IUST Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Research Paper

Multiple Target Tracking in Wireless Sensor Networks Based on Sensor Grouping and Hybrid Iterative-Heuristic Optimization

M. H. Adhami* and R. Ghazizadeh*(C.A.)

Abstract: A novel hybrid method for tracking multiple indistinguishable maneuvering targets using a wireless sensor network is introduced in this paper. The problem of tracking the location of targets is formulated as a Maximum Likelihood Estimation. We propose a hybrid optimization method, which consists of an iterative and a heuristic search method, for finding the location of targets simultaneously. The Levenberg-Marquardt (LM) algorithm is used for iterative search, while the Particle Swarm Optimization (PSO) is used for the heuristic search. We use the maximum sensors separating distance-grouping algorithm (G-MSSD), which was introduced in our previous work, to generate initial guesses for search algorithms. The estimates of both methods are compared and the best one is selected as the final estimation. We demonstrate the accuracy and performance of our new tracking method via simulations and compare our results with the Gauss-Newton (GN) method.

Keywords: Maximum Likelihood, Multi-Target Tracking, Multiple Target Tracking, Simultaneous Tracking, Wireless Sensor Network.

1 Introduction

IN the last two decades, significant advances have been made in telecommunication, microprocessors, battery-operated devices, and computer networks. Such advances have enabled us to design large-scale networks containing a lot of sensors, capable of communicating wirelessly. These networks are called wireless sensor networks (WSN). WSNs have interesting potential in diverse applications including industry, healthcare, military, home automation, environmental monitoring, and surveillance, which makes them a hot research topic in recent years.

Among the various applications of WSNs, tracking (localization) of targets is one of the most important and strategic ones. Tracking is the instantaneous determination of some pre-specified states (e.g., velocity, location) of moving targets. Two main approaches have been adopted for estimating the location of targets within a sensor network: Range-Based and Range-Free. In a range-free approach, the distance between the source node and the destination is estimated by the number of hops and size of the hops [1]. Different range-based measurement techniques have been used for tracking the location of targets, e.g. time of arrival (TOA) [2, 3], time difference of arrival (TDOA) [4], angle of arrival (AOA) [5, 6], and received signal strength indicator (RSSI) [7-9]. Some mixed measurements are also used in the literature [10, 11]. Time-based techniques, TOA/TDOA, require strict time measurements and synchronizations. TOA may impose an additional cost on the nodes, by demanding an extremely precise clock or impose complexity on the network by requiring a complicated synchronization process. Moreover, TDOA-based techniques dissipate the energy or time of the nodes by requiring two types of transceivers [13]. Sophisticated array antennas which are extremely sensitive to the fluctuation of signal strength and the multipath effect

Iranian Journal of Electrical and Electronic Engineering, 2023.
Paper first received 30 April 2021, revised 08 September 2022, and accepted 12 September 2022.
* The authors are with the Faculty of Electrical Engineering and Computer, University of Birjand, Birjand, Southern Khorasan, Iran.
E-mails: mohamad.ad@birjand.ac.ir and rghazizade@birjand.ac.ir.
Corresponding Author: R. Ghazizadeh.
https://doi.org/10.22068/IJEEE.19.1.2165

are required for angle-based measurement techniques [14]. Hence, the RSSI measurement is the simplest (cheapest) method among these measurement methods. Utilizing RSSI requires less time /energy, and reduces the complexity of nodes in both software and hardware. Accordingly, we use RSSI in our work because of its competitive advantages.

1.1 Literature Review

Tracking several moving targets, because of its amazing potential, has been attracting enormous attention, recently.

A wide range of studies conducted in the field of target tracking only discuss tracking a single target over a specified area. These types of studies are categorized under the single-target tracking (STT) problem [7], [14-18]. Nevertheless, most practical applications require tracking more than one target at the same time; hence our focus is on tracking multiple targets, and categorized under the multi-target tracking (MTT) problem. MTT is a complex multi-faceted problem that should not be considered as a simple extension of the STT problem. Different critical issues must be addressed in an MTT scenario. Hence, different aspects of the MTT e.g., deployment of sensors, selection of sensors, data association, energy consumption, movement model of targets, state estimation, utilizing filters, identifying different targets, and movement of the network or agents have been investigated in the literature. Energy consumption is a critical issue in most applications of WSN, including tracking that much research has been made to manage it. In [19] and [20], controlling the activation periods (sleeping-awakening) of the nodes, using different methods, is proposed to manage the energy. The information about the targets' movement is used to control the activation schedule of the nodes. In [21], adaptive dynamic programming scheduling is used for sensors to decrease the energy consumption during the tracking process. In [22] an artificial bee colony-inspired algorithm is used to manage the clustering process and save more energy to prolong the lifetime of the network. Although energy management has an impressive role in WSNs, it is not the key issue in tracking applications, and there are other important issues like accuracy of estimation, the number of targets, simultaneity of the tracking process, and being real-time.

In [23], a network with mobile sensor nodes is assumed for tracking multiple moving targets. The authors propose a controlling algorithm which specifies the activation cycle and location of the network nodes. Also, there are a number of studies assume moving agents with the ability of tracking particular targets. These agents pursue the specified targets in order to track their state [9]. A marine environment is considered in [24] where the multi-target tracking problem is formulated in MLE format. The security issue is also addressed by Byzantine attack. Such moveable nodes/agents increase the flexibility and accuracy of the system at the cost of increment in complexity and expense of the equipment, so we ignore them.

In some rare studies, algorithms are proposed for recovering the signal of each distinct target from the received signal. For example, authors in [25] have introduced the blind source separation algorithm to estimate the location of targets using the recovered signals. Information about the shape (type) of the signals must be available in advance. In [9], each agent uses the data of the previous step of tracking to cancel the effect of far targets on its received signal. When targets are far apart from one another, each agent can estimate the power received from its nearby target and run a separate tracking process. Limiting requirements like information about signal shapes, moving agents, and complicated signal recovering algorithms, are serious obstacles for practical implementation of such studies.

Authors in [26] have considered a highly dense network of sensor nodes; in [27], and [28] authors have considered a grid deployment for the network as well, in which each node has an equal distance to its four closest neighbors. Such considerations are not realistic in practical tracking applications, especially for randomly deployed networks. In contrast, we assume a network with normal density and random arrangement of nodes.

There are studies that have tried to model the motion pattern of the targets, and then estimate the parameters of that model. In [29], targets can be distinguished from each other, and groups of sensors are dynamically selected to track the targets. A discrete Markov process used as the basic model for the movement of targets. Authors in [30] choose a Gauss-Markov mobility model and try to determine its parameters in a way to properly match the movement pattern of the targets. Sensor nodes have the ability to measure the velocity vector of the targets, whenever targets enter the detection zone of them. Approaches for tracking multiple targets, using binary or quantized data, are proposed in the literature [26, 31].

Data association with the help of magnetic sensors has been discussed in [32]. The main focus of these studies is on managing the gathered data and adjusting the quantization level during the tracking process.

The main subject of some studies can be summarized in estimating the state (location) of targets. The algorithm of [33], uses the collected RSSI measurements to run an ML-based estimator. The tracking problem is formulated in a matrix form for better scalability. Three methods: exhaustive search, Multi-Resolution (MR) search, and Expectation-Maximization (EM) have been proposed to solve the problem. In [34], correlated noise is considered as well, and an algorithm is introduced for localizing a single acoustic source in the monitoring area. The authors derive the likelihood function in the free space environment, and they use the MR-search find the optimum of that function. In [35] a modified least squares algorithm is used to estimate the state of the moving targets. The authors assumed sensors capable of obtaining both the distance and angle from the targets.

Some works apply different filters on the movement model of mobile targets and try to determine the current or estimate the next state of them [36-39]. A combination of the Kalman filter and particle swarm optimization is proposed in [40] for tracking. Although filters improve the accuracy of tracking, they increase the complexity and delay of the tracking procedure.

1.2 Motivation

By investigating several studies, we found some critical limitations in the related existing works, the most important ones are:

- Researchers usually consider sensor nodes that can sense signals coming from anywhere in the surveillance area, which cannot be justified for large-scale networks.
- Accuracy is usually improved at the expense of increased complexity and cost arising from complex measurement techniques such as AOA or DOA.
- A pre-specified model is considered for targets' movement and filters are applied to find the current or next location of targets.
- Targets or at least their signals are assumed to be distinguishable.
- Only one type of optimization method (heuristic or non-heuristic) to estimate the desired states of the moving targets, which suffer from 'local minimum' traps.

Considering all the above points, we design a new hybrid-tracking algorithm, which enables estimating the location of multiple indistinguishable mobile targets simultaneously, based only on their power measurements (RSSI values). Our new algorithm is the same as our previous method [41] but it takes the advantage of a heuristic search algorithm when it is possible.

This work uses RSSI measurements reported by nodes and formulates a matrix-form maximum likelihood estimation (MLE) for tracking targets. As stated in [41], a closed-form solution is not accessible for the MLE problem of multiple target tracking. Therefore, an approximate solution must be found using numerical methods. Hence, in the previous work, we used a modified version of the Gauss-Newton (GN) algorithm to find the optimum solution but in the current work, we use the Levenberg-Marquardt algorithm. In section 4, we will show that using the LM can slightly improve the accuracy of the estimation in comparison to GN. The LM has two essential advantages: 1) it is more robust than the GN, 2) it finds a better solution even if it starts far from the global minimum [43].

The initial guess for the LM algorithm is provided using the G-MSSD algorithm which is introduced in detail in [35]. The G-MSSD algorithm divides nodes into groups in a way that each group only tracks its closest target. The number of groups equals the number of existing targets. In each group, the location of the node with the largest measurement (MaxLoc) is selected as an initial location. Putting all initial locations into a vector, an initial guess for the LM algorithm is provided.

1.3 Contribution

We have noticed that in our previous work, sometimes the final estimate is not accurate enough and is far from the optimal global solution. This happens especially when the number of node reports (triggered nodes) is low and subsequently a poor initial guess is provided.

When the number of reporting nodes is small, the optimization objective function (in the MLE problem) is very likely to fall into local minima traps, regardless of the type of optimization algorithm used. Therefore, the need for an auxiliary algorithm with a broader and more detailed search capability is felt next to the main algorithm. Non-heuristic algorithms (such as LM and GN) usually have a higher convergence speed, but they are very sensitive to the initial guess. Instead, heuristic algorithms usually have a larger search range and are less sensitive to the initial guess. The combination of these two types of optimization algorithms can impressively improve the accuracy of the final solution by utilizing the capabilities of both types. In our simulations, we will show that using a heuristic algorithm alone does not improve the tracking and increases the estimation error.

Therefore, we decided to utilize a heuristic optimization algorithm (PSO) as the auxiliary search algorithm in order to improve the accuracy of the final estimation. We use PSO because of its advantages [45], which are simple concept, easy implementation, robustness to control parameters, and computational efficiency when compared with other heuristic optimization techniques.

When the estimations of both algorithms are derived, we select the best one as the final estimation of the targets' location. To select the best estimate a criterion is required and we use sum (or mean) of the squared residuals as the comparing criterion. The output of the method with the smallest criterion value is selected as the targets' location estimate.

We also use the initial guess (MaxLoc) as a guide start point to determine the search area for the PSO algorithm. We define a search radius and draw a circle with this radius around each initial location; each circle is called a search region. Then the initial population of the PSO is generated inside these search regions. This helps to reduce the computational complexity of the system since the PSO explores a much smaller area rather than the whole surveillance zone. On the other hand, by controlling the search radius of PSO we allow the algorithm to adjust the area of its search zone and, therefore, the chance of convergence to the global optimum increases.

Using LM and PSO at the same time increases the computational complexity. Nevertheless, it has an impressive performance, especially in terms of accuracy. However, we should not worry about the excessive computational complexity since the algorithm is centralized and all the computations are done in the Sink node. No additional traffic is imposed on the network because all the computations are done using the same reported measurements initially sent by the triggered nodes and no new reports or measurements are required. Besides, LM and PSO can be implemented in parallel in order to reduce the delay of the tracking system. Therefore, the inherent features of the proposed algorithm solve all the disadvantages of excessive computational complexity like delay, battery depletion, network lifetime, traffic, cost overhead, and congestion.

The main contributions of the paper can be summarized as:

- Tracking multiple indistinguishable maneuvering targets simultaneously.
- Using only the cheap/simple RSSI measurements that do not require any extra equipment.
- Combining the advantages of both heuristic (PSO) and non-heuristic (LM) optimization algorithms to increase the accuracy of tracking.
- Limiting the search zone of the heuristic search algorithm, which in turn increases the convergence speed and decreases the computational complexity.
- The ability of parallel implementation for better efficiency

1.4 System Model and Assumptions

The inherent features and capabilities of this work make it suitable for outdoor applications e.g., battlefield surveillance, automotive Supervision, suburban traffic management, wildlife monitoring, and domestic animal farming. Therefore, we apply the exponential path loss model which is justified for such applications. We also consider the following assumptions and definitions (as in our previous work [41]) throughout this paper:

a) Sensors only measure the received signal power, and if the measured power is greater than a threshold (receiver sensitivity), the sensor is called a "triggered sensor".
b) Sensor nodes send their collected measurements to the Sink, and they do not perform any extra processes.
c) The maximum possible distance between the sensor and a target (when there is only one target in the surveillance area), while the sensor can detect that target's signal is called the sensing range.
d) The signals of different targets cannot be distinguished from each other.
e) The total number of targets existing within the surveillance area is fixed and known in advance.
f) Each sensor knows: 1) its location, 2) all other sensors' locations, and 3) the distance between each pair of sensors. This information is provided during the initialization (localization) phase.
g) Sensor nodes have an activation-inactivation schedule with time step T to manage the energy consumption.

The remainder of this paper is organized as follows: In Section 2, we formulate the measurement model and the ML estimator, the LM algorithm is presented in this section as well. The details of the hybrid tracking algorithm are expressed in Section 3, while the simulation results are presented in Section 4.

2 Model Formulation

Sensor nodes are assumed to be randomly dropped across the whole surveillance area. Targets are assumed to be indistinguishable, hence, each sensor node measures a signal which is the sum of all targets' signals plus a measurement noise.

2.1 Measurement Model

Let $\boldsymbol{l}_{si} = \begin{bmatrix} x_{si} & y_{si} \end{bmatrix}^T$, $\boldsymbol{l}_{tk} = \begin{bmatrix} x_{tk} & y_{tk} \end{bmatrix}^T$, $\boldsymbol{l_t} = \begin{bmatrix} \boldsymbol{l}_{t1}^T & \cdots & \boldsymbol{l}_{tK}^T \end{bmatrix}^T$ represent the location vector of the i-th sensor, the k-th target, and all targets respectively ($[.]^T$ is the matrix or vector transpose). The total number of sensors is assumed to be N_s and there exist K targets within the surveillance area. Assume that the k-th (k = 1, 2, …, K) target transmits a continuous zero-mean signal with average power p_k. Since the path loss model is exponential [42] with path loss exponent η, the signal power received from the k-th target at the location of i-th sensor node, p_{ik}^r, can be derived as

$$p_{ik}^r = p_k \left(d_0^\eta \,/\, d_{ik}^\eta \right) \tag{1}$$

where d_0 is a reference distance (usually set to 1m). Assuming targets with indistinguishable signals, the received signal by the i-th sensor is the sum of received signal from all targets. Therefore, assuming d_0 = 1 meter, the received signal power at the i-th sensor location, p_i^r, can be expressed as

$$p_i^r = \sum_{k=1}^{K} p_{ik}^r = \sum_{k=1}^{K} p_k \,/\, d_{ik}^\eta \tag{2}$$

where $d_{ik} = \left\| \boldsymbol{l}_{si} - \boldsymbol{l}_{tk} \right\|$ is the Euclidian distance between target k and sensor i, and ||.|| is the 2-norm of a vector. The sensing range of a sensor node is defined as d_{ik}^{max}.

Now using (2), the measurement of the i-th sensor node, y_i, can be modeled as

$$y_i = p_i^r + v_i \tag{3}$$

where $v_i \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$ is the measurement noise of the i-th sensor node. All nodes are assumed to have the same sensitivity $p_{\min}$, hence if $y_i \geq p_{\min}$ node i considered a triggered node otherwise it does not participate in tracking. Let the number of triggered nodes be N.

To have a simpler notation we use vector/matrix notation and formulation. We also use standardized measurements; hence the standardized measurement of the i-th sensor node, z_i, is defined as

$$z_i \triangleq \left(y_i - \mu_i\right) / \sigma_i \tag{4}$$

where μ_i and σ_i are the mean and standard deviation of the measurement noise respectively. The standardized noise of the i-th sensor is defined as

$$w_i \triangleq \left(v_i - \mu_i\right) / \sigma_i \rightarrow w_i \sim \mathcal{N}\left(0,1\right) \tag{5}$$

Let's define the following operator, which places a series of M related scalars/vectors x_j on the elements/columns of an arbitrary vector/matrix.

$$vect\left(\boldsymbol{x}_j, j, M\right) \triangleq \left[\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \dots \quad \boldsymbol{x}_M\right]^T \tag{6}$$

Using (6), we can define the distance vector of the i-th sensor and the transmitting power vector of all targets as

$$\mathbf{d}_i \triangleq vect\left(1/d_{ik}^{\eta}, k, K\right), \qquad \mathbf{p} \triangleq vect\left(p_k, k, K\right) \tag{7}$$

Using (4)-(7), the standardized measurement of the i-th sensor in (4) can be written as

$$z_i = \mathbf{d}_i^T \mathbf{p} / \sigma_i + w_i \tag{8}$$

The vector of all standardized measurements can be expressed as

$$\mathbf{z} = \mathbf{\Sigma D p} + \mathbf{w} \tag{9}$$

where $\mathbf{z} = vect(z_i, i, N)$, $\mathbf{w} = vect(w_i, i, N)$, $\mathbf{D} = vect(\mathbf{d}_i, i, N)$, and $\mathbf{\Sigma}_{N\times N} = \text{diag}[1/\sigma_1 \dots 1/\sigma_N]$.

Now, to find the location of the targets we explain two methods based on the likelihood maximization in the following parts. Then the Hybrid tracking algorithm is introduced in the next section.

2.2 Maximum Likelihood Estimation

Assuming that the measurement noise of nodes is uncorrelated, the likelihood function of $\boldsymbol{l}_t$ can be derived as

$$L\left(\boldsymbol{l}_t\right) = \left(2\pi\right)^{-N/2} \exp\left(-\left\|\mathbf{z} - \mathbf{\Sigma D p}\right\|^2 / 2\right) \tag{10}$$

Maximizing the likelihood function (10), after discarding constant terms, leads to minimizing the following cost (fitness) function

$$l\left(\boldsymbol{l}_t\right) = \left\|\mathbf{z} - \mathbf{\Sigma D p}\right\|^2 / 2 = \left\|\mathbf{r}\right\|^2 / 2 \tag{11}$$

According to [41], a closed-form solution for the minimization of (11) cannot be found and the problem must be solved using numerical methods. Minimizing (11) can be considered as a standard non-linear least-squares problem [44] and appropriate methods can be applied to obtain an approximate solution. We prefer to use the Levenberg-Marquardt [47, 48] algorithm which is more robust than the Gauss-Newton [43] method. The LM usually finds a better solution even if it starts far from the global minimum of the function. The LM tends to be a bit slower than the GN but it obtains more accurate solutions. The LM interpolates between the GN and the method of gradient-descent. The LM, like the GN, does not use the second-order derivatives, which reduces the complexity and computational load of the algorithm.

Like other Newton-based optimization methods, the LM finds the final solution through an iterative process

$$\boldsymbol{l}_t^{n+1} = \boldsymbol{l}_t^n + \alpha_n \boldsymbol{h}_n^{LM}, \ n \geq 0 \tag{12}$$

where $\boldsymbol{h}_n^{LM}$ is the search direction at the n-th iteration and α_n is a controlling parameter called step size. The search direction of the LM algorithm [49] at the n-th iteration is

$$\boldsymbol{h}_n^{LM} = -\left(\mathbf{J}_n^T \mathbf{J}_n + \lambda \mathbf{I}\right)^{-1} \mathbf{J}_n^T \mathbf{r}_n, \ n \geq 0 \tag{13}$$

where $\mathbf{J}_n$ is the Jacobean matrix of the residuals vector $\mathbf{r}$ with respect to the vector of unknown parameters $\boldsymbol{l}_t^n$, and according to [41] calculated as

$$\mathbf{J}_n = -\eta \mathbf{\Sigma} \mathbf{\Delta}_n^T \mathbf{P}_d \tag{14}$$

where $\mathbf{\Delta} = vect\left(\boldsymbol{\delta}_i, i, N\right)$, $\boldsymbol{\delta}_i = vect\left(\boldsymbol{\delta}_{ik}^T / d_{ik}^{\eta+2}, k, K\right)$, $\boldsymbol{\delta}_{ik} = \left(\boldsymbol{l}_{si} - \boldsymbol{l}_{tk}\right)$, $\mathbf{P}_d^1 = diag\left(p_1, \dots, p_k\right)$, $\mathbf{P}_d = \mathbf{P}_d^1 \otimes \mathbf{I}_q$, $\otimes$ is the tensor product, $\mathbf{I}_q$ is the $q\times q$ identity matrix, and q is the dimension of the problem space. For a 3D space $q = 3$, and for a 2D space $q = 2$.

Remark: It should be mentioned that to be able to estimate the location of K targets uniquely in a space with q dimensions using least squares (11), the number of triggered sensors (measurements) must satisfy $N \geq K(q+1)$. For example to locate a single target in a 2D space, at least 3 triggered nodes are required [51].

In (13), λ is the damping factor that should be adjusted at each iteration [42]. When the reduction of function is fast smaller values can be used and vice versa. Larger λ values bring the algorithm closer to the GN while smaller ones bring it closer to the gradient descent.

We use the delayed gratification method [49] to adjust the damping factor λ. First, select an initial guess $\hat{\boldsymbol{l}}_t^0$, initial damping factor λ_0 and two adjusting factors v_u, $v_d \geq 1$. Using (11), compute the fitness value $l\left(\hat{\boldsymbol{l}}_t^n\right)$ after one step from the starting point firstly with the damping factor $\lambda = \lambda_0$ and secondly with $\lambda = \lambda_0/v_d$. If both of these are worse than the initial guess (the fitness value is larger), then the damping factor is increased by v_u as $\lambda = \lambda_0 v_u$. Otherwise, the damping factor $\lambda = \lambda_0/v_d$ is selected. The procedure is continued and at each step, if the step is uphill ($l(\hat{\boldsymbol{l}}_t^{n+1}) \leq l(\hat{\boldsymbol{l}}_t^n)$) the current value of damping factor λ is decreased by v_d and if the step is downhill ($l(\hat{\boldsymbol{l}}_t^{n+1}) > l(\hat{\boldsymbol{l}}_t^n)$) the current damping factor λ is increased by v_u.

Using $(\mathbf{J}_n^T\mathbf{J}_n + \lambda\mathbf{I})$ in LM algorithm has the disadvantage that if λ is large, inverting $\mathbf{J}_n^T\mathbf{J}_n + \lambda\mathbf{I}$ is not used at all. Therefore, a more reliable version of the LM algorithm is used which replaces the identity matrix with diagonal elements of $\mathbf{J}_n^T\mathbf{J}_n$. In this version of the LM [49], the search direction is derived as

$$\boldsymbol{h}_n^{LM} = -\left(\mathbf{J}_n^T\mathbf{J}_n + \lambda diag\left(\mathbf{J}_n^T\mathbf{J}_n\right)\right)^{-1}\mathbf{J}_n^T\mathbf{r}_n,\ n \geq 0 \tag{15}$$

Due to the iterative nature of the PSO algorithm, conditions must be defined that will terminate the algorithm in a reasonable time if they are met. We adopt two stopping conditions as follows

a) Total number of iterations: if the number of iterations reaches a maximum acceptable number $n_{\max}$, the algorithm stops.
b) Relative difference of two Sequential iterations: if $\|\boldsymbol{l}_t^n - \boldsymbol{l}_t^{n-1}\|/\boldsymbol{l}_t^n$ becomes smaller than a threshold, the algorithm stops.

2.3 PSO Algorithm

Although the LM can provide appropriate estimates of targets' locations, in some critical situations it cannot provide estimates with the acceptable accuracy. Therefore, we decide to utilize an auxiliary method to increase the accuracy in these situations. Although using another search method can improve the accuracy, the computational load of the system will be increased. However, there is no need to worry about this because algorithms can be executed completely independently. In other words, the computational load of the system can be split among parallel computing agents and only the results are compared.

The PSO [46] starts with an initial population of particles $\{\mathbf{x}_j^0\}$, $j = 1, \ldots, N_{pop}$ and then update the location of each target according to its velocity.

$$\mathbf{x}_j^{n+1} = \mathbf{x}_j^n + \mathbf{v}_j^{n+1} \tag{16}$$

where $\mathbf{v}_j^n$ is the velocity of the j-th particle a the n-th iteration and updated as

$$\mathbf{v}_j^{n+1} = \omega^n\mathbf{v}_j^n + c_1r_1\left(\boldsymbol{P}_j^{best} - \mathbf{x}_j^n\right) + c_2r_2\left(\boldsymbol{G}^{best} - \mathbf{x}_j^n\right) \tag{17}$$

The initial velocity of all particles is assumed to be zero. $\boldsymbol{P}_j^{best}$ is the best location of the j-th particle until the current iteration, while $\boldsymbol{G}^{best}$ is the best location among all particles until the current iteration. r_1 and r_2 are two independent random variables in the range of 0 and 1. c_1 and c_2 denote the relative importance of the memory of the particle itself to the memory of the swarm. The value of c_1 and c_2 are usually assumed to be 2. ω^n is the inertia weight which dampens the velocities over time (iterations) and enabling the swarm to converge more efficiently. The inertia weight must be decreased over time and different strategies can be adopted but we prefer the linear one which updates the weight as $\omega^n = \omega_{\max} - (\omega_{\max} - \omega_{\min})n/n_{\max}$, where $\omega_{\max}$ and $\omega_{\min}$ are the initial and final weights respectively, and $n_{\max}$ is the maximum number of iterations.

The stopping criteria are the same as the LM algorithm.

3 Hybrid Tracking Algorithm

The first step of our proposed hybrid-tracking algorithm is to group the triggered sensor nodes. Therefore, we first describe the grouping algorithm. Then we explain how the search regions are prepared for the PSO algorithm, and at last, we explain the whole procedure of the hybrid-tracking algorithm in details. Fig. 1 shows all steps of the proposed hybrid multiple target tracking algorithm.

3.1 Grouping Algorithm

As mentioned before, we want to group the triggered sensor nodes into K distinct groups. Every single group is expected to belong to a single distinct target. Hence, a candidate location can be derived from the location information of each distinct group. This candidate location can be an initial guess for the location of the corresponding target.

The G-MSSD algorithm introduced in [38], works based on the distance information of the triggered sensor nodes. The main idea is that when the nodes have limited sensing range, not all network nodes are triggered, and consequently, there will be a group of triggered nodes around each target. Therefore, the algorithm starts with the K farthest triggered nodes and put them in K distinct groups. Then the algorithm check all other ungrouped nodes. Each remained ungrouped triggered node is assigned to the group to whose first member it has the shortest distance. This means that the algorithm tries to group the nodes spatially, based on the maximum distances between them.

Two candidate locations were proposed, **MaxLoc** and **MeanLoc**. **MaxLoc** is the location of the group member whose measurement is the greatest among other group

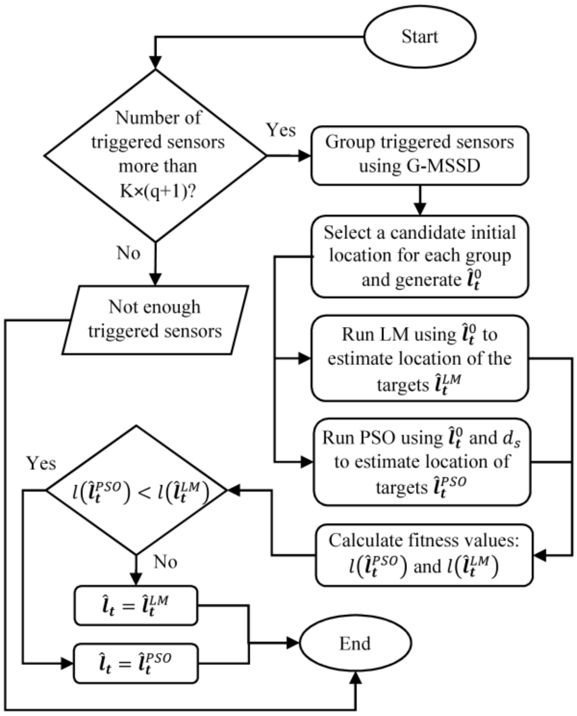


Fig. 1 The flowchart of the hybrid-tracking algorithm.

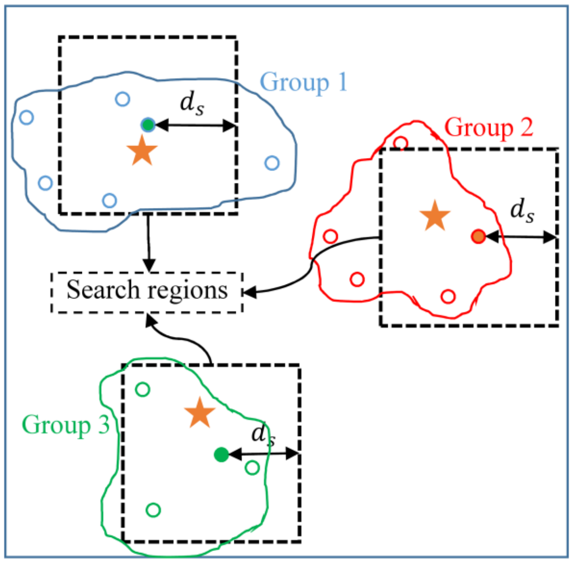


Fig. 2 The procedure of generating search regions for the PSO algorithm. Stars, and circles, respectively, show targets, and sensor nodes. Solid circle shows the node with greatest measurement of each group.

members, while **MeanLoc** is the mean location of all the group members. In this work, we use only the **MaxLoc** candidate as an initial guess for the LM and PSO algorithms.

If the initial location of target k is $\hat{\boldsymbol{l}}_{tk}^0$, the initial location vector of all targets is $\hat{\boldsymbol{l}}_t^0 = vect(\hat{\boldsymbol{l}}_{tk}^0, k, K)$.

3.2 Search Region of PSO

The original PSO algorithm generates the initial population randomly in the whole possible range of the variables. This helps the algorithm to exploit the whole feasible area however if any initial information is available about the solution it decreases the computational load of the system. Therefore, we define a search region around the initial guess to reduce the searching area and accordingly decrease the computational complexity of the PSO.

A search distance d_s is defined and a square cell is drawn around each initial location. Now K cells are obtained, and the PSO can search only inside these cells. This means that the possible range for each particle is $[\hat{\boldsymbol{l}}_t^0 - d_s, \hat{\boldsymbol{l}}_t^0 + d_s]$. At each iteration, the algorithm checks the range of the updated particles and if any of them is out of the range, the algorithm returns it (randomly) to the acceptable range. The procedure of generating search regions is illustrated in Fig. 2.

The value of d_s must be selected properly in order to increase the chance of searching the regions close to the true location of targets. If d_s set too large, the computational load and the chance to fall into local traps increases. On the other hand, if d_s set too small it is possible to miss the true location of the targets inside the search region; this means that the search region may not include the true location of a target and the algorithm cannot converge to the accurate estimate. Since the maximum distance between a single target and triggered sensor node is equal to the sensing range of that node, we set d_s equal to d_{ik}^{max} with a confidence margin as $d_s = \rho d_{ik}^{max}, \rho \geq 1$.

3.3 Hybrid Tracking

When the estimates of both LM and PSO are prepared, the tracking algorithm compares their fitness to select the best one. The estimate with the smaller fitness value is selected as the final estimate. Assume that the estimate of the LM is denoted by $\hat{\boldsymbol{l}}_t^{LM}$, and the estimate of PSO is denoted by $\hat{\boldsymbol{l}}_t^{PSO}$. The algorithm calculates the fitness of $\hat{\boldsymbol{l}}_t^{LM}$ and $\hat{\boldsymbol{l}}_t^{PSO}$ using (11). If $l(\hat{\boldsymbol{l}}_t^{PSO}) < l(\hat{\boldsymbol{l}}_t^{LM})$ the algorithm selects $\hat{\boldsymbol{l}}_t^{PSO}$ as the final estimate and if $l(\hat{\boldsymbol{l}}_t^{PSO}) \geq l(\hat{\boldsymbol{l}}_t^{LM})$, $\hat{\boldsymbol{l}}_t^{LM}$ is selected as the final estimate. Fig. 1 shows the complete procedure of the hybrid multiple target tracking algorithm.

Although **MaxLoc** can provide an appropriate initial guess for the LM and PSO, during the tracking process the information of the previous time steps can be used to provide more suitable initial guesses. If there exists an appropriate estimate for the current time step of the tracking process, this estimate can be used as an initial guess for the subsequent time step. Even the mean of the estimates of two or more previous steps can be utilized as the initial guess. However, we prefer to use only the estimate of the previous step which is called **PriorLoc**. It should be noted that **PriorLoc** can only be used in a tracking process where there exists an appropriate estimate for the current time step. For example, at the first time step of the tracking, it is not possible to use **PriorLoc**. On the other hand, during the tracking, there would be some consecutive steps that the algorithm cannot provide proper estimates (because of the low number of triggered sensor nodes or local minimum traps) and consequently, **PriorLoc** cannot be used to generate an initial guess.

3.4 Computational Complexity

Three parameters affect the computational complexity of the proposed tracking algorithm: the number of targets (K), space dimensionality (q), and the number of triggered nodes (N). We consider the complexity in terms of number of mathematical operations. We only count the number of multiplications (*MLPs*) and ignore other operations for simplicity [52]. The required number of *MLPs* for the operations: (a) matrix inversion, (b) matrix product, and (c) exponentiation must be calculated. Although novel efficient methods are introduced to execute the mentioned operations, we check the traditional methods to consider the worst case.

Let A and B be $m\times n$ and $n\times r$ matrices respectively, then the matrix product AB requires mnr *MLPs*. If A is a $n\times n$ square matrix, its inversion using Gauss-Jordan elimination requires $2n^3-n^2+1$ *MLPs*. Calculation of a^b, using Taylor series expansion approximation, requires $(4n_c-2)$ *MLPs* if only n_c first terms of the series are used for approximation.

Assuming that n_λ iterations are required to find the optimum λ, the total required number of *MLPs* for one iteration of the LM algorithm is calculated as:

$$N^2+\left[2(Kq)^2+(3q+4n_c)K+1\right]N$$
$$+2(n_\lambda+1)(Kq)^3+(n_\lambda+2)Kq+2q+n_\lambda+5, \qquad (18)$$

while, the required number of *MLPs* for each iteration of the PSO algorithm is $N_{pop}(2q+3)$.

Table 1 Simulation parameters.

Parameter	Value
Transmission power p_k	0 dBm
Receiver sensitivity $p_{\min}$	-27 dBm
The standard deviation of noise σ_w	-40 dBm
Sensing range (free space) $d_{max}^{(1)}$	22 m
Path loss exponent η	2.2
Time step	0.5 sec

4 Simulations

In this part, we explain the simulation scenarios and present the simulation results. All simulations are done using MATLAB 2018b. A surveillance area with dimensions of 110(m)×110(m) is assumed and 65 sensor nodes are randomly deployed over it. All simulations use the simulation parameters listed in Table 1 except when mentioned. All simulations are performed using the Monte Carlo method and the results are averaged over 2000 iterations. Two different scenarios are considered to better demonstrate the performance of the proposed hybrid-tracking algorithm. The Levenberg-Marquardt parameters are $v_u = 2.5$, $v_d = 1.5$, $\lambda_0 = 10^{-6}$, and the maximum number of iterations bounded to 500. The PSO population number is 15, $d_s = 0.4\times d_{\max}$ and $n_{\max} = 250$.

4.1 Uncorrelated Movement

In this section, it is assumed that each target has the same distance d_{tar} to its closest target. Targets appear randomly on the perimeter of a circle whose radius is set in a way that the distance between every two adjacent target equals d_{tar}. The location of the first target (at each time step) is randomly generated on the perimeter of the circle and the locations of other targets are generated at equal angles from the first target's location. This means, targets are placed at $2\pi/K$ angle from each other on the perimeter of the circle and the distance between every two adjacent target is d_{tar}. In this scenario, the current location of a target is completely uncorrelated from the previous time steps.

The estimation error of each target's location is calculated as $E_k = \left\|\boldsymbol{l}_{tk} - \hat{\boldsymbol{l}}_{tk}\right\|$. Hence, the average error of all targets' location estimates is $AEE \triangleq (1/K)\sum_k E_k$, and the least error of estimates is $LEE \triangleq \min_k E_k$.

At first, we calculate the average number of required

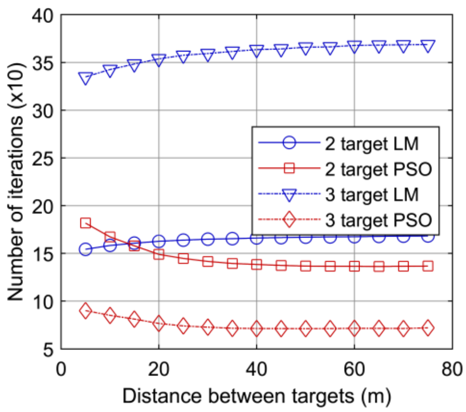


Fig. 3 The average number of required iterations of the LM and PSO for the two and three target scenario.

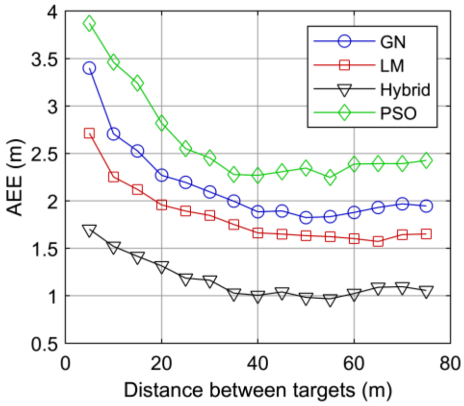


Fig. 4 The AEE of two targets versus distance between targets.

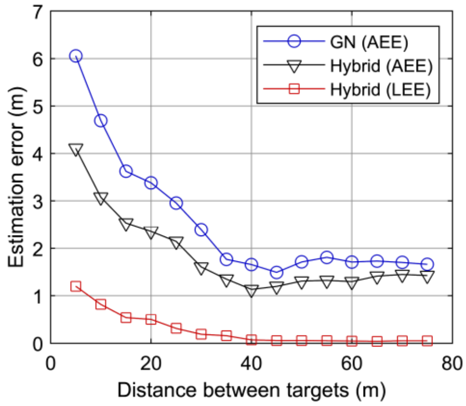


Fig. 5 The AEE and LEE for the three-target scenario.

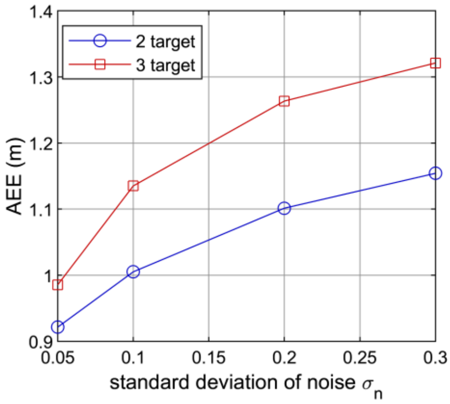
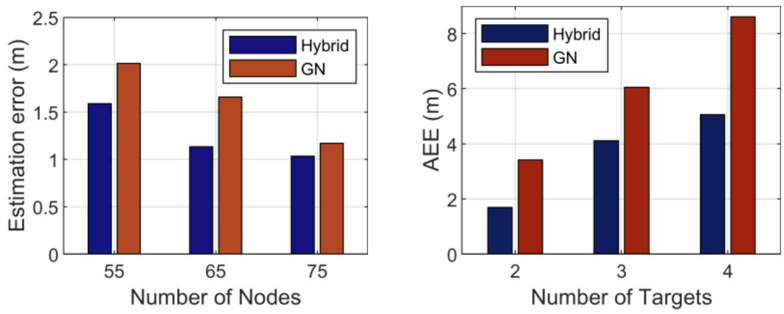


Fig. 6 The effect of noise variance on the performance of the proposed hybrid method (d_{tar} = 40 m).



(a) (b)

Fig. 7 The effect of sensor density and the number of targets on the performance of the proposed method; a) AEE versus the number of sensor nodes and b) AEE versus the number of targets.

iterations for both LM and PSO to converge. As shown in Fig. 3, by increasing the distance between targets, the iterations of slightly increased while the iterations of PSO decreased. For the 2 target scenario, in average, 150 and 165 iterations are required for the LM and PSO respectively. For the three target scenario, 70 and 350 iterations are required for the LM and PSO respectively.

Fig. 4 shows the average error (AEE) versus different distances between two targets. As illustrated, the proposed hybrid method can effectively outperform the GN and LM methods. The interesting point is that although the error of PSO is high, combining it with the LM method can significantly reduce the error. The error of estimation is decreasing with distance because when the distance between targets increases, the G-MSSD can better group the triggered nodes and subsequently the initial guess is closer to the true location of each target.

Fig. 5 shows the error for three-target scenario, and as can be observed, the AEE decreased by the distance between targets. Again, the hybrid method completely outperforms the GN method. Although the AEE is high for short distances between targets, the LEE is very small.

To see the effect of the measurement noise on the performance of the proposed method, we repeat the simulations for different σ_n and fixed distance of 40 meters between targets. The results are shown in Fig. 6. As expected, increasing the noise variance increases the AEE.

The effect of the number of sensor nodes (network density) and the number of targets is illustrated in Fig. 7. Here, the distance between targets is 40 meters. As can be observed in Fig. 7(a), by decreasing 10 nodes from the network the AEE increases 25% (on average) when the hybrid method is used while for the GN method the incensement is about 31%. Therefore, the new hybrid method is less sensitive to the network node density on average. Fig. 7(b) shows that by increasing the number of targets the error increases, while using the hybrid method can at least decrease the error by 30%.

4.2 Correlated Movement

In this section, we assume that targets are moving on arbitrary paths and the distance between them is not (necessarily) the same at each time step. The current location of a target is correlated to the previous time steps. Each target has a predefined but unknown movement path. Since targets have paths for their movement, **PriorLoc** can also be used for tracking. We assume that targets are moving based on the coordinated turn (CT) model [50] with the following dynamic equation:

$$\mathbf{x}_j^{n+1} = \boldsymbol{F}_{ct}\left(\varphi_t\right)\mathbf{x}_j^n + \boldsymbol{\omega}_j^n,\ j = 1, \ldots, K \tag{19}$$

where $\boldsymbol{\omega}$ is the zero mean white Gaussian noise which models the perturbation of the trajectory from the ideal CT motion. φ_t is the known turn rate of movement and $\boldsymbol{F}_{ct}(\varphi_t)$ is the transition matrix of the model calculated as:

$$\boldsymbol{F}_{ct}\left(\varphi_t\right) = \begin{bmatrix} 1 & \frac{\sin\left(\varphi_t\right)}{\left(\varphi_t\right)} & 0 & -\frac{1-\cos\left(\varphi_t\right)}{\left(\varphi_t\right)} \\ 0 & \cos\left(\varphi_t\right) & 0 & -\sin\left(\varphi_t\right) \\ 0 & \frac{1-\cos\left(\varphi_t\right)}{\left(\varphi_t\right)} & 1 & \frac{\sin\left(\varphi_t\right)}{\left(\varphi_t\right)} \\ 0 & \sin\left(\varphi_t\right) & 0 & \cos\left(\varphi_t\right) \end{bmatrix} \tag{20}$$

$\mathbf{x} = [x\ v_x\ y\ v_y]^T$ is the state vector of each target which include the location coordinates $[x\ y]^T$ and the velocity along each coordinate axis $[v_x\ v_y]^T$.

We assume a fixed turn rate $\varphi_t = \pi/70$ rad/sec. The initial location of targets are [30, 80], [60, 20], [80,70], and the initial velocity vectors are [1, -2], [-1, 2], and [-2, 1]. Fig. 8 shows one realization of the moving scenario. Dashed lines are the real trajectories of targets, and the asterisk lines are the estimated trajectories of the targets obtained by the proposed hybrid algorithm.

We compare the accuracy of our proposed hybrid tracking algorithm with the GN method and the results are shown in Fig. 9, as illustrated, our method can decrease the error about 62 percent whether PriorLoc is used or not. Fig. 8 shows that using PriorLoc can effectively decrease the error of tracking.

Additionally, to show the effectiveness of our proposed algorithm, we have compared our method with MR-search method [34] and Variational Bayesian Expectation-Maximization (VEM) method [53] in terms the AEE of tracking 3 targets moving according to the scenario of Fig. 8. The results are shown in Fig. 10. MR-search, initially, generates an $n_1 \times n_1$ grid covering the whole surveillance area, and assign each target one arbitrary point from this grid. Putting the coordinate vector of all targets in a single vector, it produces a candidate vector, which presents the estimated location of all targets. MR generates as many candidate vectors as possible (by assigning different points of the grid to targets), and search among the generated candidate vectors to reach the best one. At step i, MR makes a $n_i \times n_i$ grid around each location of the previous best candidate vector to increase the accuracy of the estimation. We choose a two-level MR-search with $n_1 = 15$ and $n_2 = 7$.

VEM makes an $n_g \times n_g$ equi-spaced grid covering the surveillance area. Assuming an initial guess for targets' location, it calculates the probable RSS measurement of each sensor node. Then at each iteration, the algorithm changes the grid lines using Expectation Maximization steps in order to reach the best estimate for the location of targets. Since the number of existing targets within the surveillance area is smaller than the total number of grid points, VEM can benefit from compressed sensing. We run the VEM with $n_g = 8$ grid lines and set the maximum iterations of the EM steps to 100.

As depicted in Fig. 10, the proposed hybrid-tracking algorithm outperforms both MR and VEM methods. Our proposed hybrid method has approximately 80% and 70% less AEE in comparison to MR and VEM, respectively.

5 Conclusion and Future Work

In this paper, a new hybrid multiple target tracking algorithm is introduced which uses a combination of the Levenberg-Marquardt method and the particle swarm optimization to find the maximum likelihood estimation. The new method can outperform the Gauss-Newton method, Multi-Resolution search, and Variational Bayesian Expectation-Maximization method in accuracy and performance. Using a heuristic search

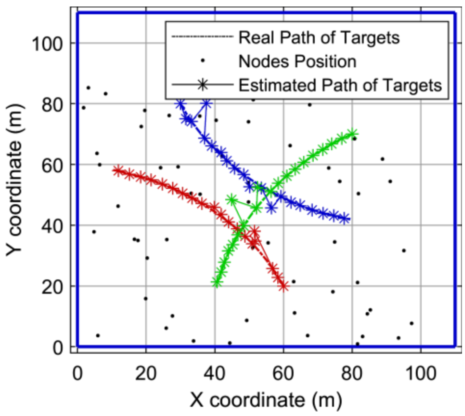


Fig. 8 One realization of the tracking scenario for three targets, the movement model is coordinated turn rate with known fixed rate $\pi/70$ rad/sec.

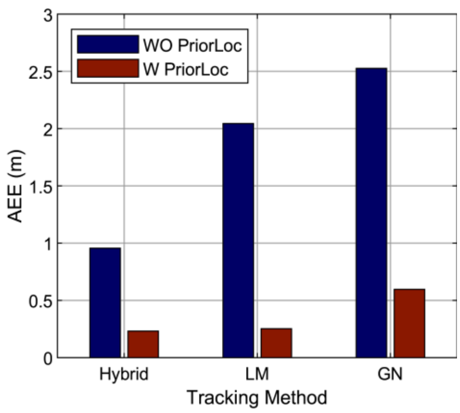


Fig. 9 The average error of tracking three moving targets with and without PriorLoc usage.

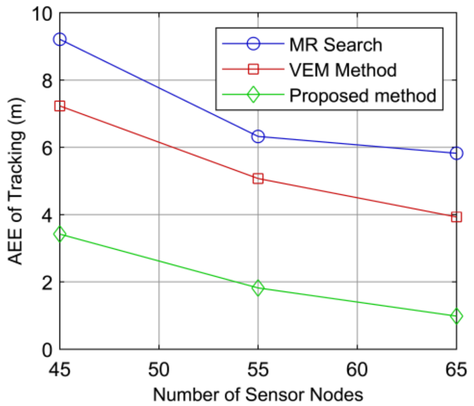


Fig. 10 The average error of tracking three targets with CT movement model, for MR search method, VEM method and the proposed hybrid method.

algorithm along with iterative search helps the tracking system to get out of the local minimum traps and consequently improve the accuracy of the tracking.

5.1 Future Work

Although the proposed hybrid tracking method prepares accurate estimates for the location of targets, there would be some time steps in which the number of triggered nodes is not enough and the algorithm may fail to achieve appropriate estimates. Hence, we suggest using a filter (e.g., particle filter, Kalman filter) to provide predictions for these time steps. Additionally, using a filter can provide a proper initial guess for the iterative search algorithm based on the prediction. Also, using the prediction of the filter, one can generate a new search region (around the predicted location of a target) for the heuristic search algorithm.

Intellectual Property

The authors confirm that they have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property.

Funding

No funding was received for this work.

CRediT Authorship Contribution Statement

M. H. Adhami: Idea & conceptualization, Research & investigation, Revise & editing. **R. Ghazizadeh:** Analysis, Supervision, Revise & editing.

Declaration of Competing Interest

The authors hereby confirm that the submitted manuscript is an original work and has not been published so far, is not under consideration for publication by any other journal and will not be submitted to any other journal until the decision will be made by this journal. All authors have approved the manuscript and agree with its submission to “Iranian Journal of Electrical and Electronic Engineering”.

References

[1] L. Liu, T. Cui, and W. Lv, “A range-free multiple target localization algorithm using compressive sensing theory in wireless sensor networks,” in *IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, Philadelphia, PA, USA, 2014.

[2] Y. Kang, Q. Wang, J. Wang, and R. Chen, “A high-accuracy TOA-based localization method without time synchronization in a three-dimensional space,” *IEEE Transactions on Industrial Information*, Vol. 15, No. 1, pp. 173–182, 2019.

[3] S. Xu, L. Wu, K. Doğançay, and M. Alaee-Kerahroodi, “A hybrid approach to optimal TOA-sensor placement with fixed shared sensors for simultaneous multi-target localization,” *IEEE Transactions on Signal Processing*, Vol. 70, pp. 1197–1212, 2022.

[4] G. Wang, A. M. C. So, and Y. Li, “Robust convex approximation methods for TDOA-based localization under NLOS conditions,” *IEEE Transactions on Signal Processing*, Vol. 64, No. 13, pp. 3281–3296, 2016.

[5] R. Zhang, J. Liu, X. Du, B. Li, and M. Guizani, “AOA-based three-dimensional multi-target localization in industrial WSNs for LOS conditions,” *Sensors*, Vol. 18, No. 8, p. 2727, 2018.

[6] A. Alexandridis and A. Mouchtaris, “Multiple sound source location estimation in wireless acoustic sensor networks using DOA estimates: the data-association problem,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 26, No. 2, pp. 342–356, 2018.

[7] E. Masazade and A. Kose, “A proportional time allocation algorithm to transmit binary sensor decisions for target tracking in a wireless sensor network,” *IEEE Transactions on Signal Processing*, Vol. 66, No. 1, pp. 86–100, 2018

[8] M. Mansouri, L. Khoukhi, H. Nounou, and M. Nounou, “Secure and robust clustering for quantized target tracking in wireless sensor networks,” *Journal of Communications and Networks*, Vol. 15, No. 2, pp. 164–172, 2013.

[9] 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.

[10] Y. Y. Li, G. Q. Qi, and A. D. Sheng, “Performance metric on the best achievable accuracy for hybrid TOA/AOA target localization,” *IEEE Transactions on Wireless Communications*, Vol. 22, No. 7, pp. 1474–1477, 2018.

[11] J. Yin, Q. Wan, S. Yang, and K. C. Ho, “A simple and accurate TDOA-AOA localization method using two stations,” *IEEE Signal Processing Letters*, Vol. 23, No. 1, pp. 144–148, 2016.

[12] A. K. Paul and T. Sato, “Localization in wireless sensor networks: a survey on algorithms, measurement techniques, applications and challenge,” *Journal of Sensor and Actuator Networks*, Vol. 6, No. 4, 2017.

[13] A. Mesmoudi, M. Feham, and N. Labraoui, “Wireless sensor networks localization algorithms: a comprehensive survey,” *International Journal of Computer Networks & Communications (IJCNC)*, Vol. 5, No. 6, pp. 45–64, 2013

[14] J. Feng, B. Lian, and H. Zhao, “Coordinated and adaptive information collecting in target tracking wireless sensor networks,” *IEEE Sensors Journal*, Vol. 15, No. 6, pp. 3436–3445, 2015.

[15] Y. Yu, “Distributed target tracking in wireless sensor networks with data association uncertainty,” *IEEE Communications Letters*, Vol. 21, No. 6, pp. 1281–1284, 2017.

[16] S. R. Jondhale and R. S. Deshpande, “Kalman filtering framework-based real time target tracking in wireless sensor networks using generalized regression neural networks,” *IEEE Sensors Journal*, Vol. 19, No. 1, pp. 224–233, 2019.

[17] Z. Li, L. Zhang, Y. Cai, and H. Ochiai, “Sensor selection for maneuvering target tracking in wireless sensor networks with uncertainty,” *IEEE Sensors Journal*, Vol. 22, No. 15, pp. 15071–15081, 2022.

[18] W. Xia, M. Sun, and Q. Wang, “Direct target tracking by distributed Gaussian particle filtering for heterogeneous networks,” *IEEE Transactions on Signal Processing*, Vol. 68, pp. 1361–1373, 2020.

[19] J. A. Fuemmeler and V. V. Veeravalli, “Energy efficient multi-object tracking in sensor networks,” *IEEE Transactions on Signal Processing*, Vol. 58, No. 7, pp. 3742–3750, 2010.

[20] F. Liu, C. Jiang, and W. Xiao, “Multistep prediction-based adaptive dynamic programming sensor scheduling approach for collaborative target tracking in energy harvesting wireless sensor networks,” *IEEE Transactions on Automation Science and Engineering*, Vol. 18, No. 2, pp. 693–704, 2021.

[21] F. Liu, C. Jiang, and W. Xiao, “Multistep prediction-based adaptive dynamic programming sensor scheduling approach for collaborative target tracking in energy harvesting wireless sensor networks,” *IEEE Transactions on Automation Science and Engineering*, Vol. 18, No. 2, pp. 693–704, 2021.

[22] A. Pathak, “An artificial bee colony inspired clustering solution to prolong lifetime of wireless sensor networks,” *Iranian Journal of Electrical and Electronic Engineering*, Vol. 16, No. 4, pp. 425–438, 2020.

[23] X. Zhang, “Adaptive control and reconfiguration of mobile wireless sensor networks for dynamic multi-target tracking,” *IEEE Transactions on Automatic Control*, Vol. 56, No. 10, pp. 2429–2444, 2011.

[24] J. Xian, H. Wu, X. Mei, Y. Zhang, H. Chen, and J. Wang, “NMTLAT: A new robust mobile multi-target localization and tracking scheme in marine search and rescue wireless sensor networks under Byzantine attack,” *Computer Communications*, Vol. 160, pp. 623–635, 2020.

[25] Y. Zhu, A. Vikram, H. Fu, and Y. Guan, “On non-cooperative multiple-target tracking with wireless sensor networks,” *IEEE Transactions on Wireless Communications*, Vol. 13, No. 11, pp. 6496–6510, 2014.

[26] M. P. Michaelides, C. Laoudias, and C. G. Panayiotou, “Fault tolerant localization and tracking of multiple sources in WSNs using binary data,” *IEEE Transactions on Mobile Computing*, Vol. 13, No. 6, pp. 1213–1227, 2014.

[27] Y. K. An, C. An, S. Yoo, and B. E. Wells, “Noise mitigation for multiple target tracking in acoustic wireless sensor networks,” in *IEEE Military Communications Conference*, Baltimore, MD, USA, 2014.

[28] J. Teng, H. Snoussi, and C. Richard, “Collaborative multi-target tracking in wireless sensor networks,” *International Journal of Systems Science*, Vol. 42, No. 9, pp. 1427–1443, 2011.

[29] Y. Fu, Q. Ling, and Z. Tian, “Distributed sensor allocation for multi-target tracking in wireless sensor networks,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, No. 4, pp. 3538–3553, 2012.

[30] B. Liu, M. Chen, and M. Tsai, “Message-efficient location prediction for mobile objects in wireless sensor networks using a maximum likelihood technique,” *IEEE Transactions on Computers*, Vol. 60, No. 6, pp. 865–-878, 2011.

[31] T. Nguyen, F. Septier, H. Rajaona, G. Peters, I. Nevat, and Y. Delignon, "A Bayesian perspective on multiple source localization in wireless sensor networks," *IEEE Transactions on Signal Processing*, Vol. 64, No. 7, pp. 1684–1699, 2016.

[32] H. Chen, R. Wang, L. Cui, and L. Zhang, "EasiDSlT: A two-layer data association method for multitarget tracking in wireless sensor networks," *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 1, pp. 434–443, 2015.

[33] X. Sheng and Y. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Transactions on Signal Processing*, Vol. 53, No. 1, pp. 44–53, 2005.

[34] E. Dranka and R. Coelho, "Robust maximum likelihood acoustic energy based source localization in correlated noisy sensing environments," *IEEE Journal of Selected Topics in Signal Processing*, Vol. 9, No. 2, pp. 259–267, 2014.

[35] X. He, T. Wang, W. Liu, and T. Luo, "Measurement data fusion based on optimized weighted least-squares algorithm for multi-target tracking," *IEEE Access*, Vol. 7, pp. 13901–13916, 2019.

[36] A. Soderlund and M. Kumar, "Optimization of multitarget tracking within a sensor network via information-guided clustering," *Journal of Guidance Control and Dynamics*, Vol. 42, No. 2, pp. 1–18, 2018.

[37] M. Beard, B. T. Vo, and B. N. Vo, "Bayesian multi-target tracking with merged measurements using labelled random finite sets," *IEEE Transactions on Signal Processing*, *Vol.* 63, No, 6, pp. 1433–1447, 2015.

[38] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi, "Non-parametric and semi-parametric RSSI/distance modeling for target tracking in wireless sensor networks," *IEEE Sensor Journal*, Vol. 16, No. 7, pp. 2115–2126, 2016.

[39] R. Havangi, "A new modified particle filter with application in target tracking," *Iranian Journal of Electrical and Electronic Engineering*, Vol. 16, No. 4, pp. 449–460, 2020.

[40] D. P. Bhagata and H. Soni, "Target tracking in a wireless sensor network using a multi-step KF-PSO model," *International Journal of Computers and Applications*, Vol. 43, No. 5, pp.1–12, 2019.

[41] M. H. Adhami, R. Ghazizade, and M. H. Majidi, "Dual-mode multiple-target tracking in wireless sensor networks based on sensor grouping and maximum likelihood estimation," *International Journal of Communication Systems*, Vol. 33, No. 8, pp. e4352, 2020.

[42] A. Goldsmith, *Wireless communications*. New York USA: Cambridge University Press, 2005.

[43] P. C. Hansen, V. Pereyra, and G. Scherer, *Least squares data fitting with applications*. Baltimore Maryland USA: The John Hopkins University Press, 2013.

[44] T. Kelley, *Iterative methods for optimization*. Philadelphia USA: Society for Industrial and Applied Mathematics, 1999.

[45] K. Y. Lee and J. B. Park, "Application of particle swarm optimization to economic dispatch problem: advantages and disadvantages," in *IEEE PES Power Systems Conference and Exposition*, Atlanta, GA, USA, 2006.

[46] S. S. Rao, *Engineering optimization theory and practice*. Hoboken New, Jersey USA: John Wiley & Sons, Inc., 2009.

[47] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, Vol. 2, No. 2, pp. 164–168, 1944.

[48] D. W. Marquardt, "An algorithm for least squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, pp. 431–441, 1963.

[49] M. K. Transtrum and J. P. Sethna, "Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization," *arXiv: 1201.5885*, 2012.

[50] X. R. Li and V. P. Jikov, "Survey of maneuvering target tracking: Part I: Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 39 No. 4, pp. 1333–1364, 2003.

[51] R. Kapoor, S. Ramasamy, A. Gardi, C. Bieber, L. Silverberg, and R. Sabatini, "A novel 3D multilateration sensor using distributed ultrasonic beacons for indoor navigation," *Sensors*, Vol. 16, No. 10, pp. 1–13, 2016.

[52] G. Allaire and S. M. Kaber, *Numerical linear algebra*. New York, USA: Springer Science & Business Media, 2008.

[53] B. Sun, Y. Guo, N. Li, and D. Fang, "Multiple target counting and localization using variational Bayesian EM algorithm in wireless sensor networks," *IEEE Transactions on Communications*, Vol. 65, No. 7, pp. 2985–2998, 2017.

M. H. Adhami received the B.Sc. from Islamic Azad University, Najafabad Branch, Iran, in 2009, and the M.Sc. from Isfahan University of Technology, Iran, in 2012, and the Ph.D. from University of Birjand, Iran, in 2021, all in Telecommunication Engineering. His recent research interests include wireless sensor networks, multi-target tracking, internet of things, and smart controlling.

R. Ghazizadeh received the B.Sc. and M.Sc. degrees in Electrical Engineering from Ferdowsi University of Mashhad, Iran, in 1992 and 1996, respectively, and the Ph.D. degree in Telecommunication Engineering from the Southwest JiaoTong University, China, in 2009. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran. His current research interests include radio resource management, analysis and optimization in the next-generation wireless networks, and wireless sensor networks.