	High computational complexity	No Gaussian assumption

TABLE III: Summary of methods for filtering measurements used by observers.

In a batch approach, the decision of connecting measurement to target estimates is postponed until more measurements are available. This comes at an increased cost of storage and computational complexity. Examples of algorithms are the Viterbi algorithm, which finds the most likely sequence of hidden states given a sequence of measurements [69]. Another is the expectation maximum algorithm described in Section IV-E. Other batch approaches can be classified as multiple hypothesis tracking, where a measurement can be associated with multiple target estimates, and it is not until later that it is associated with a single-target estimate. Examples of these methods are integer programming, Lagrangian relaxation, approximate linear programming, and Markov Chain Monte Carlo (MCMC)-based data association. In addition, particle filters described in section IV-D have also been used for solving data association problems.

H. Summary

Table III summarizes the well-known properties of the three most common types of filters used by observers in MSN settings for target tracking.

V. Trajectory Planning

In this section, we will discuss high-level control algorithms for the trajectory planning of the sensors in MSNs with an application of target searching and tracking. We divide the algorithms into three different categories. First, we present explicit control algorithms. In an explicit control algorithm, the actuator input can be calculated directly from sensor measurement, state estimates, and so on. This includes gradient-descent type algorithms, as well as classical algorithms, such as the proportional-integral-derivative (PID) algorithm. Second, we examine optimization techniques. In these algorithms, the actuator is often the decision variable in an optimization problem, and it can, for example, be implemented in a receding horizon fashion. Other decision variables can be, for example, the location of each sensor. This typically leads to an implicit formulation for the actuator input. Finally, we explore a category called heuristics. Here, each sensor operates in a rule-based fashion. It is similar to explicit control, but has more of an if-else sentence structure. All types of algorithms can be implemented both in a distributed and centralized fashion. However, the trend is that explicit and heuristic algorithms often are distributed, while optimization strategies typically are centralized.

A. Explicit Control

In explicit control, the actuator input can be calculated as a function of the sensor states, state of other sensors, and the estimated state of the target(s), and, sometimes, the state of the environment. Let the state of sensor i be z_i and the actuator input be u_i. Furthermore, let $\boldsymbol{z} = [z_1, z_2, \dots, z_n]$ denote the state of all sensors and $\boldsymbol{x} = [x_1, x_2, \dots, x_n]$ denote the state of all targets. Notice here that we use the notation for the actual state x and not the estimate state $\hat{x}$. In practice, we will always have to use an estimated state in the control law, but, for simplicity, we will use the notation for the actual state through this section and the remaining part of this survey. We can write the general sensor dynamic equation and feedback control law as:

$$\dot{z}_i = f(z_i, u_i) \tag{11a}$$

$$u_i = g(\boldsymbol{z}, \boldsymbol{x}) \tag{11b}$$

In the remainder of this section we will discuss different types of explicit control used by mobile sensor networks. First is a gradient-based algorithm. In these algorithms a potential function is constructed, and the derivative is used by the trajectory planner. The potential function can be based on distance to neighboring sensors, probability of finding a target, and so forth. Second is flocking control. Here, the goal is to get the sensors to behave as a unit with a distributed control algorithm. It can be thought of as a special case of gradient-based control. Third, we will discuss P-controllers and other controllers applied by the sensors in MSNs.

1) Gradient-based control: Gradient-based control is well-suited for distributed approaches for MSNs. They usually work by utilizing local information about other sensors, obstacles, and targets to construct a potential, or cost, function. Then, the gradient of the potential function is calculated and used as an input to the actuator function. The sensor decreases the potential function by moving in the direction of the gradient. This can be written as

$$U(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{o}) = w_s u_s(\boldsymbol{z}) + w_o u_o(z_i, \boldsymbol{o}) + w_n u_n(z_i, \boldsymbol{x}) \tag{12a}$$

$$u_i = \nabla_{z_i} U(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{o}) \tag{12b}$$

where $U(\boldsymbol{z}, \boldsymbol{x})$ is the potential function based on the state of all sensors $\boldsymbol{z}$, the target state $\boldsymbol{x}$, and obstacles $\boldsymbol{o}$. The three terms $u_s()$, $u_o()$, and $u_n()$ correspond to collision avoidance and connectivity to other sensors, obstacle avoidance, and navigation, respectively. Connectivity is to keep sensors in communication range with each other. The constants w_s, w_o and w_n are tunable constants to weight the different objectives against each other. We use ∇_{z_i} to denote the gradient with respect to the state of sensor i. Usually, the gradient is found with respect to the position of the sensor.

There are some challenges with a gradient-based approach. The potential function should be convex to move the sensor towards a global optimum. The sensor risks getting stuck in a local minimum with a non-convex potential function. Another challenge is to weight different objectives. A gradient function usually has multiple tunable constants for weight collision and obstacle avoidance, target tracking, and connectivity. It is also often necessary to have a constant deciding how fast the sensor should move towards reducing the potential function.

The typical collision and connectivity potential function is based on distance between a sensor and its neighboring sensors. If we let p_i denote the position of sensor i, we can write

$$u_s(\boldsymbol{z}) = \sum_{j \in \mathcal{N}_i} \varphi(||p_i - p_j||) \tag{13}$$

where $\mathcal{N}_i$ denotes the neighborhood of sensor i, and $\varphi()$ is a function designed to either repel sensors when they get too close or attract them when they are too far away. Often it does both. A typical choice for $\varphi()$ is an inverse proportional function, such as, for example, Giannini et al. [39], Zhao et al. [70], Ma et al. [71], Hu et al. [72] and Yang et al. [73]. Another option is to use an exponential function, as used by [74, 40]. An interesting choice by the authors of Li et al. [75] is to design $\varphi()$ such that it is zero in the interval $||p_i - p_j|| \in [\delta_1, R_c\delta_2]$, where δ_1 and δ_2 are small constants, and R_c is the communication range of the sensors.

Similarly to the sensor function, an obstacle function can be constructed as

$$u_o(z_i, \boldsymbol{o}) = \sum_{j \in \mathcal{N}_i} \varphi(||p_i - o_j||) \tag{14}$$

where $\mathcal{N}_i$ denotes the obstacles in the neighborhood of sensor i. This time $\varphi()$ is designed solely to repel the sensors from the obstacles.

To design the navigation term, the following properties are often utilized: The probability distribution of a target obtained from an observer (see Section IV), probability of detection, and distance to the target. It is therefore hard to write a more specific equation for u_n than the one written in (12a). Instead, we will discuss some of the approaches taken in the literature.

In single-target tracking applications, there are multiple ways of designing the navigation term. In Rigatos [40] and Zhao et al. [70], the authors take into consideration the heading of the sensors. The goal of Zhao et al. [4] is to get the sensors on an ellipse surrounding the target within a certain distance of each other. In Rigatos [40], a stochastic variable is added to the gradient algorithm to avoid local minimums. A simpler approach is to use a threshold and constant attractive force for the target on the sensors, such as that used by Ma et al. [71]. Both Chattopadhyay et al. [76] and Giannini et al. [39] use the distance to the target when designing the navigation term. Finally, the authors of Gusrialdi et al. [77] and Li et al. [75] apply probability calculations to design the navigation term. In Gusrialdi et al. [77], the authors calculate the expected probability of finding the target, while Li et al. [75] use the probability that the target is in a region. The author also uses a gradient-based approach that utilizes multiple objectives: maximize connectivity, minimize movement, and minimize target escape probability [78]. This approach is later expanded to handle noisy measurements in Li and Liu [79].

Another approach to dealing with single-target searching is that of Hutchinson and Bretl [80]. Here the authors use the probability of missed detection to design the navigation term. The author Nelson has two papers dealing with evasive targets, in which he uses a gradient approach. In Nelson and Freeman [60]. he uses the anticipated measurement to design the navigation term, while in Nelson and Freeman [59] the authors use the trace of the covariance matrix from a H_∞ filter (see Section IV-F).

Rout and Roy [81], Hu et al. [72], Jha et al. [82], and Yang et al. [73] deal with multiple-target tracking. They all apply distance to target to the design of the navigation term. In Mathew et al. [83], the authors use a probability distribution for the targets to calculate a gradient-based algorithm.

Finally, we will discuss two papers that use a similar approach to gradient-based control without first constructing a potential function. First, in Sun et al. [84], the authors work on multiple target searching and tracking. They use two modes, search and track, each with a separate control law. In the search mode, the control law consists of multiple terms: a term for probability of detecting a target, another term for collision avoidance, a third to keep the sensor within the search area, and, finally, a momentum term to make it difficult for the UAV to change direction rapidly. Second, the authors of Yanmaz and Guclu [85] calculate forces based on distance to other sensors and move the sensor away from the others to achieve full coverage of an area when performing target searching.

In the papers referenced above, the authors must often make many assumptions and set multiple constants for the algorithm to perform well. Usually, the algorithm is demonstrated in a simulation, where the authors can select parameters such that their algorithm performs well and fulfills their assumptions. However, multiple assumptions and user set constants will likely make these algorithms difficult to apply in practice. A problem with designing the navigation term based only on the position of the target(s) is that the algorithm must know the location of the target to find it. This make the application of these algorithm limited when it comes to searching for unknown target.

2) Flocking Control: Flocking control can be considered a version of gradient-based control, but we have chosen to discuss it in its own section as it usually deals with a specific problem, namely single-target tracking with a distributed control algorithm. It is inspired by biological systems of birds, fish, and insects.

The behaviors that lead to simulated flocking were first stated by Reynolds [86] (in decreasing precedence):

1) Collision Avoidance: avoiding collisions with nearby flockmates;
2) Velocity matching: attempting to match velocity with nearby flockmates; and
3) Flock Centering: attempting to stay close to nearby flockmates.

As pointed out by Olfati-Saber [87], these rules are often referred to as cohesion, separation, and alignment rules in the literature. Olfati-Saber also points out that these rules have a broad interpretation and that it is not trivial to implement them. Additionally, he proves that an algorithm following the above rules does not necessarily lead to uniform flocking behavior. For example, the phenomena where multiple separate flocks are formed, which is called fragmentation, can occur.

Each sensor in a flocking algorithm is typically modeled as a second-order integrator

$$\dot{q}_i = p_i \tag{15a}$$

$$\dot{p}_i = u_i \tag{15b}$$

where q and p are the position and velocity of sensor number i. Normally, they are of dimension two. The actuator is u_i and typically consists of three terms:

$$u_i = f_i^g + f_i^d + f_i^\gamma \tag{16}$$

where f_i^g controls the distance between the sensors, f_i^d is a damping term that ensures velocity matching between the sensors, while f_i^γ is navigational feedback based on the flock's objective

In the paper [87], Olfati-Saber introduces three different flocking algorithms, two for free-space and one with obstacle avoidance. It can be viewed as a tutorial paper for flocking algorithms and is one of the most cited papers within the field and it is also commonly used for algorithm comparison. In later work, the same author has applied his algorithm to single-target tracking [88, 33]. He also concludes that the practical need for collision avoidance, combined with a moving rendezvous, leads to an emergence of flocking behavior even without explicit communication between sensors. In terms of cooperation, Olfati-Saber has also applied the flocking algorithm to a coverage problem for multi-target tracking [34].

The work of Olfati-Saber and his collaborators has been extended by La and Sheng in multiple aspects. First, in single-target tracking, the actuator input for the sensors, equation (16), consists of multiple tunable gains for each term. To select these gains, they introduce the following objective function in La et al. [89]

$$F = \frac{\sum_i \int_0^T ||q_i(t) - q_t(t)||dt}{T \sum_i ||q_i(t=0) - q_t(t=0)||} \tag{17}$$

here q_i is the position of each sensor, with q_t is the position of the target, and T is the simulation time. Minimizing this objective function corresponds to minimizing the time and distance for the sensor network to catch up to the target. This is a non-convex and non-differentiable objective function. The authors apply a genetic algorithm to solve the optimization problem. A genetic algorithm is inspired by Charles Darwin's theory of natural selection. A set of solutions are generated randomly, and the objective function is used for evaluation. Then, the best solutions are merged together, which is called matching. The new solutions are called offspring. Often some of the offspring are also exposed to random mutation, which are small random changes. Solutions that perform poorly are eliminated. The above steps are performed iteratively until a satisfying solution is produced. In La et al. [89], the authors use a Gaussian distribution to generate solutions. The solutions that do not satisfy certain constraints are eliminated. Matching is performed, but they do not use mutations. Another solution to the objective function (17) is that of Khodayari et al. [90]. They suggest their own algorithm, which they call the gravitational algorithm. This is inspired by Newton's law of gravity.

La and Sheng also extend the work in obstacle management. In Olfati-Saber [87], the sensors split when they meet obstacles, which can lead the network becoming disconnected. La and Sheng [91] introduce what we will name the squeezing algorithm. This works by manipulating the required distance between sensors when passing obstacles. Instead of going around an obstacle on separate sides, the sensors "squeeze" together such that all can pass on the same side. Another extension of Olfati-Saber by La and Sheng is the center of mass (CoM) algorithm. If we consider a 2D view of an MSN and think of it as a web, the algorithm from Olfati-Saber [88] has no guarantee that the target will be at the center of the web during the tracking. In La and Sheng [92], the authors introduce an algorithm in which each sensor estimates the center of the MSN. This enables the sensor network to act a single unit when tracking a target. In later work, the authors combine the CoM algorithm with a distributed filter for tracking [63]. The CoM algorithm is also extended to handle noisy measurements of each sensor's position and velocity [93]. A third extension of Olfati-Saber [87] by La and Sheng is in multiple-target tracking [94]. Here, they split a flock into two when encountering a new target. To split the sensors, they use a seeding algorithm. It is initiated by the sensor closest to the new target, which messages it closest neighbors to follow the new target. This continues until the number of sensors following the new target reach a predetermined number. Merging happens when a target disappears, in the same way as a flock is formed. In La and Sheng [95], the multiple-target tracking algorithm is combined with the "squeezing" algorithm from La and Sheng [91].

In recent years, there have been several other aspects using flocking algorithms for single-target tracking by an MSN. Gu and Hu [96] study tracking an evasive target. They use a distributed minmax filter, which tries to maximize the worst case of the tracking performance. In Gu and Hu [50], the same authors combine flocking control with a distributed particle filter. They apply the approximation of the posterior distribution from the particle filter in the flocking control algorithm such that all the sensors are driven towards the target. Tu, Wang and their collaborators designed an algorithm for deployment around a slow-moving target, which considers the heading of the sensors [97]. The flocking algorithm is demonstrated in experiments in Wang et al. [98]. The authors also have an algorithm for deployment which requires only 1-hop neighborhood communication between sensors [99]. This algorithm also avoids "holes" in the network. A hole in an

MSN can be a small area which no sensor is covering, thus making it possible for a target to hide within the network. Another approach that also focuses on avoiding holes is Zhang and Zhu [100]. Most papers concerning flocking consider the control and target state estimators separately. In Jin et al. [38], the authors combine a flocking algorithm similar to Olfati-Saber with a distributed Kalman filter. They analyze the coupled controller and observer using Lyapunov theory to prove global asymptotic stability. An additional application is coupled-target tracking studied by Su et al. [35]. Here, two targets with coupled behavior are tracked by splitting the sensor network into two. Each part has a sensor acting as leader with the ability to broadcast messages to the other sensors, which only have local communication abilities. By taking advantage of the coupled behavior, the authors manage to get increased performance compared to treating the two targets as separate tasks. The authors of Dang and Horn [101] study a centralized approach to flocking for multi-target tracking. Here, the authors use a splitting algorithm to divide a flock when a new target is encountered. In addition to the terms in the actuator equation (16), the authors use a merging term for the free-sensors that do not belong to a flock. The same authors also work on single-target tracking in a noisy environment [102]. Here, they develop a distributed algorithm, where one sensor is selected to be the leader and follow the target, while the others avoid collision and follow the leader. The leader is chosen based on the distance to the target. The sensor network uses a V-shape when chasing a target and assumes a circular shape around the target when it is caught. In Jiang et al. [103], the authors study single-target tracking by combining flocking control and a distributed filter. The authors use only a potential function for distance between sensors to avoid collision and have no velocity matching. Instead, they use a proportional controller (P-controller) to match each sensor to the position, velocity, and acceleration of the target.

Compared to the problem of searching and tracking multiple targets, flocking control has limited applicability. The most common situation, for a flocking algorithm is to have many sensors and track a single target. For example,in a searching application, it is often not desired to have the sensors flock together, but rather spread out to cover as much ground as possible. Often cost is an important factor, which makes it questionable to use multiple sensors if one is sufficient.

3) P-controller: In this section, we present control strategies which can be classified as, or similar to, a P-controller. A P-controller uses a desired state as input. Then, it calculates the difference between the actual and desired states and applies an actuator signal proportional to the difference. This is typically written as

$$u_i = k_p(z_{i,\text{desired}} - z_i) \tag{18}$$

where k_p is a tunable gain with $z_{i,\text{desired}}$ and z_i denoting the desired and actual state of sensor i, respectively. This type of controller is often used in cooperation with a high-level control, which is, for example, used to find the desired state.

In Xie et al. [56], the authors formulate an optimization problem to obtain the optimal distance and heading towards a target. Then, they use a P-controller to move each sensor towards its optimal distance and heading. The authors of Tan et al. [104] calculate a Voronoi diagram (a way of partitioning the plane into a number of equal regions) and use a P-controller to move each sensor towards the center of its cell. A more complex control law is:

$$u_i = -k_i h'(C^* - \mathcal{T}_{\mathcal{N}}), \tag{19}$$

which is applied by Wang et al. [62] to track a single target. Here k_i and h' are a constant and a penalty function, respectively. The desired coverage of the target is C^* while the actual coverage by the mobile sensor network is $\mathcal{T}_{\mathcal{N}}$. The authors prove asymptotic stability of the controller using the Lyapunov theory. In Kuo et al. [105], the authors form a control law to trap a target within a polygon formed by the mobile sensors. To move the polygon, each sensor continuously computes its next waypoint and move towards it using a P-controller.

A P-controller is, in general, not recommended for a high level trajectory algorithm. It has limited applicability since it is difficult to incorporate a higher level objectives. In the above examples, it is only used to move the sensors straight for a desired position.

4) Other explicit controllers: In this last section on explicit control, we present a paper that does fit into the above categories, but which can be considered to cover an explicit controller.

In Xu et al. [106], the authors design a control law for multiple-target tracking. They assume that the targets move in an acceleration field and use this to design a control law similar to backstepping. Backstepping is a technique for recursively designed stabilizing control for each subsystem. A disadvantage with backstepping is that it requires an accurate model.

B. Optimization

We define optimization control to minimize (or maximize) any objective function using actuator input or sensor location as manipulative variables. This lead to an implicit formulation for the actuator input. The actuator input can, for example, be applied to each sensor in a finite horizon fashion or as an assignment. A general optimization control problem can be written as

$$\min_{\boldsymbol{u}} f(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{u}) \tag{20a}$$

s.t.

$$g_1(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{u}) \geq 0 \tag{20b}$$

$$g_2(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{u}) = 0 \tag{20c}$$

where $\boldsymbol{z}$, $\boldsymbol{x}$, $\boldsymbol{u}$ are the aggregated sensor states, target states, and actuator inputs, respectively. The objective function is $f()$ with inequality and equality constraints denoted $g_1()$ and $g_2()$, respectively. First, we will discuss task allocation algorithms. These are typically applied in a centralized fashion and set the new waypoint for each sensor. Then, we will go into optimal control. Here the actuator of the sensors is used as a decision variable, and the optimization is continuous. In the last part of this section, we will discuss various other optimization strategies.

1) Task allocation: A typical task allocation deals with either single- or multiple-target tracking. There are usually two steps involved. First, a set of new positions is obtained for the sensors based on, for example, target positions, expected measurements, and so on. Second, a combinatorial formulation is used to set up an assignment problem, which, in its most basic form, can be solved by the Hungarian algorithm. A centralized formulation for the assignment problem is

$$\min_{a} \sum_{i\in\mathcal{P}_{\text{current}}} \sum_{j\in\mathcal{P}_{\text{new}}} C(i,j)a_{ij} \tag{21a}$$

$$s.t.$$

$$\sum_{i\in\mathcal{P}_{\text{current}}} a_{ij} = 1 \quad \forall j \in \mathcal{P}_{\text{new}} \tag{21b}$$

$$\sum_{j\in\mathcal{P}_{\text{new}}} a_{ij} = 1 \quad \forall i \in \mathcal{P}_{\text{current}} \tag{21c}$$

where the sets of current and new sensor positions are $\mathcal{P}_{\text{current}}$ and $\mathcal{P}_{\text{new}}$, respectively. Note that these do not have to be of equal size, as some sensors may not have to move. Each move is associated with a cost stored in the matrix $C(i,j)$. Here we use a binary matrix (values 0 or 1) with entries denoted a_{ij}, which is 1 if the sensor at position i should move to position j and 0 otherwise.

Usually, the objective is to minimize energy consumption of the network. Normally, this is done by minimizing the movement of the sensors by letting the cost, $C(i,j)$, in equation (21) represent distance. This is a different problem than a typical gradient-based control problem, where the sensors are expected to be continuously moving. An advantage over gradient-based approaches is that task allocation typically leads to an optimal solution even with non-convex problems. The assignment problem can be solved in polynomial time with the Hungarian algorithm [68]. Unfortunately, many of the target tracking formulations for MSNs also include non-linear constraints, making the problem NP-hard, with no known solution within polynomial time. Another challenge is to implement assignment problems in a distributed fashion.

There are multiple papers that deal with single tracking, which uses an objective function in which the cost matrix consists of distances between current and new sensor positions. In Bai et al. [26], the new sensor positions are found by minimizing target uncertainty. The assignment problem, equation (21), is solved using the Hungarian method [68]. The authors of Qi et al. [107] modify the constraint (21c) to only require that each sensor visits at most a new position (they switch out the equality for inequality constraints). In addition, they add a non-linear constraint that requires the probability of detection to be above a given threshold. This is also the criteria they use to find new positions. To solve the problem, they divide it into two sub-problems, one of which can be solved online through the Hungarian algorithm. Instead of reducing the minimum distance, the sensors should travel. The authors of Li and Liu [108] minimize the velocity of each sensor

$$\min \sum_{i\in\mathcal{S}} ||v_i|| \tag{22}$$

where $\mathcal{S}$ is the set of all sensors, and the velocity of sensor i is v_i. The constraints they use are related to tracking performance of the target and connectivity (communication between nodes). The authors of Mahboubi et al. [109] assume that the movement of the sensors is negligible compared to communication with respect to energy consumption. Energy consumption is considered proportional to the distance between the sensors. The authors suggest an algorithm to optimize the movement of the sensors to minimize the communication distance between them while tracking a target. In another paper by the same author [110], he converts an MSN to a graph and finds a weight for each edge corresponding to a communication cost. A shortest path algorithm is then used to find the path which minimizes the energy necessary to message the location of the target to the base station. Similar approaches are applied in Mahboubi et al. [111, 112, 113]. A distributed approach is presented in Zou and Chakrabarty [114]. The algorithm works in two steps. First, each sensor applies a Baysian filer to decide a new position to move to in order to improve sensing of the target. At the same time, the cost associated with the movement is calculated. This is communicated back to a base station, as well as neighboring sensors, before a decision is made as to which sensors should move. Another centralized approach is Mourad et al. [115], where the authors use both stationary and mobile sensors. The stationary sensors are used to cover the area, while the mobile sensors are used to track a target. The tracking is done by estimating new positions for the sensors based on the target model and then setting up an assignment problem to minimize the moving distance for the mobile sensors, which is solved by ant colony optimization.

Multiple-target tracking by MSNs can also be solved by formulations similar to the assignment problem using distance as the cost function in equation (21). In Liao et al. [116], the authors use Voronoi diagrams to find the new positions of sensors before applying the Hungarian algorithm to solve the assignment problem. The authors of Sharma [117] estimate the next position of each target and use the ant colony optimization technique to solve the assignment problem. Another similar approach is by Selvaraj and Balaji [118]. Here, the authors find the best positions for the sensor to cover the area before applying particle swarm optimization to assign sensors to those positions. A set of Kalman filters are used for the target's estimates in Fu and Yang [119]. They then use the inverse of the covariance matrix, along with the Cramer Rao Lower Bond [120], as an objective function. In addition, they minimize energy consumption and use a constraint to maintain connectivity. They prove that the resulting optimization problem is NP-hard, and they develop an approximate algorithm that runs in polynomial time. In Gao et al. [121], the authors formulate an assignment problem, which minimizes sensor movement, while requiring coverage of the monitored area. The authors of

Kamath et al. [122] study the problem of assigning two sensors to track each of multiple targets. They prove that the problem is NP-hard, and suggest an approximate algorithm to solve it in polynomial time. A distributed approach is introduced by Zorbas and Razafindralambo [123] in which the goal is to maximize network lifetime. The algorithm works in two steps. First, connectivity is ensured, and a leader is selected. Second, minimum movement with required coverage of the targets is used to select the movement of sensors.

Besides minimizing movement distance, there are some papers that try to handle energy consumption more directly for single-target tracking. In Marbukh et al. [124], the authors model battery consumption of the sensors and use an optimization technique called simulated annealing to trade-off sensor movement versus battery consumption. The authors of [125] allow sensors to be sleeping or listening, in addition to measuring and moving and formulate an integer optimization problem to maximize the lifetime of the MSN

In this paragraph, we present a few approaches that use a task allocation algorithm, but not to minimize energy consumption of the MSN. Low and his collaborators study target searching and tracking inside an indoor space using an MSN [126, 127, 128]. They use an ant colony optimization to dynamically allocate sensors to different areas, depending on the number of targets in each area. The author of [129] uses a distributed approach to allocate sensors to multiple stationary targets. He assumes that each sensor knows the number of sensors required to monitor each target and the location of the targets, but not the location of the other sensors. To discover this, each sensor needs to visit a target to see if it is monitored by enough sensors. The strategy is based on each sensor weighing the benefits of visiting targets to decide its visiting sequence. Another allocation problem is studied by Chang et al. [130]. Here, the problem is to patrol a given number of stationary targets. First, the number of sensors is decided and then multiple patrolling paths for the sensors are constructed. The authors develop a task allocation algorithm for matching sensors to patrol paths.

The assignment problem usually assumes that the positions of the targets are known, or at least have a solid estimate. In other words, these types of formulations are poor at searching for targets. Another, challenge is that the assignment problem, equation (21), is static. Searching and tracking problems are often highly dynamic, and solving an optimal problem for an instance, might lead to undesired local minimums in a longer time horizon. Finally, some of the assumptions applied to get the optimization problem on the form of equation (21) might make the solution invalid to the original problem. For example, some sensors might have nonholonomic moving constraints, which are often assumed to be holnomic.

2) Optimal Control: Optimal control problems seek to find the optimal actuator input given some objective function along with state dynamics and other constraints. A general optimal control problem can be written as

$$\min_{u(\cdot)} \int_{t_0}^{t_f} L(\xi, u) dt \tag{23a}$$

s.t.

$$\dot{\xi} = f(\xi, u) \tag{23b}$$

$$\xi_{\min} \leq \xi \leq \xi_{\max} \tag{23c}$$

$$u_{\min} \leq u \leq u_{\max} \tag{23d}$$

where $L(\xi, u)$ is the cost function with state ξ and actuator input u. The state dynamics are given by the differential function $f(\xi, u)$, with limits on the state and actuator given by equations (23c) and (23d), respectively.

An optimal control problem is typically converted to a large non-linear problem (NLP), which is a set of algebraic equations. Techniques, such as collocation, can be applied for approximating the objective function. Usually an interior-point or active set technique is applied to solve the NLP problem. See Betts [131] and Biegler [132] for details.

In target tracking, an optimal control strategy enables the sensor dynamics to be taken into consideration, as well as more complex objective functions. In Wei and Ferrari [133], the authors study single-target searching. They use an objective function which is a joint probability function of the position, heading, and velocity of the target. Sensor dynamics are included in the constraint (23b), and the actuator is limited. To solve the optimal control problem, the authors apply an approximation method called a variational iteration method. Optimal control techniques have also been applied to multiple-target tracking. In Haugen and Imsland [27], the authors use UAVs to track icebergs. They apply Kalman filters to the tracking and use the trace covariance matrices in the objective function, along with the actuator of the UAVs. The dynamics of both the sensors (UAVs) and targets (icebergs) are included in the formulation, and a constraint is used to handle collision avoidance. To solve the optimal control problem, they use collocation to transform it into an NLP, which is solved with an interior-point solver. Another optimal control formulation is presented by Baumgartner et al. [134] for multiple-target tracking using underwater vehicles as sensors. The objective function tries to maximize coverage along with minimizing energy consumption of the sensors. The dynamics of the sensors are included along with environmental information such as current as constraints. To solve the optimal control problem, the authors use a direct shooting method.

Optimal control problems are, in general, highly non-convex. This means that the solution we find is will depended on the initialization of the problem. In the above papers, there is often a lack of discussion of how to initialize these problems even though this can be critical to the resulting solution. Another problem is the computational complexity. In a highly dynamic problem, solving equation (23) might take too long, when the solution is ready the original problem is no longer valid. Finally, there might be several user set parameters to make equation (23) possible to solve numerically. This often requires expert knowledge to apply. For example, the horizon can be difficult to select, as a short horizon can lead to getting stuck in a local optima, while a long horizon might take too long to compute.

3) Other Optimization Controls: There are a few papers which do not fit directly into the above categories. We present them here. First, Cheng et al. [24] studies single-target tracking and formulates an optimization problem, which tries to maximize the sensing quality of the target while maintaining coverage of the search area. The controller is implemented in a receding horizon fashion. This means that the optimization problem is solved and only the first sequence of the solution is used. Then, the optimization problem is solved again. This approach is typically applied to dynamic problems, where new information frequently becomes readily available. The authors of Park and Hutchinson [135] study a special formulation for single-target tracking, in which some of the sensors are expected to fail or even send erroneous messages. To solve this problem, the authors models it as an adversarial task, where some of the sensors are trying to sabotage detection of the target. The authors apply multi-stage decision to model the problem and solve it using dynamic programming over a receding horizon.

As with optimal control, it is challenging to select an appropriate time horizon for receding horizon problem.

C. Heuristic Control

We mean rule-based control when we discuss heuristic control. This can, for example, take a structure such as if-else sentences or be a grid map with rules for what actions a sensor should take in each cell. Typical tools are partially observable Markov decision processes or a dynamic Bayesian network to model the problem, and strategies from the field of artificial intelligence are often applied.

A popular problem often solved using techniques from artificial intelligence is evasive target searching or tracking. A simple strategy is applied by Chen et al. [136] to solve this problem. Here, the authors use a grid map to model the probability of capturing a target in each cell. The sensors then move to the adjacent cell with the highest probability of target capture. The authors analyze how many sensors are necessary to apply to a given area to have a given probability of capturing a target within a given deadline. The work is later expanded in Hsu et al. [137]. A similar approach is taken by Imai and Ushio [138] to tracking multiple non-evasive targets. Another approach for an evasive single target is presented by Chin et al. [139]. They study the problem both from the pursuer (sensor) and target's perspectives. They utilize game theory to design strategies. In addition, a communication protocol is developed for the sensors to cooperate. The authors of Ferrari et al. [140] study multiple evasive target capture. The area is modeled as a grid, which is decomposed into a connectivity graph. The search is performed by an A* algorithm utilizing an expression for the probability of detection.

Another set of target tracking problems from the field of artificial intelligence is cluster and Q-learning. In Prabhavathi and Rajeshwari [25], the target state is estimated using a Kalman filter. Then, a sensor is chosen to be the cluster head. This sensor gathers measurements of the target from the sensors in its area and fuses them together. As the target moves, the cluster head continues to track the target until it is outside of its area. The tracking information is then passed on by the cluster head to an adjacent cluster head. The strategy for choosing cluster heads is to maximize the lifetime of the network. The authors of Ferrari et al. [74] combine a gradient-based approach for obstacle and collision avoidance with a Q-learning algorithm to decide the action for each sensor. Q-learning is a learning technique where the sensors decide what actions to take based on the information utilities of the available actions [141].

Finally, we have some approaches that are rule-based with a similar structure such as a set of if-else sentences. The authors of Krishna et al. [142] study multiple-target searching. The area is modeled using a grid, dividing it into cells. When a target is discovered by a sensor, the sensor uses fuzzy control to decide whether to continue to search or to start to track that target. Another rule-based approach is that of Takahashi et al. [143], which focuses on single-target tracking with obstacles which prevent communication to a base station. To maintain contact with the base station, each sensor uses a set of rules to decide its actions. The problem of multiple evasive target searching is studied by Rahman et al. [144]. Here, the authors suggest that the sensors scan for targets using different predefined formations.

An advantage with heuristic approaches is that they often will work better in real-world experiments as they often have fewer limiting assumptions. They can also often adapt to the situation they are in like, for example, Q-learning. A challenge is that it can be difficult to design good heuristic algorithms, and it can be hard to prove overall desired behavior.

VI. Classification of Trajectory Planning

The papers presented in this survey (Sections IV and V) have multiple problem formulations. In addition, the solutions have some different characteristics. In this section, we present a classification of all the papers presented in this survey.

We have chosen to classify each paper based on three criteria. First, we separate the papers based on whether the approach is centralized or distributed/decentralized. Second, does a paper usually focuses on a single target or multiple targets? Third, the categories tracking, searching and a combination of both are used to separate the papers. Tracking papers usually have a priori estimates of each target, and the focus is on tracking those targets. In papers about searching the authors suggest strategies for finding targets in a given area. Finally, the hardest problem is managing both. Here, sensors usually must weigh the trade-off between searching for new targets and tracking detected targets.

VII. Discussion and Future Work

In this section, we will discuss the research within target searching and tracking, as well as suggest directions for future research efforts.

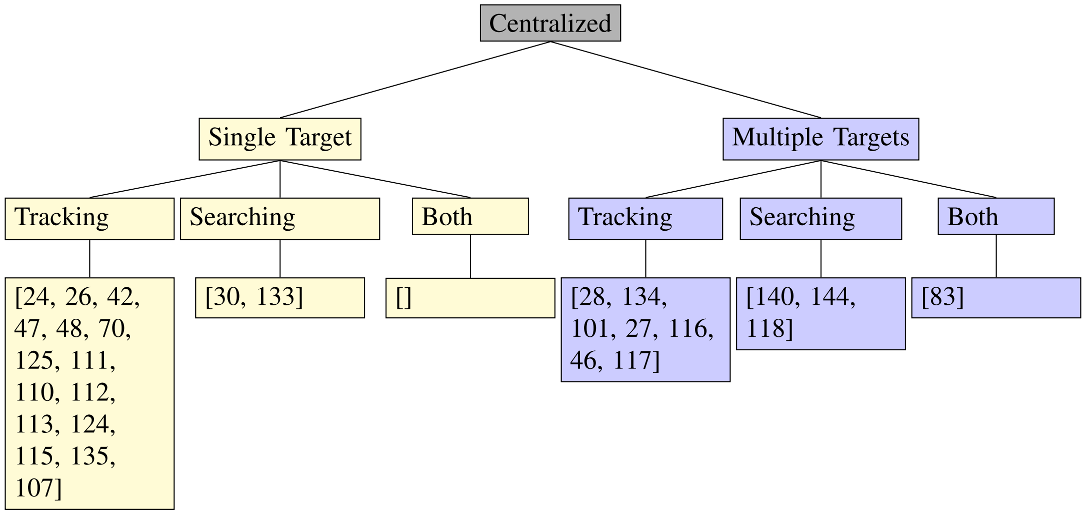

Fig. 5: Classification of centralized approaches presented in this survey

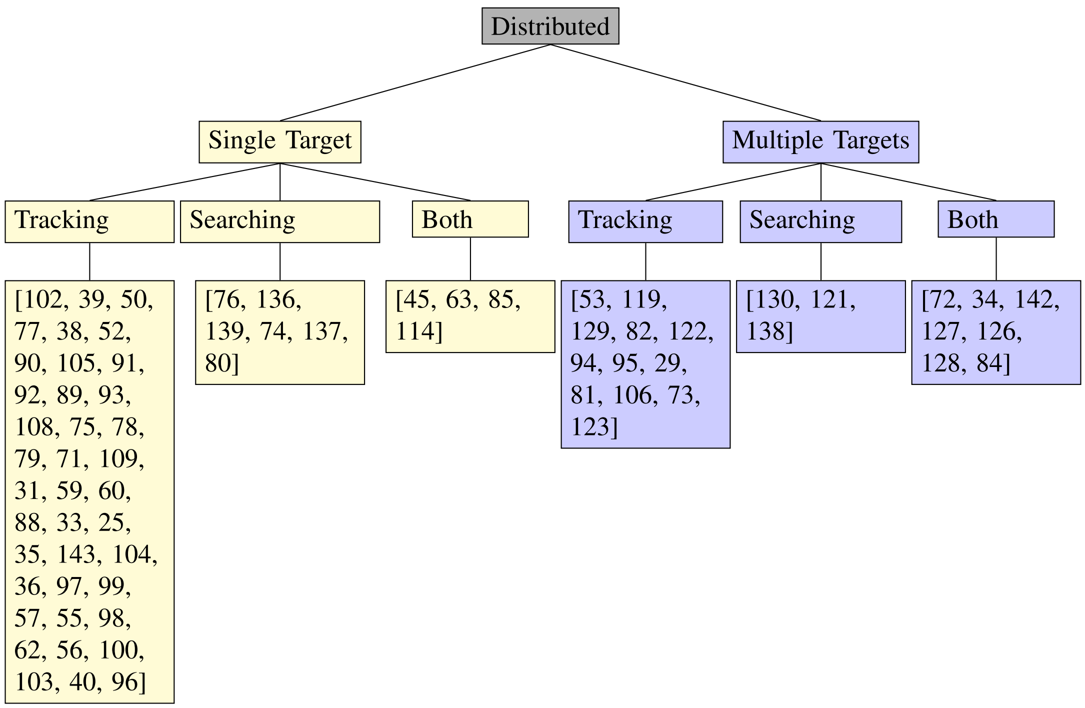

Fig. 6: Classification of distributed approaches presented in this survey

A challenge with this literature is that there are almost as many problem formulations as there are strategies to solve them. Both problem definition and the performance measurements are not well defined. This makes it hard to compare approaches and decide what the state-of-the-art is for a specific problem. The reason is that there are so many different applications involving targets for MSNs. However, it would be useful to have some clear problem definition, along with a performance score, to compare different solution strategies.

There is a lack of real-world experiments. Even with the immense potential for MSN there has yet to be reported an industrial-sized implementation. Most papers only demonstrate their approaches in simulations, and the few experiments reported are done in lab settings. More real-world implementations will also make the requirements for the MSN clearer.

Most papers focus on single-target tracking. However, using an MSN to track a single target can be too simple for many real-world applications. For example, in a search and rescue operation, it is not likely that there will be a sufficient number of sensors to cover the entire area and there might not even be a sensor per target. The challenge then becomes balancing both target searching and tracking with fewer sensors than targets. In most approaches for multiple-target searching and tracking, there are more sensors than targets, which makes it possible to split the sensors into different groups such that each target can be tracked separately.

When it comes to the development of efficient filters there is still a gap between research and real-world application. Very few articles focus on the data association problem, Section IV-D, which becomes even more difficult when measurements must be coordinated between multiple sensors. Another challenge with both Kalman and particle filters is that they require a model of the target they are estimating, in addition to a probability distribution for the measurement and process noise. It is possible to use general models, but these might be suboptimal as target behavior can be revealed through measurements. One potential future direction for research could be to introduce filter with learning ability. For example, first classify the type of target and then apply an appropriate model and probability density function.

Another related topic to filters is the how long a target should be observed which relates to the quality of the estimate. A common approach when using a Kalman filter is to use the covariance matrix of the estimate error, P, to measure the quality of the current estimate. However, this depends on the covariance of the process noise. If these are inaccurate it is difficult to know how long to stay with a target to obtain sufficiently confidence in an estimate. The quality of the estimate is also central to how different estimates should be merged together between multiple sensors, as discussed in Section IV-C and IV-E. More research efforts should be put into quantity the target estimate. An additional application for this quantification could be to decide the trade-off between searching for new targets and tracking old targets.

All the observers presented in this survey are based on a theoretical formulation. This involves making assumptions that might be violated in an actual application. The new techniques and success by machine learning could be an alternative way of designing observers. A data-driven approach would potentially be closer to a real-world application. An exciting research opportunity might be to combine traditional techniques with machine learning to get the best from both worlds.

The high-level trajectory algorithms presented in this survey are often well suited for either searching or, more often, tracking. There are few strategies that deal well with both and has a efficient way to deal with the trade-off between them. This is reflected in Figure 5 and 6 where there are few papers that deals with the combination of both searching and tracking. In comparison many of the applications presented in I-B require the combination. Future research should aim at developing algorithms that can do both searching and tracking targets.

The high-level trajectory algorithms often have different weaknesses. For example, the assignment problem simplifies the dynamics of the sensors to suit the problem formulation, while optimal control problems often are challenging to initialize. A possible opportunity for future research could be to combine these two algorithms. For example by using the assignment problem to initialize an optimal control problem. Another possible combination could be to use a simplified problem solved by either assignment or optimal control and use a heuristic algorithm to guide towards the solution. A third option could be to combine centralized and decentralized approaches. A case could be that the planning is done centrally when possible, but if communication between a sensor and base station is lost the sensor starts operating on its own.

VIII. Conclusion

In this survey, we have discussed MSNs applied to target searching and/or tracking. The focus of the survey has been on the state observer, which utilizes filters to estimate the states of one or multiple targets, and the trajectory planner, which uses information from the state observer to decide trajectories for the sensors. We have discussed the two most popular approaches in detail i.e., the Kalman and particle filters. In addition, we have also discussed some other types of filers. For the trajectory planning, we have divided the approaches into three: explicit, optimization, and heuristic. The explicit and optimization strategies have been divided into subgroups, each of which has been discussed in detail. The papers reviewed in this survey have been classified based on problem and solution characteristics. Finally, we have discussed the current state of the research, as well as possible directions for future research efforts.

References

[1] J. Zhang, L. Jia, S. Niu, F. Zhang, L. Tong, and X. Zhou, "A space-time network-based modeling framework for dynamic unmanned aerial vehicle routing in traffic incident monitoring applications," *Sensors*, vol. 15, no. 6, pp. 13 874–13 898, 2015.

[2] I. Lešinskis and A. Pavlovičs, "The aspects of implementation of unmanned aerial vehicles for ice situation awareness in maritime traffic," *Transport Means - Proceedings of the International Conference*, pp. 65–68, 2011.

[3] T. Latif, E. Whitmire, T. Novak, and A. Bozkurt, "Sound localization sensors for search and rescue biobots," *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3444–3453, 2016.

[4] W. Zhao, Z. Tang, Y. Yang, L. Wang, and S. Lan, "Cooperative search and rescue with artificial fishes based on fish-swarm algorithm for underwater wireless sensor networks," *The Scientific World Journal*, vol. 2014, no. 145306, 2014.

[5] E. Pereira, R. Bencatel, J. Correia, L. Félix, G. Gonçalves, J. Morgado, and J. Sousa, "Unmanned air vehicles for coastal and environmental research," *Journal of Coastal Research*, vol. II, pp. 1557–1561, 2009.

[6] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," *ACM SIGARCH Computer Architecture News*, vol. 30, no. 5, pp. 96–107, 2002.

[7] M. Jadaliha and J. Choi, “Environmental monitoring using autonomous aquatic robots: Sampling algorithms and experiments,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 899–905, 2013.
[8] F. Lalooses, H. Susanto, and C. H. Chang, “Recovery target tracking in wildlife,” *Wireless and Optical Communication, Montreal, Canada*, 2005.
[9] I. Amundson and X. Koutsoukos, “A survey on localization for mobile wireless sensor networks,” *Mobile entity localization and tracking in GPS-less environments*, pp. 235–254, 2009.
[10] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, and L. T. Yang, “A survey on communication and data management issues in mobile sensor networks,” *Wireless Communications and Mobile Computing*, vol. 14, no. 1, pp. 19–36, 2014.
[11] X. Ma and J. Tan, “Active sensing with mobile sensor networks: A survey.” *JCM*, vol. 8, no. 2, pp. 110–127, 2013.
[12] O. Dagdeviren, I. Korkmaz, F. Tekbacak, and K. Erciyes, “A survey of agent technologies for wireless sensor networks,” *IETE Technical Review*, vol. 28, no. 2, pp. 168–184, 2011.
[13] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
[14] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
[15] M. Naderan, M. Dehghan, and H. Pedram, “Mobile object tracking techniques in wireless sensor networks,” in *Ultra Modern Telecommunications & Workshops, 2009. ICUMT’09. International Conference on*. IEEE, 2009, pp. 1–8.
[16] M. Naderan, M. Dehghan, H. Pedram, and V. Hakami, “Survey of mobile object tracking protocols in wireless sensor networks: a network–centric perspective,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 11, no. 1, pp. 34–63, 2012.
[17] É. L. Souza, E. F. Nakamura, and R. W. Pazzi, “Target tracking for sensor networks: A survey,” *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, p. 30, 2016.
[18] C. Robin and S. Lacroix, “Multi-robot target detection and tracking: taxonomy and survey,” *Autonomous Robots*, vol. 40, no. 4, pp. 729–760, 2016.
[19] D. Portugal and R. Rocha, “A survey on multi-robot patrolling algorithms,” *Technological Innovation for Sustainability*, pp. 139–146, 2011.
[20] T. H. Chung, G. A. Hollinger, and V. Isler, “Search and pursuit-evasion in mobile robotics,” *Autonomous Robots*, vol. 31, no. 4, p. 299, 2011.
[21] L. D. Stone, J. O. Royset, and A. R. Washburn, *Optimal Search for Moving Targets*. Springer, 2016, vol. 237.
[22] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
[23] A. Gelb, *Applied optimal estimation*. MIT press, 1974.
[24] P. Cheng, X. Cao, J. Bai, and Y. Sun, “On optimizing sensing quality with guaranteed coverage in autonomous mobile sensor networks,” *Computer Communications*, vol. 35, no. 9, pp. 1107–1114, 2012.
[25] M. Prabhavathi and R. Rajeshwari, “Cluster-based mobility management for target tracking in mobile sensor networks,” in *Advanced Computing (ICoAC), 2011 Third International Conference on*. IEEE, 2011, pp. 198–203.
[26] J. Bai, P. Cheng, J. Chen, A. Guenard, and Y. Song, “Target tracking with limited sensing range in autonomous mobile sensor networks,” in *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on*. IEEE, 2012, pp. 329–334.
[27] J. Haugen and L. Imsland, “Monitoring moving objects using aerial mobile sensors,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 475–486, 2016.
[28] A. Albert, F. S. Leira, and L. Imsland, “Uav path planning using milp with experiments,” *Modeling, Identification and Control*, vol. 38, no. 1, p. 21, 2017.
[29] G. Ren, V. Maroulas, and I. D. Schizas, “Exploiting sensor mobility and covariance sparsity for distributed tracking of multiple sparse targets,” *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 53, 2016.
[30] W. Wu, F. Zhang, and Y. Wardi, “Target localization: Energy-information trade-offs using mobile sensor networks,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 2944–2949.
[31] S. Martínez and F. Bullo, “Optimal sensor placement and motion coordination for target tracking,” *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.
[32] R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 7036–7042.
[33] R. Olfati-Saber and P. Jalalkamali, “Coupled distributed estimation and control for mobile sensor networks,” *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2609–2614, 2012.
[34] P. Jalalkamali and R. Olfati-Saber, “Information-driven self-deployment and dynamic sensor coverage for mobile sensor networks,” in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 4933–4938.
[35] H. Su, Z. Li, and M. Z. Chen, “Distributed estimation and control for two-target tracking mobile sensor networks,” *Journal of the Franklin Institute*, vol. 354, no. 7, pp. 2994–3007, 2017.
[36] Q. Tan, X. Dong, Q. Li, and Z. Ren, “Weighted average consensus-based cubature Kalman filtering for mobile sensor networks with switching topologies,” in *Control*

& Automation (ICCA), 2017 13th IEEE International Conference on. IEEE, 2017, pp. 271–276.

[37] I. Arasaratnam and S. Haykin, “Cubature Kalman filters,” *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[38] D. Jin, W. Zeng, H. Su, H. Zhou, and M. Delie, “Distributed estimation and control of mobile sensor networks based only on position measurements,” *IET Control Theory & Applications*, vol. 11, no. 10, pp. 1627–1633, 2017.

[39] S. Giannini, D. Di Paola, and A. Rizzo, “Coverage aware distributed target tracking for mobile sensor networks,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 1386–1391.

[40] G. G. Rigatos, “A distributed motion planning and distributed filtering approach for target tracking in mobile sensor networks,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 4771–4778, 2011.

[41] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[42] Y. Li and P. M. Djuric, “Particle filtering for target tracking with mobile sensors,” in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 2. IEEE, 2007, pp. 1101–1104.

[43] ——, “Target tracking with mobile sensors using cost-reference particle filtering,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 2549–2552.

[44] J. Míguez, M. F. Bugallo, and P. M. Djurić, “A new class of particle filters for random dynamic systems with unknown statistics,” *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, pp. 2278–2294, 2004.

[45] G. M. Hoffmann and C. J. Tomlin, “Mobile sensor network control using mutual information methods and particle filters,” *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2010.

[46] W. Lu, G. Zhang, S. Ferrari, M. Anderson, and R. Fierro, “A particle-filter information potential method for tracking and monitoring maneuvering targets using a mobile sensor agent,” *The Journal of Defense Modeling and Simulation*, vol. 11, no. 1, pp. 47–58, 2014.

[47] Z. Juan-Yi, “An improved target tracking accuracy algorithm based on particle filtering in wmsn,” in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*. IEEE, 2011, pp. 131–134.

[48] F. Hu and C. Tu, “An optimization model for target tracking of mobile sensor network based on motion state prediction in emerging sensor networks,” *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 5, pp. 3509–3524, 2017.

[49] H. Chen and K. Sezaki, “Distributed target tracking algorithm for wireless sensor networks,” in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.

[50] D. Gu and H. Hu, “Rényi entropy based target tracking in mobile sensor networks,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13 558–13 563, 2011.

[51] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[52] Z. Kan, E. L. Pasiliao, J. W. Curtis, and W. E. Dixon, “Particle filter based average consensus target tracking with preservation of network connectivity,” in *Military Communications Conference, 2012-MILCOM 2012*. IEEE, 2012, pp. 1–6.

[53] J. P. Beaudeau, M. F. Bugallo, and P. M. Djurić, “Rssi-based multi-target tracking by cooperative agents using fusion of cross-target information,” *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5033–5044, 2015.

[54] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. Ieee, 2000, pp. 153–158.

[55] L. Wang, N. Wang, and H. Zhu, “Consensus based distributed unscented information filtering for air mobile sensor networks,” in *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, vol. 2. IEEE, 2010, pp. 492–495.

[56] F. Xie, H. Xiao, and Y. Wang, “Coordinated target estimation and tracking control by mobile sensor networks,” in *Advanced Robotics and Mechatronics (ICARM), International Conference on*. IEEE, 2016, pp. 399–404.

[57] L. Wang, Q. Zhang, H. Zhu, and L. Shen, “Adaptive consensus fusion estimation for MSN with communication delays and switching network topologies,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 2087–2092.

[58] U. Shaked and Y. Theodor, “H/sub infinity/-optimal estimation: a tutorial,” in *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*. IEEE, 1992, pp. 2278–2286.

[59] T. R. Nelson and R. A. Freeman, “Decentralized H ∞ filtering in a multi-agent system,” in *American Control Conference, 2009. ACC'09*. IEEE, 2009, pp. 5755–5760.

[60] ——, “Set-valued estimation for mobile sensor networks,” in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 2681–2686.

[61] N. A. Carlson, “Federated square root filter for de-

centralized parallel processors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 3, pp. 517–525, 1990.

[62] X. Wang, T. Song, and Y. Wu, "Covering a mobile target using mobile sensor networks," in *Control and Decision Conference (CCDC), 2016 Chinese*. IEEE, 2016, pp. 1433–1437.

[63] H. M. La and W. Sheng, "Cooperative sensing in mobile sensor networks based on distributed consensus," in *Proc. Signal Data Process. Small Targets Conf*, 2011, pp. 81 370Y1–81 370Y14.

[64] S. S. Blackman, "Theoretical approaches to data association and fusion," in *Sensor Fusion*, vol. 931. International Society for Optics and Photonics, 1988, pp. 50–56.

[65] G. Pulford, "Taxonomy of multiple target tracking methods," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 291–304, 2005.

[66] M. Mallick, S. Coraluppi, and C. Carthel, *Multitarget tracking using multiple hypothesis tracking*. Wiley Online Library, 2016, pp. 163–203.

[67] C. Qiu, Z. Zhang, H. Lu, and H. Luo, "A survey of motion-based multitarget tracking methods," *Progress in Electromagnetics Research B*, vol. 62, pp. 195–223, 2015.

[68] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics (NRL)*, vol. 2, no. 1-2, pp. 83–97, 1955.

[69] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," in *The Foundations of the Digital Wireless World: Selected Works of AJ Viterbi*. World Scientific, 2010, pp. 41–50.

[70] S. Zhao, B. M. Chen, and T. H. Lee, "Optimal deployment of mobile sensors for target tracking in 2D and 3D spaces," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 1, pp. 24–30, 2014.

[71] K. Ma, Y. Zhang, and W. Trappe, "Managing the mobility of a mobile sensor network using network dynamics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 1, pp. 106–120, 2008.

[72] J. Hu, L. Xie, and C. Zhang, "Energy-based multiple target localization and pursuit in mobile sensor networks," *IEEE Transactions on Instrumentation and measurement*, vol. 61, no. 1, pp. 212–220, 2012.

[73] M. Yang, Y. Cao, L. Tan, and J. Yu, "An enhanced precise self-deployment algorithm in mobile sensor network," in *Information Science and Engineering, 2008. ISISE'08. International Symposium on*, vol. 2. IEEE, 2008, pp. 786–789.

[74] S. Ferrari, M. Anderson, R. Fierro, and W. Lu, "Cooperative navigation for heterogeneous autonomous vehicles via approximate dynamic programming," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 121–127.

[75] Y. Li, Y.-h. Liu, and X. Cai, "Local control strategy for target tracking in mobile sensor networks," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*. IEEE, 2007, pp. 674–679.

[76] P. Chattopadhyay, D. K. Jha, S. Sarkar, and A. Ray, "Path planning in GPS-denied environments: A collective intelligence approach," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 3082–3087.

[77] A. Gusrialdi, T. Hatanaka, and M. Fujita, "Coverage control for mobile networks with limited-range anisotropic sensors," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 4263–4268.

[78] Y. Li, Y.-h. Liu, H. Zhang, H. Wang, X. Cai, and D. Zhou, "Distributed target tracking with energy consideration using mobile sensor networks," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3280–3285.

[79] Y.-y. Li and Y.-h. Liu, "Tracking point or diffusing targets using mobile sensor networks under sensing noises," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 564–569.

[80] S. Hutchinson and T. Bretl, "Robust optimal deployment of mobile sensor networks," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 671–676.

[81] M. Rout and R. Roy, "Self-deployment of mobile sensors to achieve target coverage in the presence of obstacles," *IEEE Sensors Journal*, vol. 16, no. 14, pp. 5837–5842, 2016.

[82] D. K. Jha, P. Chattopadhyay, S. Sarkar, and A. Ray, "Path planning in GPS-denied environments via collective intelligence of distributed sensor networks," *International Journal of Control*, vol. 89, no. 5, pp. 984–999, 2016.

[83] G. Mathew, A. Surana, and I. Mezić, "Uniform coverage control of mobile sensor networks for dynamic target detection," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 7292–7299.

[84] L. Sun, S. Baek, and D. Pack, "Distributed probabilistic search and tracking of agile mobile ground targets using a network of unmanned aerial vehicles," in *Human Behavior Understanding in Networked Sensing*. Springer, 2014, pp. 301–319.

[85] E. Yanmaz and H. Guclu, "Stationary and mobile target detection using mobile wireless sensor networks," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*. IEEE, 2010, pp. 1–5.

[86] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[87] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[88] ——, "Distributed tracking for mobile sensor networks with information-driven mobility," in *American Control*

Conference, 2007. ACC'07. IEEE, 2007, pp. 4606–4612.

[89] H. M. La, T. H. Nguyen, C. H. Nguyen, and H. N. Nguyen, "Optimal flocking control for a mobile sensor network based a moving target tracking," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. IEEE, 2009, pp. 4801–4806.

[90] E. Khodayari, V. Sattari-Naeini, and M. Mirhosseini, "Flocking control with single-com for tracking a moving target in mobile sensor network using gravitational search algorithm," in *Swarm Intelligence and Evolutionary Computation (CSIEC), 2016 1st Conference on*. IEEE, 2016, pp. 125–130.

[91] H. M. La and W. Sheng, "Adaptive flocking control for dynamic target tracking in mobile sensor networks," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 4843–4848.

[92] ——, "Flocking control of a mobile sensor network to track and observe a moving target," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3129–3134.

[93] ——, "Flocking control algorithms for multiple agents in cluttered and noisy environments." *Studies in Computational Intelligence*, vol. 335, pp. 53–79, 2011.

[94] ——, "Moving targets tracking and observing in a distributed mobile sensor network," in *American Control Conference, 2009. ACC'09*. IEEE, 2009, pp. 3319–3324.

[95] ——, "Dynamic target tracking and observing in a mobile sensor network," *Robotics and Autonomous Systems*, vol. 60, no. 7, pp. 996–1009, 2012.

[96] D. Gu and H. Hu, "Distributed minimax filter for tracking and flocking," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3562–3567.

[97] Z. Tu, Q. Wang, H. Qi, and Y. Shen, "Flocking based distributed self-deployment algorithms in mobile sensor networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 3, pp. 437–449, 2012.

[98] Y. Wang, Z. Tu, Q. Wang, Y. Shen, and J. Li, "Flocking based distributed deployment for target monitoring in mobile sensor networks: Algorithm and implementation," in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*. IEEE, 2012, pp. 2472–2477.

[99] Z. Tu, Q. Wang, H. Qi, and Y. Shen, "Flocking based sensor deployment in mobile sensor networks," *Computer Communications*, vol. 35, no. 7, pp. 849–860, 2012.

[100] L. Zhang and Y. Zhu, "Mobile sensor deployment based on distributed flocking algorithm," in *Signal Processing, Communications and Computing (ICSPCC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–5.

[101] A. D. Dang and J. Horn, "A mobile sensor network tracking moving targets in a dynamic environment," *IFAC-PapersOnline*, vol. 48, no. 5, pp. 1–6, 2015.

[102] A. D. Dang, H. M. La, and J. Horn, "Distributed formation control for autonomous robots following desired shapes in noisy environment," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2016 IEEE International Conference on*. IEEE, 2016, pp. 285–290.

[103] S. Jiang, L. Dou, and H. Fang, "Target tracking based on federated filter for mobile sensor networks," in *Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference on*. IEEE, 2013, pp. 1263–1268.

[104] J. Tan, N. Xi, W. Sheng, and J. Xiao, "Modeling multiple robot systems for area coverage and cooperation," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 3. IEEE, 2004, pp. 2568–2573.

[105] C.-H. Kuo, T.-S. Chen, and S.-C. Syu, "Adaptive trap coverage in mobile sensor networks," *Procedia Computer Science*, vol. 110, pp. 102–109, 2017.

[106] Y. Xu, S. Salapaka, and C. L. Beck, "Dynamic maximum entropy algorithms for clustering and coverage control," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 1836–1841.

[107] Y. Qi, P. Cheng, J. Bai, J. Chen, A. Guenard, Y.-Q. Song, and Z. Shi, "Energy-efficient target tracking by mobile sensors with limited sensing range," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 6949–6961, 2016.

[108] Y. Li and Y.-H. Liu, "Energy saving target tracking using mobile sensor networks," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3653–3658.

[109] H. Mahboubi, A. Momeni, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Optimal target tracking strategy with controlled node mobility in mobile sensor networks," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 2921–2928.

[110] H. Mahboubi, W. Masoudimansour, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Maximum life span strategy for target tracking in mobile sensor networks," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 5096–5101.

[111] H. Mahboubi, W. Masoudimansour, A. G. Aghdam, and K. Sayrafian-Pour, "Cost-efficient routing with controlled node mobility in sensor networks," in *Control Applications (CCA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1238–1243.

[112] ——, "Maximum lifetime strategy for target monitoring with controlled node mobility in sensor networks with obstacles," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3493–3508, 2016.

[113] ——, "An energy-efficient target-tracking strategy for mobile sensor networks," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 511–523, 2017.

[114] Y. Zou and K. Chakrabarty, "Distributed mobility man-

agement for target tracking in mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, 2007.

[115] F. Mourad, H. Chehade, H. Snoussi, F. Yalaoui, L. Amodeo, and C. Richard, "Controlled mobility sensor networks for target tracking using ant colony optimization," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1261–1273, 2012.

[116] Z. Liao, J. Wang, S. Zhang, J. Cao, and G. Min, "Minimizing movement for target coverage and network connectivity in mobile sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 1971–1983, 2015.

[117] S. Sharma, "Target tracking technique in wireless sensor network," in *Computing, Communication & Automation (ICCCA), 2015 International Conference on*. IEEE, 2015, pp. 486–491.

[118] K. Selvaraj and S. Balaji, "Controlled mobility sensor networks for target tracking using particle swarm optimization," in *Current Trends in Engineering and Technology (ICCTET), 2013 International Conference on*. IEEE, 2013, pp. 388–391.

[119] Y. Fu and L. Yang, "Sensor mobility control for multitarget tracking in mobile sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, no. 278179, 2014.

[120] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, 1998.

[121] X. Gao, Z. Chen, F. Wu, and G. Chen, "Energy efficient algorithms for k-sink minimum movement target coverage problem in mobile sensor network," *IEEE/ACM Transactions on Networking*, Oct. 2017.

[122] S. Kamath, E. Meisner, and V. Isler, "Triangulation based multi target tracking with mobile sensor networks," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3283–3288.

[123] D. Zorbas and T. Razafindralambo, "Prolonging network lifetime under probabilistic target coverage in wireless mobile sensor networks," *Computer Communications*, vol. 36, no. 9, pp. 1039–1053, 2013.

[124] V. Marbukh, K. Sayrafian-Pour, H. Mahboubi, A. Momeni, and A. G. Aghdam, "Towards evolutionary-pricing framework for mobile sensor network self-organization," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[125] L. Liu, Y. Sun, and Z. Wang, "CTCOMSN: Collaborative target coverage optimization in mobile sensor networks," no. 533. IET Conference Publications, Dec. 2007, pp. 636–639.

[126] K. H. Low, W. K. Leow, and M. H. Ang Jr, "Task allocation via self-organizing swarm coalitions in distributed mobile sensor network," in *AAAI*, vol. 4, 2004, pp. 28–33.

[127] K. H. Low, W. K. Leow, and M. H. Ang, "Reactive, distributed layered architecture for resource-bounded multi-robot cooperation: Application to mobile sensor network coverage," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3747–3752.

[128] ——, "Autonomic mobile sensor network with self-coordinated task allocation and execution," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 36, no. 3, pp. 315–327, 2006.

[129] L.-L. Hung, "Efficient algorithms for sensor detachment in WMSNs," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 16, no. 3, pp. 172–182, 2014.

[130] C.-Y. Chang, G. Chen, G.-J. Yu, T.-L. Wang, and T.-C. Wang, "TCWTP: Time-constrained weighted targets patrolling mechanism in wireless mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 6, pp. 901–914, 2015.

[131] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[132] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.

[133] H. Wei and S. Ferrari, "A geometric transversals approach to sensor motion planning for tracking maneuvering targets," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2773–2778, 2015.

[134] K. A. Baumgartner, S. Ferrari, and A. V. Rao, "Optimal control of an underwater sensor network for cooperative target tracking," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 4, pp. 678–697, 2009.

[135] H. Park and S. Hutchinson, "Worst-case performance of a mobile sensor network under individual sensor failure," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 895–900.

[136] Y.-Y. Chen, C.-C. Hsu, C.-F. Chou, and K. Lin, "On detecting mobile target with deadline constraint in mobile sensor networks," in *Proceedings of IEEE Sensors*, no. 6411422. IEEE, 2012.

[137] C.-C. Hsu, Y.-Y. Chen, C.-F. Chou, and L. Golubchik, "On design of collaborative mobile sensor networks for deadline-sensitive mobile target detection," *IEEE Sensors Journal*, vol. 13, no. 8, pp. 2962–2972, 2013.

[138] K. Imai and T. Ushio, "Effective combination of search policy based on probability and entropy for heterogeneous mobile sensors," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1981–1986.

[139] J.-C. Chin, Y. Dong, W.-K. Hon, C. Y.-T. Ma, and D. K. Yau, "Detection of intelligent mobile target in a mobile sensor network," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 1, pp. 41–52, 2010.

[140] S. Ferrari, R. Fierro, B. Perteet, C. Cai, and K. Baumgartner, "A geometric optimization approach to detect-

ing and intercepting dynamic targets using a mobile sensor network," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 292–320, 2009.

[141] S. J. Russell and P. Norvig, "Artificial intelligence: a modern approach (international edition)," 2002.

[142] K. M. Krishna, H. Hexmoor, S. Pasupuleti, and S. Chellappa, "A surveillance system based on multiple mobile sensors." in *FLAIRS Conference*, 2004, pp. 128–133.

[143] J. Takahashi, K. Sekiyama, and T. Fukuda, "Cooperative object tracking with mobile robotic sensor network," *Distributed Autonomous Robotic Systems. Springer, Berlin*, pp. 51–62, 2009.

[144] M. J. A. Rahman, A. I. Abu-El-Haija, and H. M. Al-Najjar, "On the detection of intelligent mobile targets in a mobile sensor network," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*. IEEE, 2011, pp. 1268–1275.