IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 27, 2022, accepted April 13, 2022, date of publication April 25, 2022, date of current version May 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3170477

Distributed Target Tracking in Sensor Networks by Consistency Algorithm and Semantic Moving Computing of Internet of Things

YUN WANG[1], HAN LIU[2], JUN ZHOU[3,4], AND YI WAN[5]

[1]Department of Computer Engineering, Shanxi Polytechnic College, Taiyuan 030006, China
[2]College of Quality and Technical Supervision, Hebei University, Baoding 071002, China
[3]School of Artificial Intelligence, Chongqing Business Vocational College, Chongqing 401331, China
[4]Southwest Technology and Engineering Research Institute, Chongqing 401331, China
[5]School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China

Corresponding author: Jun Zhou (hgzhou2008@163.com)

This work was supported in part by the "Research on Key Technologies of Iris Recognition Based on Wavelet Packet Analysis" Project of Science and Technology Research Program of Chongqing Education Commission of China under Grant KJZD-K202004401, and in part by the Artificial Intelligence Stage Achievement of Application Collaborative Innovation Center of Chongqing Business Vocational College.

ABSTRACT The target tracking algorithm of mobile wireless sensor networks involves target motion trend prediction and subsequent node guidance. This study aims to solve the problems of global consistency of node information and significant errors in forecasting fast-moving targets' trajectories through traditional distributed tracking methods in sensor networks. Initially, the average consistency algorithm is used to average the local measurements of each node to achieve global consistency. Then, semantic moving computing of the Internet of Things calculates and analyzes the node movement to support the subsequent movement guidance of nodes and target movement prediction. Finally, the simulation experiment is carried out to evaluate the commonly used target trajectory prediction model. The simulation results show that the node movement algorithm by average consistency can effectively improve the positioning accuracy of the network for moving targets. Besides, the positioning error decreases with the increase of the sensing radius R, the number of moving nodes nm, and the total number of nodes ns deployed in a particular range in a two-dimensional (2D) space. The positioning error after node movement in 2D space is about 20%−30%R lower than that in a static state. After node movement in a three-dimensional (3D) space, the positioning error is about 40%−50%R lower than in a dormant state. When the target moves at a speed greater than 7m/s, the consistency-based moving computing algorithm's target loss rate and tracking errors are about 0∼10% and 1.5%∼2% lower than the target tracking algorithm via Kalman Filter. Therefore, the algorithm reported here can precisely track the high-speed moving target. The existing research on point target tracking has problems of insufficient accuracy and robustness. The algorithm proposed here has stronger robustness, reduced data error in multi-node, and more flexible node movements, providing a reference for the subsequent research on distributed point target tracking.

INDEX TERMS Consistency algorithm, moving computing algorithm, mobile wireless sensor network, distributed target tracking.

I. INTRODUCTION

The rapid development of sensors, video equipment, and semiconductor technology has improved sensor networks. Target tracking is a critical sensor network application to obtain the state information of moving targets immediately and accurately [1]. Target tracking is crucial in many aspects, such as civil or military navigation, timekeeping, and tracking. Among the technologies around target tracking, the distributed tracking technology of point targets under complex backgrounds has attracted increasing attention and research because of its extensive application range [2]. The distributed tracking method has the characteristics of easy establishment and high fault tolerance to node failure. It is suitable for

The associate editor coordinating the review of this manuscript and approving it for publication was Rahim Rahmani.

multi-node sensor networks requiring big data processing. Mobile wireless sensor networks are composed of decentralized moving nodes. Each node has traditional static nodes' sensing, computing, and communication capabilities and additional mobility. Nodes can be arranged independently without attaching to other objects [3]. The entire wireless mobile sensor network needs to move with the targets and keep them visible to track moving targets. This feature distinguishes the dynamic target tracking network from traditional ones [4]. There are two data processing frameworks for wireless sensor networks: centralized and distributed. The distributed framework has low communication consumption and substantial stability and robustness to a node or link failure [5]. Therefore, the distributed framework has attracted much attention [6]. There are two main data processing methods in the distributed framework: consistency strategy and cooperative diffusion strategy. The distributed algorithm via consistency strategy may fail to work due to communication delay or intermittent network connection. Therefore, this paper studies the distributed target tracking problem in sensor networks with consistency policy optimized by semantic IoT node moving computing.

There is little research on the target tracking problem of a nonlinear dynamic system using the distributed framework. The average consistency algorithm is used to de alienate the data of each node and enhance its robustness. Then, the algorithm is combined with the node movement algorithm to predict the movement trend of nodes and targets in 2D and 3D space, which provides a reference for the study of node movement. Two algorithms are used to optimize the standard target trajectory prediction model. The simulation results show a decrease in the model's target loss and tracking error. This paper provides a specific significant reference for the research of distributed target tracking in mobile wireless sensor networks.

II. RELATED WORK

At present, the Kalman Filter algorithm is one of the most commonly used algorithms to deal with the time-domain noise of real-time online target tracking. The classical Kalman Filter algorithm is a linear equation based on state and searches primarily for point target tracking in a linear system. However, nonlinear systems are more common in practical applications than linear systems [7]. Yu *et al.* proposed a distributed extended Kalman Filter algorithm based on optimizing the classical Kalman filter algorithm for nonlinear systems. Still, this algorithm introduced linearization error, resulting in limitations such as slow convergence speed [8]. Wang and Ren put forward a distributed mixed information filtering algorithm by combining extended information filtering with the covariance intersection algorithm [9]. With the development of wireless networks and computing devices, the Distributed Particle Filter algorithm eliminates dynamic systems' linear and Gaussian constraints. Besides, it is simpler and easier to operate than the traditional filter algorithms [10]. The primary challenge of mobile wireless sensor networks is to keep the global estimation provided by each node consistent through the local measurement of each node and the communication interaction between adjacent nodes. Consistency theory is a communication-based cooperative control theory for multi-intelligent systems. The consistency algorithm ensures that each node achieves a consistent information interaction rule. It can solve a series of problems covering control and state estimation in wireless mobile sensor networks. The commonly used consistency algorithms include Raft, Paxos, and Practical Byzantine Fault Tolerance (PBFT) algorithms [11]. Zaidi *et al.* proposed two data processing methods of discrete change and continuous change in the consistency of multi-individual information under dynamic topology. They also developed a consistency algorithm via the distributed observer for the multi-agent system with time-varying communication topology. They completed the construction through a new framework guiding the consistent synchronization of multi-agent and complex networks [12]. Thomson *et al.* combined the consistency algorithm and potential game theory to obtain the optimal joint decision matrix for collaborative control by mapping each state of the intelligent system [13]. Venâncio *et al.* explored how to eliminate the excessive dependence of software-defined networks on the centralized controller and the limited availability and scalability of the controller. They adopted the Paxos algorithm to ensure the strong consistency between multiple controllers in the software-defined network. In this scheme, the controller could perform its control plane activities without the high consumption tasks required to maintain consistency. The experimental results showed that this scheme could effectively reduce costs and improve the actual benefits, especially in low controller overhead [14]. Hu and Liu reported that the Raft consistency algorithm was one of the typical consistency algorithms in distributed systems to manage the consistency of log replication. Besides, it had the same function as the Paxos algorithm. However, the Raft algorithm was easier to understand and apply to practical systems than Paxos [15]. Li *et al.* proposed an optimized consistency algorithm based on PBFT. Aiming at the shortcomings of PBFT, such as unable to dynamically join nodes, low consistency efficiency of multiple nodes, and selection of controller nodes, the authors designed a hierarchical structure to improve scalability and consistency efficiency. The simulation results showed that compared with the traditional PBFT algorithm and RAFT algorithms, the optimized consistency algorithm supported more nodes, improved the data throughput, and effectively reduced the consistency delay and the number of communication between nodes [16].

Semantic technology can solve the heterogeneous problems of data, protocols, and systems in the IoT. Jiang *et al.* proposed a semantic gateway framework, which could provide public interfaces for many heterogeneous devices and networks to add semantics to the data of terminal nodes and translate interconnection protocols [17]. Semantic IoT refers to a new network that associates the semantic web with the IoT and associates semantic information with natural objects,

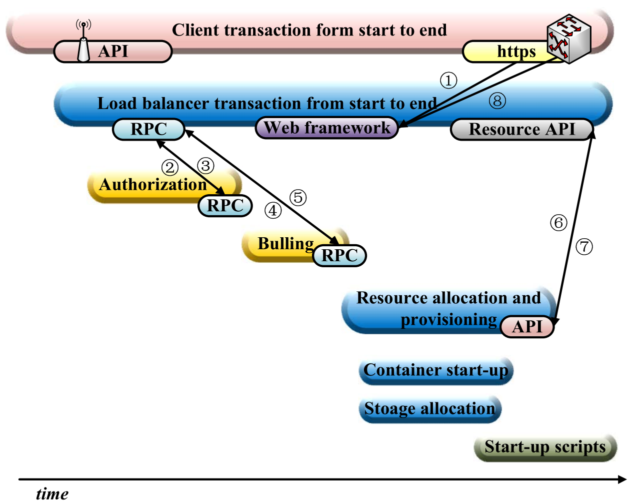

FIGURE 1. Invocation time and relationship of components in distributed target tracking.

locations, events, and other state information. Formal semantics carries out data exchange in a machine-understandable way, and the semantically annotated data can also be found, integrated, and used [18]. IoT devices communicate with the cloud platform and provide computing, analysis, and storage near the terminal at the edge node of the communication network. The actual data is annotated and supplied to the terminal node through the semantic gateway. The node resets the state of the edge node through the moving algorithm to continuously track the motion tracking target according to the changing network edge.

III. NODE MOVEMENT MODEL CONSTRUCTION AND PARAMETER SETTING BY AVERAGE CONSISTENCY FOR THE SIMULATION EXPERIMENT

A. BASIC PRINCIPLE OF DISTRIBUTED TARGET TRACKING

There are parallel execution tracks in the distributed tracking process, which is a directed acyclic graph composed of multiple spans. Each span represents a continuous execution segment named and timed in tracking. In distributed target tracking, each component usually contains one or more spans. FIGURE 1 illustrates each element's composition relationship, including the component's calling time and calling correlation.

FIGURE 1 shows the authorization between clients and load balancers, the invocation time between remote calls and assemblers, the hierarchical relationship between services, and the serial/parallel invocation relationship between processes and tasks. These are conducive to finding and tracking the critical path of the system invocation. Analyzing the execution process of the essential way can optimize the crucial position in the course, maximize the system performance, and enhance the target tracking efficiency.

B. AVERAGE CONSISTENCY ALGORITHM

Sensors with different observation performances usually receive various echoes. Thus, this paper selects the consistency principle of local average to locally average the data after the interaction of sensor network measurement information and obtain the average node measurement and neighbor node measurement.

One of the primary research methods of the consistency algorithm is the graph theory, which takes a graph consisting of given points and connecting lines as the research object. It is usually used to describe the relationship between things [19]. In this study, the point represents the tracking target and sensor network; the edge represents the point communication between the two components. Correspondingly, the graph represents various communication relationships in the designed target tracking system. There is a target tracking network composed of n randomly separated sensor nodes in region G. An undirected connection graph represents the communication topology between sensor nodes in the network, denoted as $M = (A, B, C)$. Among them, $A = \{a_1, a_2, L, a_n\}$ indicates the set of n nodes in the network; n refers to a non-zero integer; $B \subseteq A \times A$ expresses the opposite side set of node order in the graph, denoting the interaction link of working data between nodes. Besides, $b_{ij} = (a_i, a_j)$ indicates the node communication, where a_i is the starting node and a_j stands for the terminal node. If $b_{ij} \in B$, the node a_j is the neighbor node of the node a_i. For the node a_i, there is a neighbor node set N_i. $C = [c_{ij}]$ indicates the weighted adjacency matrix of the graph M, where $c_{ij} = 0$. Additionally, if $a_{ji} \in B$, the weight $c_{ij} > 0$, otherwise $c_{ij} = 0$. If all nodes meet $\sum_{j=1}^{n} c_{ij} = \sum_{j=1}^{n} c_{ji}$, the graph M is balanced, and the in-degree of each node is equal to the out-degree.

$d_{ij} = \sum_{j=1}^{n} c_{i\rho}(i = 1, 2, \ldots\ldots, n)$ is defined as the in-degree of the node a_{i}, demonstrating the number of edges pointing to the node a_{i} in the graph M. $D = [d_{ij}]$ refers to the in-degree matrix of the graph M. Definition is made on the harmonic matrix $L_{\mathrm{n}} = D - C = [l_{ij}]$ of the graph M, D represents the in-degree matrix and C means the weighted adjacency matrix. Eq. (1) illustrates the relationship between the parameters.

$$l_{ij} = \begin{cases} \sum_{j=1, j\neq i}^{n} c_{ij}, & i = j \\ -c_{ij}, & i \neq j \end{cases} \tag{1}$$

Eq. (1) indicates that if $c_{ij} \notin B$, there is $l_{ij} = -c_{ij} = 0$. Additionally, L_n accords to Eq. (2).

$$l_{ij} < 0; \quad \sum_{j=1}^{n} l_{ij} = 0, \quad i = 1, 2, \ldots, n \tag{2}$$

Eq. (2) reveals that the harmonic matrix L_n contains "zero eigenvalues". In graph M, if any node a_i and node a_j meet $c_{ji} = c_{ij}$, the graph is undirected M and vice versa. An undirected graph can be considered a specially directed graph whose weighted adjacency matrix is symmetric, so an undirected graph is always balanced [20]. The consistency algorithm updates the state variables of nodes in the network through the communication between nodes to ensure that the state variables of each node converge to the same stable value. For a multi-agent system with n nodes, the state variable of the i_{th} agent is set as y_i. Eq. (3) manifests the expression of the average consistency algorithm in continuous time.

$$y_i(t) = -\sum_{j=1}^{n} c_{ij}\left(y_i(t) - y_j(t)\right), \quad i = 1, 2, \ldots, n \quad (3)$$

In Eq. (3), c_{ij} refers to the $(i, j)_{\text{th}}$ term of the weighted adjacency matrix C of graph M. This is the standard form of average consistency algorithm. Each node first produces its initial state variable in the system, recorded as $y_i(0)$. Then, each node communicates with each other according to the given relationship and uses the consistency algorithm to calculate according to the information sent by the neighbor nodes. When the system reaches a steady state, trends of all the y_{i} are consistent, as illustrated by Eq. (4).

$$\lim_{t\to\infty} y_i(t) = 0 \quad (4)$$

In the standard form, the information state variables of all nodes will reach the initial average of all $y_i(t)$, as signified by Eq. (5).

$$\lim_{t\to\infty} y_i(t) = \frac{1}{n}\sum_{i=1}^{n} y_i(0) \quad (5)$$

In practical application, the steady-state value generally does not accord with Eq. (5). In general, as long as Eq. (4) is satisfied in a steady state, it is considered that the state variable y_i converges to the uniform state. According to the properties of the matrix, the multi-agent system corresponding to the connected graph can converge to uniform.

C. NODE MOVING COMPUTING ALGORITHM

IoT is a network refers to a network that establishes communication, connection, and information sharing between things and the Internet through standard protocols to realize positioning, monitoring, and management. The core elements of the IoT are terminal perception, network communication, and application services [21]. Semantic technology is a specific method to connect a computer's representation with the real world. It is critical to solving the integration and cooperation problems of heterogeneous systems. By integrating semantic web and big data technology, the semantic description of IoT information is realized to implement semantic interoperability and collaboration between heterogeneous systems of IoT and the intellectualization of IoT applications [22]. With moving computing technology, computers and other intelligent terminal devices can realize data transmission and resource sharing in a wireless environment and transfer information to a distributed computing environment under a remote server [23]. Semantics annotates the data at the node into a computing language that computers and intelligent devices can understand to assist in the movement of real-time state data of nodes during target tracking.

The node moving model is constructed. In the model, the sensing radius of each node in the network is set to be the same, and the target is detected and located within the node sensing radius. In the model, the nodes outside the target sensing range are selected to move to the target. FIGURE 2 displays the node moving model of the wireless sensor network in the 2D planar.

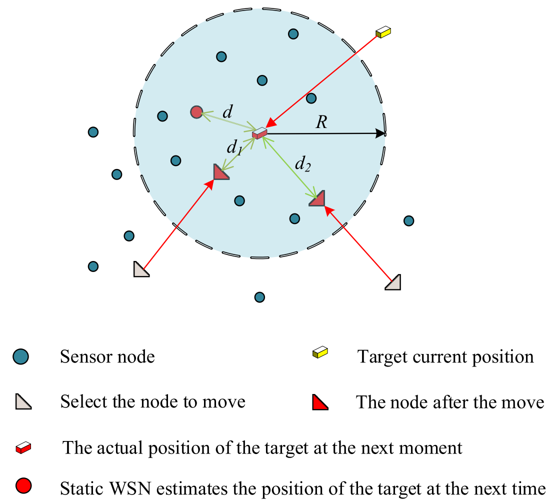

FIGURE 2. Node moving model of wireless sensor networks in 2D space.

In FIGURE 2, R represents the radius of the sensing node, and d_i refers to the distance between the moving node and the target at the next time. In the 2D plane, assuming that a node perceives the target in the static state, its coordinates are $(x_{\text{i}}, y_{\text{i}})$, $i = 1, 2, \ldots, n$, Eq. (6) describes the target estimation and positioning.

$$S_{i2D} = \left(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n}\right) \quad (6)$$

Eq. (7) determines the positioning error P_{r2D} of the target under the static network.

$$P_{r2D} = \sqrt{\left(\frac{\sum_{i=1}^{n} x_i}{n} - x\right)^2 + \left(\frac{\sum_{i=1}^{n} y_i}{n} - y\right)^2} \quad (7)$$

In Eq. (7), r refers to the perceived range radius. It is assumed that n_m selected moving nodes in the network are within the range after moving. The coordinates of them after moving are $(x_{n+i}, y_{n+i}, z_{n+i})$, $i = 1, 2, \ldots, n_m$, Eq. (8) illustrates the new definition error P'_{r2D} of the network to the target after moving.

$$P'_{r2D} = 1/2(\left(\frac{\sum_{i=1}^{n} x_i}{n} - x\right)^2 + \left(\frac{\sum_{i=1}^{n} y_i}{n} - y\right)^2) \quad (8)$$

Concurrently, if the node movement algorithm can meet the following conditions, the positioning error of the target will be dramatically reduced; meanwhile, the positioning accuracy will be significantly improved, which can be expressed as $P'_{r2D} < P_{r2D}$.

$$\begin{aligned} P_{i2D} &= \sqrt{(x_{n+i} - x)^2 + (y_{n+i} - y)^2} \\ &< P_{r2D}, \quad i = 1, 2, \ldots, n_m \end{aligned} \quad (9)$$

In Eq. (9), P_{i2D} represents the new positioning error of a location network to the target after moving. Compared with P'_{r2D}, if the moving length is less than the moving length of

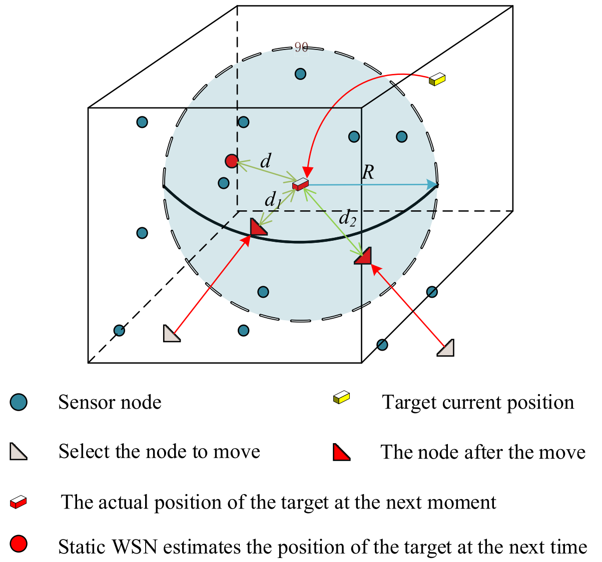

FIGURE 3. Node moving model of wireless sensor networks in a 3D space.

the maximum error within the allowed range, the tracking accuracy is acceptable.

Considering the actual situations, this paper also explores the problem of target tracking in the 3D space. FIGURE 3 displays the moving model of wireless sensor network nodes in a 3D space.

The meaning of each letter in FIGURE 3 is consistent with that of the 2D model. In the 3D space of real life, the small yellow box is the target's current position, and the small red box is the target's actual position the next time. Assuming that the position coordinates are (x, y, z), it is necessary to locate it at first. The sphere area can perceive the position of the target at the next time, and its internal nodes can perceive the target and locate the target. Now several nodes need to be selected from outside the sphere area to move towards the next time position of the target, and the red triangle is the node after moving. Assuming that a node perceives the target in the static state and its coordinates are (x_i, y_i, z_i), $i = 1, 2, \ldots, n$, Eq. (10) manifests the calculation of the target estimation and positioning.

$$S_{i3D} = \left(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n}, \frac{\sum_{i=1}^{n} z_i}{n} \right) \tag{10}$$

Eq. (11) can further illustrate calculation of the positioning error of the target in the static network, (11), as shown at the bottom of the next page.

In Eq. (11), r refers to the sensing range radius. Assume that the selected moving nodes in the network are all in the sphere area after moving, and the position coordinates after moving are $(x_{n+i}, y_{n+i}, z_{n+i})$, $i = 1, 2, \ldots, n_m$. Then, the network has a new definition error for the target after the node's moving, as shown in Eq. (12), as shown at the bottom of the next page.

Moreover, the node movement algorithm meeting the following conditions can significantly reduce the target positioning error and improve the positioning accuracy, expressed as $P'_{r3D} < P_{r3D}$. Eq. (13) signifies the process, where $i = 1, 2, \ldots, n_m$.

$$P_{i3D} = \sqrt{(x_{n+i} - x)^2 + (y_{n+i} - y)^2 + (z_{n+i} - z)^2} < P_{r3D} \tag{13}$$

In Eq. (13), P_{i3D} denotes the new positioning error of a location network to the target after moving. If the moving length is less than the moving length of the maximum error within the allowed range, the tracking accuracy is acceptable compared with P'_{r3D}.

This experiment sets several nodes to move to the next position to obtain satisfying tracking and positioning effect. Two nodes in the randomly distributed node network are selected to move to the next position in the 2D space. Besides, three nodes are chosen to approach the target in the 3D space to get precise positioning results. In addition, this experiment evaluates the performance of the node mobility algorithm through the target location estimated by the centroid localization algorithm, the perceived distance of each node in the network, and the area of the deployment area.

D. TARGET TRAJECTORY PREDICTION MODEL BASED ON AVERAGE CONSISTENCY AND MOVING COMPUTING ALGORITHM

This experiment tests the target trajectory prediction effects of the above two algorithms. The specific experiment steps are as follows. The target moving trajectory is divided into linear motions in several small clusters, i.e., linear segments [24]. Suppose that the current calculated target position is $L(x_t, y_t)$, the target position at the previous time is $L_{t-1}(x_{t-1}, y_{t-1})$, and the sampling interval Δt. Eqs. (13) and (14) describe the velocity and acceleration of the target.

$$v_t = \sqrt{(y_t - y_{t-1})^2 + (x_t - x_{t-1})^2} / \Delta t \tag{14}$$

$$a_t = (v_t - v_{t-1}) / \Delta t \tag{15}$$

Eq. (16) demonstrates the moving direction θ of the target.

$$\theta = arccos \frac{x_t - x_{t-1}}{\sqrt{(y_t - y_{t-1})^2 + (x_t - x_{t-1})^2}} \tag{16}$$

Eq. (17) signifies the position $L_{t+1}(x_{t+1}, y_{t+1})$ of the target at the next moment.

$$\begin{cases} x_{t+1} = x_t + v_t \Delta t \, cos\theta + \dfrac{1}{2} a_t \, cos\theta \, \Delta t^2 \\ y_{t+1} = y_t + v_t \Delta t \, cos\theta + \dfrac{1}{2} a_t \, sin\theta \, \Delta t^2 \end{cases} \tag{17}$$

The moving speed and direction of the target determine the residence time and moving distance of the target in the current cluster. The linear function of the target $y = ax + b$ can be constructed using the historical position information of the target to calculate the sum of parameters a and b. The cluster head node of the current target tracking cluster is set as the coordinate' origin. $x^2 + y^2 = r^2$ is considered as the curve function of the perception boundary of the current cluster head, where r refers to the radius of the current monitoring area, and x and y are the unknown coordinate information of the target, respectively. The motion trajectory and curve functions are connected to obtain Eq. (18).

$$\begin{cases} x^2 + y^2 = r^2 \\ y = ax + b \end{cases} \tag{18}$$

Eqs. (19) and (20) calculate the final position coordinates L of the tracking target leaving the current cluster.

$$L = \left(\frac{-ab \pm c}{1 + a^2}, \frac{-a^2 b \pm ac}{1 + a^2} + b \right) \tag{19}$$

$$c = \sqrt{a^2 r^2 + r^2 - b^2} \tag{20}$$

In Eq. (20), c stands for a constant parameter.

The final position of the target is determined by Eq. (19) and the target's moving direction. The target moving direction and the final position after the target leaves the current cluster are determined by judging the difference of axis coordinates. The target's initial position entering the current cluster is set as $A_0(x_0, y_0)$. Eq. (20) demonstrates the value of the moving distance d of the target in the current cluster.

$$d = \sqrt{\left(\frac{-ab \pm c}{1 + a^2} - x_0 \right)^2 + \left(\frac{-a^2 b \pm ac}{1 + a^2} + b - y_0 \right)^2} \tag{21}$$

Eq. (20) signifies the residence time t_{stqv} of the moving target in the current cluster.

$$t_{stqv} = \begin{cases} \dfrac{d}{v_t} & a_t = 0 \\ \dfrac{\sqrt{v_t^2 - 2a_t d} - v_t}{a_t} & a_t \neq 0 \end{cases} \tag{22}$$

After forecasting the target position, the cluster head node judges the number of nodes in the cluster where the target is about to arrive in the tracking area. It also calculates the predicted position of the target and the current position of the cluster of the target. Meanwhile, it controls the nodes in the target area to turn on the tracking state. The nodes receiving the message first confirm whether it is in the target prediction area. If yes, it will turn on the working state; otherwise, it will discard the information. The nodes in the tracking state continuously perceive the target's information, while the nodes out of the tracking maintain the original periodic detection mode [25]. When the target moves from a tracking cluster to a new tracking cluster, it needs to constantly update the motion state and position of the target. Before time t_{stqv} runs out, the head node of the current tracking cluster wakes up the adjacent cluster head nodes according to the predicted final position of the current cluster to track the hands of clusters.

TABLE 1. Basic parameter setting of simulation.

Parameter setting	Value
Monitoring area	200*200/m
Node density	$1.5*10^{-3}$
Attenuation index a	2
Check threshold e_0	1
Communication radius R	40m
Strength s of the target signal source	100
Noise distribution	$n_i \sim N(0, \sigma^2)$

E. SIMULATION SETTINGS OF THE ALGORITHM

The target trajectory prediction is simulated and tested by MATLAB to verify the feasibility and accuracy of the algorithm. Besides, this experiment compares the tracking target loss rate and tracking error of this algorithm and the Extended Kalman Filter (EKF) for performance verification [26]. TABLE 1 lists the basic settings of the simulation.

IV. RESULTS AND DISCUSSIONS

A. SIMULATION RESULTS OF THE NODE MOVEMENT ALGORITHM IN 2D AND 3D SPACES

In this experiment, n_m equals 0 and 2 in 2D and 3D space, respectively; n_s equals 20 and 50 in 2D space, and it equals 60 and 100 in 3D space. FIGURE 4 reveals the experimental simulation results as the sensing nodes' radius changes.

FIGURE 4 shows the positioning errors of the target in the static network when n_m equals 0 and n_m equals 2. The target tracking model via the node movement algorithm effectively improves the positioning accuracy of the network for moving targets. The positioning error after nodes move in 2D space is $20\%-30\%R$ less than that in the static state; the positioning error after nodes move in 3D space is about $40\%-50\%R$ less than that in a static state. Concurrently, the four curves in the 2D space decrease with the growth of the sensing radius, suggesting that the increase of the sensing radius can improve the positioning accuracy of the network in the 2D space. In the 3D space, with the increase of the perceived radius,

$$P_{r3D} = \sqrt{\left(\frac{\sum_{i=1}^{n} x_i}{n} - x \right)^2 + \left(\frac{\sum_{i=1}^{n} y_i}{n} - y \right)^2 + \left(\frac{\sum_{i=1}^{n} z_i}{n} - z \right)^2} \tag{11}$$

$$P'_{r3D} = \sqrt{\left(\frac{\sum_{i=1}^{n+n_m} x_i}{n + n_m} - x \right)^2 + \left(\frac{\sum_{i=1}^{n+n_m} y_i}{n + n_m} - y \right)^2 + \left(\frac{\sum_{i=1}^{n+n_m} z_i}{n + n_m} - z \right)^2} \tag{12}$$

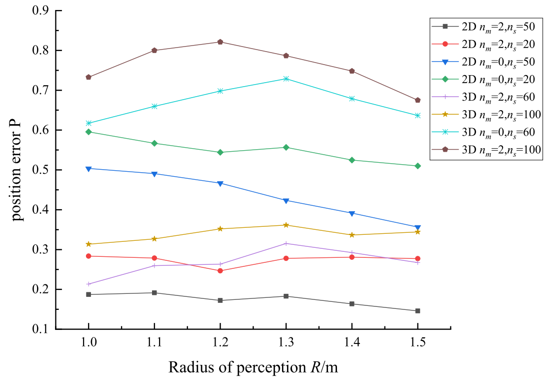

FIGURE 4. **Variation trend of positioning error under the changes of the radius of sensing nodes.**

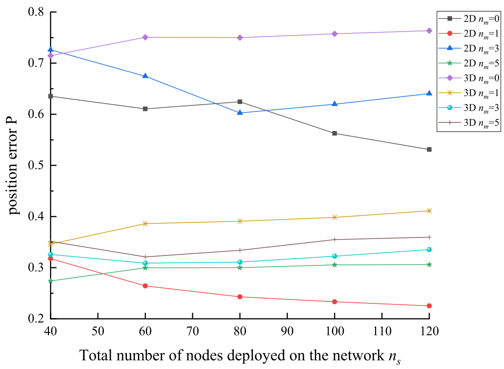

FIGURE 5. **Positioning error under different network deployment areas.**

the positioning error increases at first and then decreases, and the four 3D curves have maximum values at 1.3m. This result indicates that the positioning performance of the network to the moving target is the worst when the perceived radius is 1.3m.

Suppose that in 2D and 3D space, the average sensing radius R is 2m, and n_m equals 0, 1, 3, and 5 in 2D and 3D space, respectively. FIGURE 5 presents the experimental simulation results under an increasing number of node deployments n_s.

FIGURE 5 reveals that when the node sensing radius R is 2m, the positioning error of the network to the target changes with values of n_s. In 2D space, with the increase of the total number of nodes deployed, the network's target positioning error gradually declines. Except when n_m equals 3, the positioning error increases when n_s equals 80, which may result from the information acquisition of nodes. Besides, the positioning accuracy of the network increases with more mobile nodes, and the positioning accuracy is higher than that of the static network. However, the target positioning error of the network in the 3D space does not change sharply with the rise in n_s. The reason may be the impact of n_m on the positioning error. FIGURE 5 also indicates that when n_m equals 3 in the 3D space, the network achieves the best performance and the highest positioning accuracy. Meanwhile, when the number of moving nodes is 0, the positioning error in the 3D space is much higher than the other three cases in the 2D space. This phenomenon indicates that in the 3D space, the number of moving nodes can significantly reduce the positioning error of network targets.

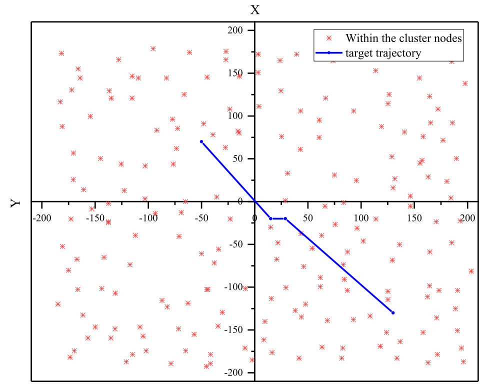

FIGURE 6. **Node distribution and target trajectory.**

B. SIMULATION RESULTS OF TARGET TRAJECTORY PREDICTION

This test verifies the feasibility of the algorithm. The curve moving trajectory in the target is divided into several linear movements in clusters, and the plane coordinate system is set to $x = [-200, 200]$, $y = [-200, 200]$. The initial target position is set as $(-50, 70)$, and the speed variation range of the target is 1-20m/s. FIGURE 6 provides the node distribution and target trajectory.

According to FIGURE 6, the algorithm can predict the direction of the target in different directions and locate it in time. The actual trajectory of the target is an irregular curve; the whole monitoring area is divided into several clusters. The irregular curve is wirelessly divided into small segments according to calculus theory. When the target moves at a random speed, the algorithm can also predict the motion trajectory of the target.

This algorithm is compared with the tracking trajectory of the commonly used target tracking algorithm based on the Kalman Filter. When the target motion direction changes slightly, both algorithms can track the actual motion trajectory of the target. However, when the motion direction changes by more than or equal to 90°, the algorithm is more effective for piecewise linear fitting prediction.

Target loss rate refers to the proportion of invalid positioning times to the total number of positioning times in target tracking [27]. The target loss rate is calculated according to the research algorithm and the target tracking algorithm based on the Kalman Filter. FIGURE 7 illustrates the calculation results.

FIGURE 7 proves that the target loss rate of the two algorithms increases with the increase of target speed. The

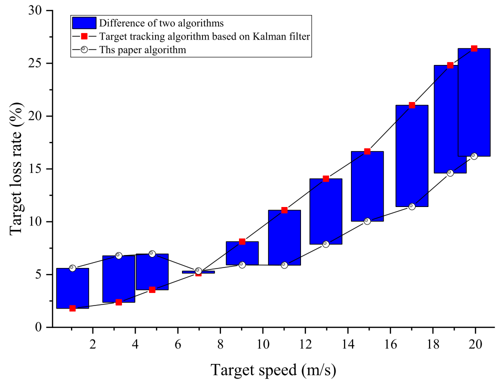

FIGURE 7. Comparison of target loss rates.

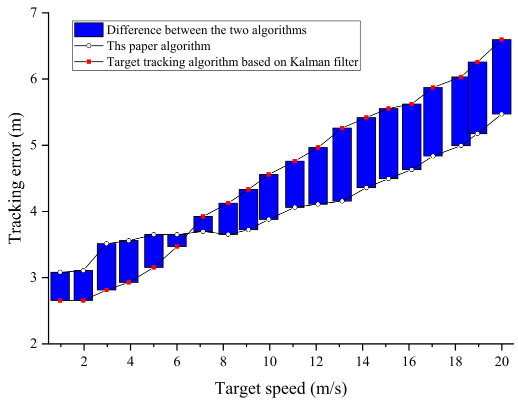

FIGURE 8. Comparison of tracking errors.

two algorithms have stability and a low loss rate when the target moves slowly because the algorithms adopt an effective prediction mechanism to adapt to the change of speed. Hence, the target loss rate increases slightly, and the target loss rate is less than 30% in the speed change range. Compared with the two algorithms, when the target speed is higher than 7m/s, the target loss rate of the research algorithm is lower, and the difference between the two algorithms increases gradually. The reason is that this algorithm combines static clustering and a reasonable target recovery mechanism. It can quickly find the lost target and reduce the target loss rate [28].

The tracking error can reflect the tracking accuracy of the algorithm. The tracking error is calculated according to the research algorithm and the target tracking algorithm via Kalman Filter. FIGURE 8 shows the calculation results.

In FIGURE 8, the tracking errors of the two algorithms are generally low when the target motion speed is low. They can both meet the target tracking requirements below 4m. When the target speed reaches more than 7m/s, the tracking error of the research algorithm is lower than that of the target tracking algorithm via Kalman Filter because the prediction mechanism adopted by the algorithm can motivate the tracking target in advance and reduce the error.

V. CONCLUSION

The consistency algorithm applied to the system is characterized by security, high availability, and independent timing to ensure consistency. The node movement algorithm can reduce the impact of network delay for real-time calculation. Therefore, the average consistency and semantic IoT node movement algorithm establish the target trajectory prediction model. The simulation results of the node movement algorithm show that within a specific range, the positioning error decreases with the increase of sensing radius R, the number of nodes moving n_m, and the total number of nodes n_s deployed in 2D space. However, there is no obvious law in 3D space. The experimental results of the prediction model show that the target loss rate and tracking error distance of the two algorithms grow with the increase of the tracking target speed. When the target tracking speed is greater than 7m/s, the research algorithm's target loss rate and tracking errors are about 0$\sim$10% and 1.5%$\sim$2% lower than the target tracking algorithm via Kalman Filter, respectively. However, when the tracking speed of the research algorithm is less than 7m/s, the tracking accuracy is lower than the Kalman Filter algorithm.

Previous studies have insufficient accuracy and robustness in target tracking. Combining the two algorithms makes a more robust target tracking method possible. In other words, the scheme reported here reduces the data error of multiple nodes, continuously updates the node cluster state, and maintains the original periodic detection mode of untracked nodes. Besides, the mobility of nodes is more flexible than in a static state, inspiring future distributed point target tracking. Due to the limited energy, this experiment only compares and evaluates the proposed algorithm with a general algorithm. The follow-up research will comprehensively evaluate multiple algorithms according to the specific situation to provide a reference for improving the algorithm.

REFERENCES

[1] A. Liu and S. Zhao, "High-performance target tracking scheme with low prediction precision requirement in WSNs," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 29, no. 4, pp. 270–289, Nov. 2018.
[2] S. Guo, X. Zhang, Y. Du, Y. Zheng, and Z. Cao, "Path planning of coastal ships based on optimized DQN reward function," *J. Mar. Sci. Eng.*, vol. 9, no. 2, pp. 210–223, Feb. 2021.
[3] X. Wang, L. Sun, T. Wang, and Y. Shi, "Temperature-compensated pH microelectrode array sensors based on copper-oxide/polyaniline modification," *IEEE Sensors J.*, vol. 20, no. 24, pp. 14598–14606, Dec. 2020.
[4] D. Deepakraj and K. Raja, "Markov-chain based optimization algorithm for efficient routing in wireless sensor networks," *Int. J. Inf. Technol.*, vol. 13, no. 3, pp. 897–904, Mar. 2021.
[5] V. Suma, "Wearable IoT based distributed framework for ubiquitous computing," *J. Ubiquitous Comput. Commun. Technol.*, vol. 3, no. 1, pp. 23–32, May 2021.
[6] S. K. Somov, "Creation of information-technological reserve in distributed data processing systems," *Autom. Remote Control*, vol. 80, no. 4, pp. 781–790, Apr. 2019.
[7] P. Ren, S. Wang, M. He, and W. Cao, "Novel strategy based on improved Kalman filter algorithm for state of health evaluation of hybrid electric vehicles Li-ion batteries during short- and longer term operating conditions," *J. Power Electron.*, vol. 21, no. 8, pp. 1190–1199, May 2021.
[8] Z. Yu, Di Shi, Z. Wang, Q. Zhang, J. Huang, and S. Pan, "Distributed estimation of oscillations in power systems: An extended Kalman filtering approach," *CSEE J. Power Energy Syst.*, vol. 5, no. 2, pp. 181–189, May 2019.

[9] S. Wang and W. Ren, "On the convergence conditions of distributed dynamic state estimation using sensor networks: A unified framework," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1300–1316, Jul. 2018.
[10] M. Valipour and L. A. Ricardez-Sandoval, "Assessing the impact of EKF as the arrival cost in the moving horizon estimation under nonlinear model predictive control," *Ind. Eng. Chem. Res.*, vol. 60, no. 7, pp. 2994–3012, Feb. 2021.
[11] B. Zhang, J. Zhu, X. Lu, Y. Gu, J. Li, X. Liu, and M. Zhang, "An infrared dim target detection algorithm based on density peak search and region consistency," *Opt. Quantum Electron.*, vol. 53, no. 7, pp. 1–18, Jul. 2021.
[12] A. Zaidi, M. Kazim, R. Weng, D. Wang, and X. Zhang, "Distributed observer-based leader following consensus tracking protocol for a swarm of drones," *J. Intell. Robotic Syst.*, vol. 102, no. 3, pp. 1–22, Jun. 2021.
[13] C. Thomson, I. Wadhaj, Z. Tan, and A. Al-Dubai, "A mobility aware duty cycling and preambling solution for wireless sensor network with mobile sink node," *Wireless Netw.*, vol. 27, no. 5, pp. 3423–3439, Mar. 2021.
[14] G. Venâncio, R. C. Turchetti, E. T. Camargo, and E. P. Duarte, "VNF-consensus: A virtual network function for maintaining a consistent distributed software-defined network control plane," *Int. J. Netw. Manage.*, vol. 31, no. 3, p. e2124, May 2021.
[15] J. Hu and K. Liu, "Raft consensus mechanism and the applications," *J. Phys., Conf. Ser.*, vol. 1544, no. 1, May 2020, Art. no. 012079.
[16] Y. Li, L. Qiao, and Z. Lv, "An optimized Byzantine fault tolerance algorithm for consortium blockchain," *Peer-Peer Netw. Appl.*, vol. 14, no. 5, pp. 2826–2839, Mar. 2021.
[17] W. Jiang, X. Liu, X. Liu, Y. Wang, S. Lv, and F. Ye, "A new behavior-assisted semantic recognition method for smart home," *China Commun.*, vol. 17, no. 6, pp. 26–36, Jun. 2020.
[18] A. Al-Shdifat, C. Emmanouilidis, M. Khan, and A. Starr, "Ontology–based context resolution in Internet of Things enabled diagnostics," *IFAC-PapersOnLine*, vol. 53, no. 3, pp. 251–256, Dec. 2020.
[19] L. Zhang, M. Jin, and S. J. Yoo, "A clustering detector with graph theory for blind detection of spatial modulation systems," *Wireless Netw.*, vol. 27, no. 4, pp. 1–9, Apr. 2021.
[20] M. Guay, "Distributed optimization by Newton consensus over undirected graphs," *IFAC-PapersOnLine*, vol. 54, no. 9, pp. 636–641, Jul. 2021.
[21] A. I. Khan and A. Al-Badi, "Open source machine learning frameworks for industrial Internet of Things," *Proc. Comput. Sci.*, vol. 170, no. 10, pp. 571–577, 2020.
[22] L. Greco, P. Ritrovato, and M. Vento, "On the use of semantic technologies for video analytics," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 567–587, Jan. 2021.
[23] H. C. Ke, H. Wang, H. W. Zhao, and W. J. Sun, "Deep reinforcement learning-based computation offloading and resource allocation in security-aware mobile edge computing," *Wireless Netw.*, vol. 27, no. 5, pp. 3357–3373, May 2021.
[24] X. Zhang, H. Cui, Q. Luo, and H. Zhang, "Nonlinear distributed model predictive control for multiple missiles against maneuvering target with a trajectory predictor," *J. Shanghai Jiaotong Univ., Sci.*, vol. 25, no. 6, pp. 1–11, Oct. 2020.
[25] S. Chauhan, M. Singh, and A. K. Aggarwal, "Cluster head selection in heterogeneous wireless sensor network using a new evolutionary algorithm," *Wireless Pers. Commun.*, vol. 119, pp. 585–616, Feb. 2021.
[26] M. V. Kulikova and G. Y. Kulikov, "MATLAB-based general approach for square-root extended-unscented and fifth-degree cubature Kalman filtering methods," *Eur. J. Control*, vol. 59, pp. 1–12, May 2021.
[27] Z. Ji, Q. Wang, B. Cui, Y. Pang, X. Cao, and X. Li, "A semi-supervised zero-shot image classification method based on soft-target," *Neural Netw.*, vol. 143, pp. 88–96, Nov. 2021.
[28] H. Qabouche, A. Sahel, and A. Badri, "Hybrid energy efficient static routing protocol for homogeneous and heterogeneous large scale WSN," *Wireless Netw.*, vol. 27, no. 1, pp. 575–587, Jan. 2021.

YUN WANG was born in Taiyuan, Shanxi, China, in 1982. He received the master's degree from Shanxi University, China. He is currently an Associate Professor with the Department of Computer Engineering, Shanxi Polytechnic College. His major research interests include artificial intelligence, electronic image processing, and the Internet of Things Technology.

HAN LIU is currently pursuing the degree with the College of Quality and Technical Supervision, Hebei University. His research interests include sensor management and target tracking in wireless sensor networks.

JUN ZHOU was born in Gaoan, Jiangxi, China, in 1981. He received the Ph.D. degree from Logistic Engineering University, China. He currently works with the Chongqing Business Vocational College. His research interests include artificial intelligence and image processing.

YI WAN was born in Zigong, Sichuan, China, in 1980. He received the master's degree from the University of Electronic Science and Technology of China, China. He is currently with the School of Information Engineering, Southwest University of Science and Technology. His research interests include automatic control theory, image processing and pattern recognition, and intelligent control.