Target Tracking Based on DDPG in Wireless Sensor Network

Yinhua Liao(✉) and Qiang Liu

University of Electronic Science and Technology of China,
Chengdu 611731, Sichuan, China
lynch_jach@126.com, liuqiang@uestc.edu.cn

Abstract. For target tracking in mission critical sensors and sensor networks (MC-SSN), the contribution of the measured value of each sensor node to the data fusion center is different, so better weighted node fusion and scheduling node participation in tracking can obtain better tracking performance. In this paper, to address this problem and fully utilize the network transmission capability, we proposed a collaborative perception and intelligent scheduling to jointly optimize system responding latency and tracking accuracy while guaranteeing low energy consumption. Based on the unreliable historical tracking data, we formulate the joint optimization problem as the infinite horizon Markov Decision Process (MDP), we propose an intelligent collaboration scheme based on the deep deterministic policy gradient (DDPG) approach to perform the optimal tracking with low energy consumption and high tracking accuracy.

Keywords: Wireless sensor network · Target tracking · Collaborative perception · Deep deterministic policy gradient

1 Introduction

With the rapid development of technology and the development of mature chip designs and embedded systems, sensors are gradually developing towards miniaturization and integration with sensing computing and network communication functions. Wireless sensor networks (WSNs) [1] has been developed for more than 30 years since it was proposed, it consists of a large number of tiny nodes with sensing, computing, and wireless communication capabilities. Its purpose is to monitor the environment rather than communicate. Target detection and tracking [2] is a classic application of MC-SSN [3]. As [4] summarizes, related research can be divided into five categories: tree-based [5], based on clustered [6], prediction-based [7], mobile messaging-based [8] and hybrid method [9]. However, most sensors set the sensor to stable, which limits Tracking algorithm performance.

In the research of target tracking based on wireless sensor networks, because the cost of sensor nodes may not be fully covered, it is necessary to rely on the movement of nodes to track in real time. The research mainly has the following two challenges:

Y. Chen et al. (Eds.): BICT 2020, LNICST 329, pp. 253–267, 2020.
https://doi.org/10.1007/978-3-030-57115-3_22

1. *Node coordination problem*. Due to the characteristics of the wireless sensor network itself, it is difficult to locate the target through a single node Effective tracking, so multiple nodes need to coordinate tracking in the target tracking process. In the process of collaborative tracking, it is mainly constrained by the conditions such as energy consumption and tracking accuracy in the network. It is necessary to design an optimal collaborative tracking algorithm to select cluster nodes to track the target, thereby improving accuracy and prolonging the network life cycle.
2. *Uncertainty in the tracking process* The uncertainty of the target tracking process is mainly reflected in the uncertainty of the number of targets and the detection data. Certainty. In the study of target tracking, when monitoring a certain area, the timing and number of target intrusions cannot be determined in advance, and information matching and accuracy improvement cannot be effectively performed. Therefore, how to track more reliably should be considered when designing the algorithm. The uncertainty of the detected data is mainly due to the real-time update of the environment in the wireless sensor network and the interference of the clutter during the data transmission. In this regard, the accuracy of data collection and filtering should be considered.

Target tracking is one of its important applications among the many applications of wireless sensor networks, the potential of target tracking in WSN is also increasing, such as indoor location, target detection, driverless vehicles and intelligent monitoring systems, and has shown good prospects in the battlefield environment. Most excellent works had investigated in mobile target tracking sensor network with single or multiple targets. Considering the issue of the target tracking accuracy in indoor wireless networks, [10] proposed a grid-based indoor collaborative location tracking algorithm (CLTA) to locate indoor complex and crowded environments. The algorithm is divided into offline phase and online phase. In the offline stage, a collaborative positioning fingerprint database is established based on reliable nodes. In the online phase, the area overlap mechanism and prediction mechanism are used to reduce the location area in a multi-network environment. [11] designed a tracking solution called "t-tracking", which aims to achieve two main goals: WSN's high QoT and high energy efficiency. The author proposes a completely distributed tracking algorithm. When the target moves on the face, the facial nodes close to its estimated motion will calculate the sequence of the target's motion and predict when the target will move to another face. Multi-rate distributed fusion estimation for maneuvering target tracking WSN. [12] proposed a multi-rate fusion strategy and a hierarchical two-stage fusion structure. The algorithm mainly focuses on the consideration of energy efficiency and tracking accuracy. In the first stage, a locally modified strong tracking filter estimator is designed to obtain local estimates in each cluster head in WSN; in the second stage, a multi-rate fusion estimator is designed to generate a fusion estimate with higher estimation accuracy. The author of [13] proposed an energy-saving strategy that uses mobile sensor networks to track moving targets in an environment with obstacles. The algorithm uses a sufficiently small cell network to reflect the

sensor's energy consumption through appropriate weighting. And use the shortest path algorithm to search the best position of the sensor at different times. In [14], author proposed a novel distributed unbiased finite impulse response (UFIR) filter, which has a powerful modeling error in an uncertain noise environment, and it can provide good services in mobile sensor network for estimating the target state and position. [15] proposed a potential game-based non-myopia planning method for mobile sensor networks in target tracking environment. Select the order of sensing points on multiple future time steps to maximize the information about the target state.

However, the latest research on mobile MC-SSN focuses on increasing coverage, designing effective routing protocols or optimizing data collection. As for improving the precise detection and real-time tracking of targets, current research is very limited. The main reason for this scarcity is that due to the application of the group movement model in the network, the computational complexity has increased significantly, which makes the research more challenging, especially in the field of target tracking.

The main contributions summarized are as follows:

1. we give a feasible scheme of anchor nodes in intraregional and regional deployment. Our goal is to reduce the probility of tracking failure, and we adjust locations of moving anchor nodes after each tracking processing.
2. After achieving the task of deployment, in order to track in real-time, the trajectory prediction is proposed to improve tracking accuracy based on theoretic foundation of Kalman filter in discrete time.
3. Under the premise of tracking accuracy, the no-convex objective function of minimizing energy consumption with noticeable constraints, in which most are nonlinear, is formulated. Fortunately, AI algorithm is proposed in this paper to solve this dilemma, and we use Deep Deterministic Policy Gradient (DDPG) algorithm, which is proper to dispose problems in discrete time.
4. We evaluate the effectiveness of our scheme through theoretical analysis and simulations under the environment of TensorFlow and Python.

The rest of this paper is organized as follows. A brief review of Deep Deterministic Policy Gradient (DDPG) is given by Sect. 2. The system model and the introduction of target tracking are given by Sect. 3. Section 4 presents the target tracking process based on DDPG. Simulation results and analysis are presented in Sect. 5. Finally Sect. 6 concludes the paper.

2 Brief Review of DDPG

Deep Deterministic policy gradient is an algorithm framework that applies deep reinforcement learning algorithm to continuous action space [16]. It combines deep neural network and DPG (Deterministic Policy Gradient) algorithm, The actor-critic algorithm is taken as the basic structure of the algorithm.

Since the reinforcement learning using only this neural network algorithm, the learning process of the action value (Q value) may be unstable, because the

parameters of the value network are used to calculate the gradient of the strategy network while frequent gradient updates, therefore, The DDPG algorithm creates two neural networks for the strategy network and the value network, one for the online network and the other for the target network. [17,18]. Both actors and critics include two network models, online strategy and target strategy, in which actors are responsible for strategy networks and critics are responsible for value networks. Through the process of interaction between actors and the environment, the samples generated by the interaction are stored in the experience pool. In the next time step, the experience pool transfers small batches of sample data to the actors and critics for calculation. The architecture of networks is shown in Fig. 1 [19].

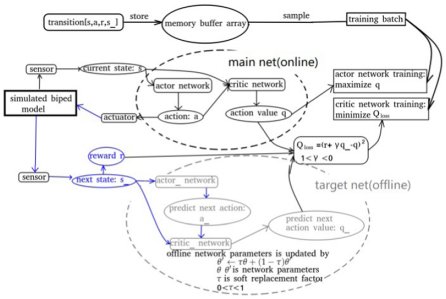

Fig. 1. The critic and actor networks.

The objective function of the DDPG algorithm is defined as the expectation of discounted cumulative reward, it's defined as follows:

$$J_\beta(\mu) = \mathbb{E}_\mu[r_1 + \gamma r_2 + \gamma^2 r_2 + ... + \gamma^n r_n] \tag{1}$$

In order to find the optimal deterministic behavior strategy μ^*, wait for the strategy in maximizing the objective function $J_\beta(\mu)$.

$$\mu^* = \underset{\mu}{argmax} J(\mu) \tag{2}$$

In [20], it is proved that the gradient of the objective function $J_\beta(\mu)$. With respect to the policy network parameter θ^μ is equivalent to the expected gradient of the action function $Q(s, a; \theta^Q)$ with respect to θ^μ, so the objective function is derived by following the chain derivation rule to obtain the update method of the actor network.

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s_t \sim \rho^\beta}[\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)}] \tag{3}$$

where $Q(s, a|\theta^Q)$ represents the action state Q that can be generated when the action is selected according to the deterministic strategy μ in the state s, and $\mathbb{E}_{s_t \sim \rho^\beta}$ represents the expectation of the Q value when the state s meets the

distribution ρ^{β}. Because of the deterministic strategy $a = \mu(s; \theta^{\mu})$, the update method is as follows:

$$\nabla_{\theta^{\mu}} J \approx \mathbb{E}_{s_t \sim \rho^{\beta}}[\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s=s_t}] \tag{4}$$

The gradient ascent algorithm is used to optimize the calculation of the objective function, and the goal of gradient ascent is to increase the expectation of the cumulative reward of the discount factor. Finally, the algorithm updates the parameters of the strategy network along the direction of action value $Q(s, a; \theta^Q)$.

To update the critic network through the DQN (Deep Q-Learning, DQN) method of updating the value network, the gradient of the value network is as follow:

$$\nabla_{\theta^Q} = \mathbb{E}_{s,a,r,s' \sim R}[(TargetQ - Q(s, a; \theta^Q)) \nabla_{\theta^Q} Q(s, a; \theta^Q)] \tag{5}$$

where $TargetQ = r + \nabla_{\theta^{Q'}} Q^{'}(s^{'}, \mu(s^{'}; \theta^{\mu^{'}}))$

The purpose of DDPG algorithm training is to maximize the objective function J_{β} while minimizing the loss of the value network Q.

With the development of reinforcement learning, more and more researchers apply it to mobile node control. The purpose of reinforcement learning is to learn optimal behavior through interaction with the environment. Compared with traditional machine learning, reinforcement learning has the following advantages: first, because it does not require a sample labeling process, it can more effectively solve the special situation in the environment; second, it can treat the entire system as a whole, so that Some of these modules are more robust; third, reinforcement learning makes it easier to learn a range of behaviors. These characteristics are all applicable to mobile node decision control. Therefore, we present the architecture of DDPG in mobile MC-SSN.

3 System Model and Performance Indicators

In this section, we will consider the system model, which includes the node's deployment model and data mode. We firstly consider the deployment of anchor nodes for satisfying real-time detection and tracking, the basic theory of Kalman filter, and different states among anchor nodes to improve tracking accuracy, to apply schedule scheme for sink nodes, and to save energy consumption, respectively. By this, we set the index set $\mathcal{S} = \{1, 2, ..., S\}$ and $\mathcal{B} = \{1, 2, ..., B\}$ about regions and boundary nodes of each region, respectively. The intra-regional anchor nodes of each region are indexed by $\mathcal{M} = \{1, 2, ..., M\}$. The scene is shown in Fig. 2, where the green line represents the target movement trajectory, the red node represents the internal mobile node, and the blue node represents the edge static node.

3.1 Deployment Model

In this subsection, we consider that the deployment of the boundary nodes and intra-regional nodes for each region. The former mainly designs relationship

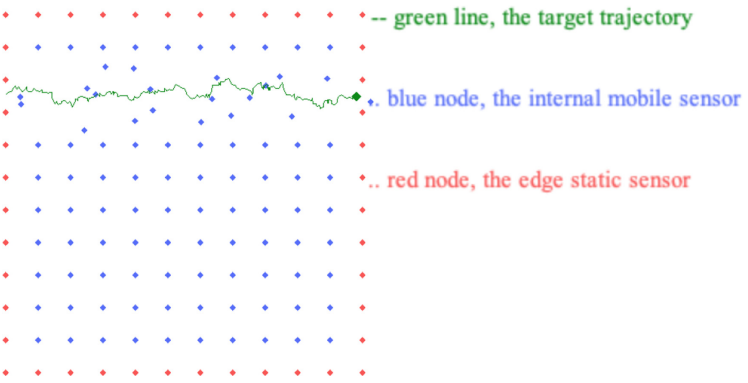

Fig. 2. The paradigm of target tracking in MC-SSN.

based on connectivity of the graph theory and the later mainly focuses on the coverage area in the process of nodes moving to track the target. In order to imp tracking accuracy and to inform boundary nodes of next region that the target is approaching to, we assume all the boundary nodes are static. When the target moves from region i to region j, the high connectivity is required. High connectivity is required.

3.2 Data Model

In order to reduce the system energy consumption and prolong the lifetime of network, the multimode of anchors is designed and are divided into idle, checking, working, and sleeping, which are represented by a vector $\overrightarrow{STA} = [s^{idle}, s^{sleep}, s^{check}, s^{work}]$. Assume that all the nodes are idle states and just receive data from others at the initial period, and nodes awake and sleep periodically. State is switched to checking as long as receiving data from the target, note that idle state answered for detecting in time is always kept by boundary nodes of any region. Anchors in checking state, which always keeps awake, transmit prediction and estimate information to sink node, and keeping current state or not is determined by the passback information. Switch into working state if scheduled to track the target. Otherwise, checking state, which is returned to idle unless receiving data from the target within a given time. After achieving a task of tracking in a period of time, anchors are in idle state, and continue to switch sleep state when not receive information about the target in a given time. Note that anchors stayed idle state for a certain time also switch sleep state and sink node will awake those anchors in sleep state after a certain time. All the cases of states switched possibly in the next time is shown in Fig. 3. These states are not the specific data form of each node in real time, but the descriptions defined for the overall operation of the real scene. It is a state description form defined by the node's detected target radius and its own remaining energy.

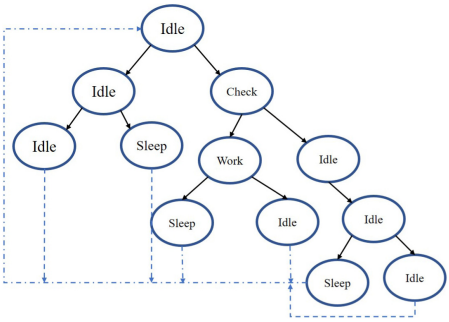

Fig. 3. All the cases of the node states.

3.3 Performance Indicators

Target Tracking Accuracy. Target Tracking accuracy mainly reflects the tracking effect of the target. In this paper, we adopt the Root Mean Square Error (RMSE) to reflect the target tracking accuracy [21]. The RMSE of the target position is as follows:

$$RMSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left((x_{k,m} - \widetilde{x}_{k,m})^2 + (y_{k,m} - \widetilde{y}_{k,m})^2\right)} \tag{6}$$

where M represents the number of simulations times, $(x_{k,m}, y_{k,m})$ and $(\widetilde{x}_{k,m}, \widetilde{y}_{k,m})$ are the true position and the estimated position of the target at time k in the m-th simulation respectively.

Tracking Response Time. It is the time required to track the information communication response of the node and the node's dispatch center and the time required for the data fusion of the dispatch center [22]. When the data transmission of the nodes is the same, this time is mainly determined by the data fusion calculation time.

Energy Consumption. The sensor nodes, we adopt the network energy consumption model in [23,24]. The energy consumption includes the energy consumption sending a b-bit packet and the energy consumption of receiving a b-bit packet, it's as follow:.

$$E_{send} = (e_s + e_d d^{\beta})b \tag{7}$$

$$E_{receive} = e_r b \tag{8}$$

where e_s is the transmit radio frequency energy consumption coefficient, e_d is the coefficient of the amplifier circuit, d is the euclidean distance between the transmitting and receiving nodes, β is the attenuation coefficient, e_r is the radio frequency consumption coefficient of the receiving node, and b is the number of data bits.

Reward. In this paper, we define a contribution degree to represent the reward of each mobile node. The energy consumption and moving time during node movement are used as the calculation of the contribution degree, and its expression is as follows:

$$Con_i = \omega E_{t,t+1}/E_s + (1-\omega)T_{t,t+1}/T_m \tag{9}$$

where ω is weight coefficients, $E_{t,t+1}$ is the energy consumed by the mobile node i from t to t + 1, E_s is the total energy of the node, $T_{t,t+1}$ is the time consumed by the node from t to t + 1, and Tm is the maximum time that the node can move at a uniform speed within the node detection radius. All divisions are normalized for node contribution.

4 The Target Tracking Process Based on DDPG

In this section, we will expand the specific scheduling algorithm of DDPG in the MC-SSN based on the foregoing. The following describes the specific data update and calculation during node movement. MC-SSN's target tracking algorithm needs to consider multiple performance indicators, such as tracking accuracy, tracking response time, and energy consumption. The ideal target tracking system has higher tracking accuracy, less tracking response time and energy loss. These performance parameters influence each other on the entire monitoring system. For example, if the tracking accuracy of the system needs to be improved, more measurement data of sensor nodes is required, which results in greater energy consumption for transmitting more information.

4.1 Prediction and Tracking

Based on above state, the scheme of deployment is designed for coverage of MSNs and the next work is performed by anchors. Assume that the location of target in time t is (x_t, y_t) and the target moves by the uniform motion with perturbation, and the horizontal and vertical velocities are given by $\widetilde{x}_t, \widetilde{y}_t$. The target motion is given by

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \omega_t \tag{10}$$

where $\mathbf{x}_t$ is a vector shown as $[x_t y_t, \widetilde{x}_t, \widetilde{y}_t]^T$, $\mathbf{F}$ is transfer matrix based on t time, and ω_t is noise matrix, which is obeyed zero mean value and covariance matrix δ. $\mathbf{F}$is given by

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

Algorithm 1

Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^Q)$ with weights θ^Q and θ^μ
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1,M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t=1,T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random mini batch of N transition (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss:
      $L = \frac{1}{N}\Sigma_i(y_i - Q(s_i, a_i|\theta^Q)^2)$
    Update the actor policy using the sampled gradient:
      $\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$
    Update the target networks:
      $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$
      $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
    The dispatch center dispatches nodes to perform tracking tasks
  end for
end for

where T is the interval between consecutive sensor measurements.

Therefore, according to Kalman filtering, the possible position of the target at time $t+1$ can be estimated from time t, thereby improving the accuracy of system tracking.

In addition to adding the Kalman filter to predict the target moving trajectory in the above system, a one-to-one model of the mobile tracking node and the target node is also established, it is shown in Fig. 4: where M is the mobile node, T is the target node, ν_m is the linear speed of the mobile node, θ_m is the speed direction angle of the mobile node, ν_t is the speed direction angle of the target node, θ_t is the speed direction angle of the target node, and $\delta = \arctan(\frac{y_t - y_m}{x_t - x_m})$ is the mobile node The angle of sight between the target node and the kinematic model between the two nodes is as follows:

$$\begin{aligned} \widetilde{x} &= \nu_i \cos(\theta_i) \\ \widetilde{y} &= \nu_i \sin(\theta_i) \\ \widetilde{\theta_i} &= \mu_i \\ \psi_i &= \delta - \theta_i \end{aligned} \tag{12}$$

where i is the mobile node M and the target node T, (x_i, y_i) is the node position, θ_i is the direction angle, μ_i is the steering angle, $\mu_i \in [-\mu_{imax}, -\mu_{imax}]$, ν_i is the node moving speed, and ψ_i is the deviation between the speed direction angle and the sight angle.

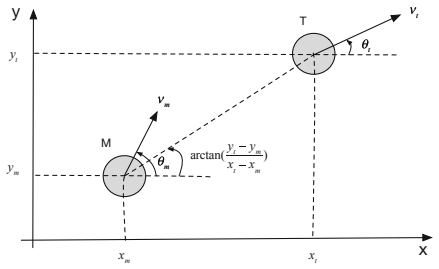

Fig. 4. Tracking model of mobile node and target node.

4.2 The Algorithm Flow

In the target tracking process, when the target passes through a certain monitoring area, it can only wake up the sensor nodes and cluster heads in the small area where the target is located, and the remaining sensor nodes are idle, so as to save energy. The monitoring center analyzes the measurement data from the sensor nodes participating in the observation task at time k, and predicts that the target may move at the next $k + 1$, so that the node is in the inspection state for easy activation and tracking. Once the target leaves the current small area, the cluster head node will pass the target state information to the next activated cluster head at the last sampling moment.

The pseudocode of the proposed algorithm is as Algorithm 1, which is using ddpg for scheduling decisions

5 Simulation and Analysis

In order to evaluate the proposed target tracking strategy, we have gathered multitude simulations. In this section, we verify proposed scheme validation and evaluate the performance of strategy through Python 3.7. combined with TensorFlow and Pyglet modules to build a target tracking simulation platform. Among them, TensorFlow is a deep learning framework for computing in the form of a computing graph, and Pyglet is a cross-platform window and multimedia library for Python, used to develop games and other visually rich applications.

Table 1. Simulation parameters of MC-SSN environment

Parameters description	Value
Area of each region	400 m * 400 m
Number of target nodes	1
Velocity of target nodes	10 m/s
Number of sensor nodes	100
Velocity of sensor nodes	20 m/s
Maximum tolerate delay of target node	3 s
Primary energy of each sensor node	400 J
Energy consumption for static nodes	0.1 J/ unit time
Energy consumption for moving nodes	0.8 J/ unit time
Learning rate for actor and critic	0.001
Discount factor	0.9
Size of min-bath	32
Size of replay memory	500

5.1 Simulation Setup

We set up a moving target tracking scenario, where the entire area is 400 m $*$ 400 m. The number of target nodes $\mathcal{T} = 1$, the number of internal mobile sensor nodes $\mathcal{N} = 100$, the static node sensor nodes set at each edge $\mathcal{M} = 11$. We set the energy of each sensor node $p_i = 400\,\text{J}, \forall_i \subset \mathcal{N}$. Let the target enter the monitoring from any angle), and the movement trajectory is randomly distributed. Not only that, the energy consumption of the static node per unit time length t is also considered to be 0.1 J, while the energy consumption of each motion sensor is 0.6 J. For clarity, other simulation parameters of M-MMT used to perform critical tasks are summarized in Table 1.

5.2 Results Analysis

In this subsection, we further study the performance of in target tracking based on DDPG, which comparing with the random selection scheduling scheme. The random selection scheduling scheme, which does not involve the reward and sensor nodes are selected through random probability.

When we choose the sensor node to track the target, we use the Kalman filter to predict the trajectory of the target, so as to get the wake-up nodes according to the position where the target may appear at the next moment. This article uses this point to define one of the contribution value of each node. When multiple nodes detect the target at the same time, the node with high contribution value has priority tracking.

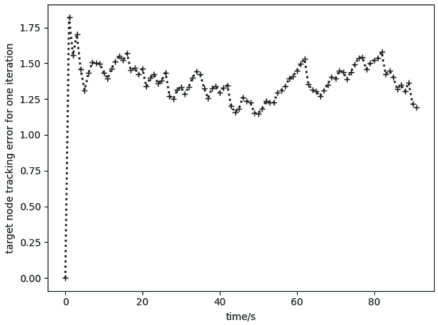

Fig. 5. Target node tracking error for one iteration.

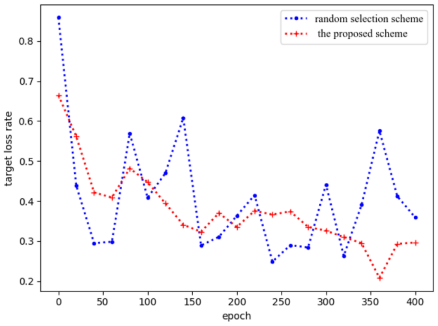

Fig. 6. Target loss rate of irregular pattern with by different scheduling schemes.

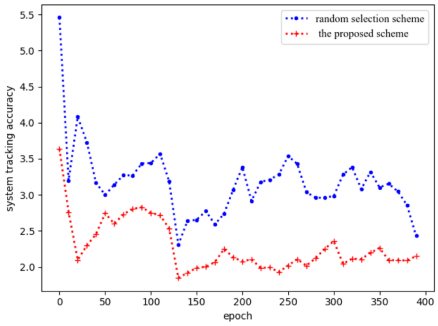

Fig. 7. Cumulative average tracking error by different scheduling schemes.

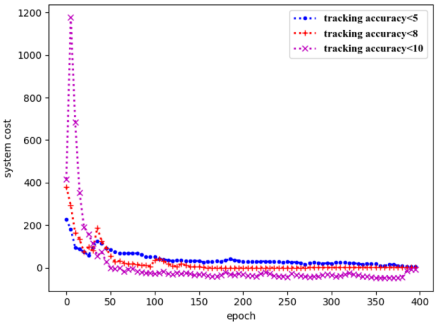

Fig. 8. Average system cost achieved by different tracking accuracy.

Figure 5 shows, as time increases, the trajectory prediction error of the target node becomes smaller and smaller, which can ensure that the contribution of each node is more reliable in iterative learning.

One of the important principles of the target tracking task is to minimize the loss of the target, so we first measure the effectiveness of target tracking based on DDPG based on the target loss rate. Only when the target loss rate is low enough, the target tracking algorithm can really play a role. Note that the target loss rate is expressed by dividing the time step of losing the target by the total time step.

In Fig. 6, we can see the random selection scheduling scheme does not perform well, since the strategy selection of sensors is low. Because there is no learning choice, the target tracking is triggered by the detection of the sensor. Whether the target is lost is related to the target's motion trajectory, and the sensor is very active in decision tracking. Compared with random selection scheduling scheme, the target loss rate of the intelligent scheduling scheme gradually decreases with

the increase of the number of iterations, and reaches a certain level of 0.2. There may be room for optimization in the future.

Figure 7 shows a comparison of the cumulative system tracking errors of different scheduling schemes based on the DDPG learning algorithm. When a target enters the monitoring area, as the number of iterations increases, both scheduling schemes can approach its stable cumulative tracking error. We can draw the following observations from Fig. 7. First, compared with the proposed scheme, the tracking accuracy of the randomly selected scheduling scheme is lower. The reason is that the sink node needs to adjust the state of each sensor node, where the self-energy and tracking capabilities determine the best scheduling, rather than random selection. Second, intelligent scheduling scheme can enhance tracking accuracy in he complicated environment. Thirdly, our proposed scheme can improve tracking accuracy 23.5% approximately, compared with random selection scheduling scheme.

In Fig. 8, we compare the cumulative system cost achieved by different tracking accuracy in intelligent scheduling scheme. Firstly, We compared the learning costs of the system for accuracy less than 5 m, 8 m, and 10 m, respectively. It can be seen that when the target tracking accuracy is high, the value is small, the system cost is significantly more. Then, In the iterative learning of MC-SSN networks, when the episode reaches 50, the overhead gradually slows down, and when all three accuracy are around 100, the system learning overhead tends to be fixed. Moreover, the results demonstrates the long term tracking is guaranteed based the proposed scheme, especially in the critical missions.

6 Conclusion

This paper proposes a novel collaborative moving system based DDPG for MC-SSN to track target, consisting of strategy selection layer, and diffusion strategy. Our objective is to maximize the best tracking with low energy consumption and high tracking accuracy. We introduced the training process of DDPG, and deduced the corresponding training and verification algorithms. Combining the continuous action space of DDPG, we proposed an algorithm for tracking moving target nodes, which makes tracking more accurate and training faster. It is proved that MC-SSN based on DDPG algorithm can learn continuous target tracking policy through trial and error mode without a large number of artificially produced data. After a series of simulation verification, the numerical results show that the scheme can ensure a certain tracking accuracy, and reduce the scheduling delay and system energy consumption.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China (Grants No. 61731006) and Zhongshan City Team Project (Grant No. 180809162197874).

References

1. Suriyachai, P., Roedig, U., Scott, A.: A survey of mac protocols for mission-critical applications in wireless sensor networks. IEEE Commun. Surv. Tutor. **14**(99), 1–25 (2012)
2. Liang, J., Hu, Y., Liu, H., Mao, C.: Fuzzy clustering in radar sensor networks for target detection. Ad Hoc Netw. **58**, 150–159 (2016)
3. Qiao, G., Leng, S., Zhang, K., He, Y.: Collaborative task offloading in vehicular edge multi-access networks. IEEE Commun. Mag. **56**(8), 48–54 (2018)
4. Bhatti, S., Xu, J.: Survey of target tracking protocols using wireless sensor network. In: 2009 Fifth International Conference on Wireless and Mobile Communications, pp. 110–115. IEEE (2009)
5. Lin, C.Y., Peng, W.C., Tseng, Y.C.: Efficient in-network moving object tracking in wireless sensor networks. IEEE Trans. Mob. Comput. **5**(8), 1044–1056 (2006)
6. Wälchli, M., Skoczylas, P., Meer, M., Braun, T.: Distributed event localization and tracking with wireless sensors. In: Boavida, F., Monteiro, E., Mascolo, S., Koucheryavy, Y. (eds.) WWIC 2007. LNCS, vol. 4517, pp. 247–258. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72697-5_21
7. Xu, Y., Lee, W.C.: On localized prediction for power efficient object tracking in sensor networks. In: 23rd International Conference on Distributed Computing Systems Workshops, Proceedings, pp. 434–439. IEEE (2003)
8. Chen, Y.S., Ann, S.Y., Lin, Y.W.: VE-mobicast: a variant-egg-based mobicast routing protocol for sensornets. Wirel. Netw. **14**(2), 199–218 (2008). https://doi.org/10.1007/s11276-006-9957-9
9. Jin, G.Y., Lu, X.Y., Park, M.S.: Dynamic clustering for object tracking in wireless sensor networks. In: Youn, H.Y., Kim, M., Morikawa, H. (eds.) UCS 2006. LNCS, vol. 4239, pp. 200–209. Springer, Heidelberg (2006). https://doi.org/10.1007/11890348_16
10. Luo, J., Zhang, Z., Liu, C., Luo, H.: Reliable and cooperative target tracking based on WSN and WiFi in indoor wireless networks. IEEE Access **6**, 24846–24855 (2018)
11. Bhuiyan, M.Z.A., Wang, G., Vasilakos, A.V.: Local area prediction-based mobile target tracking in wireless sensor networks. IEEE Trans. Comput. **64**(7), 1968–1982 (2014)
12. Yang, X., Zhang, W.A., Yu, L., Xing, K.: Multi-rate distributed fusion estimation for sensor network-based target tracking. IEEE Sens. J. **16**(5), 1233–1242 (2015)
13. Mahboubi, H., Masoudimansour, W., Aghdam, A.G., Sayrafian-Pour, K.: An energy-efficient target-tracking strategy for mobile sensor networks. IEEE Trans. Cybern. **47**(2), 1–13 (2016)
14. Lee, S.J., Park, S.S., Choi, H.L.: Potential game-based non-myopic sensor network planning for multi-target tracking. IEEE Access **6**, 79245–79257 (2018)
15. Vazquez-Olguin, M., Shmaliy, Y.S., Ibarra-Manzano, O.G.: Distributed unbiased FIR filtering with average consensus on measurements for WSNs. IEEE Trans. Ind. Inform. **13**(3), 1440–1447 (2017)
16. Fan, B., Leng, S., Yang, K.: A dynamic bandwidth allocation algorithm in mobile networks with big data of users and networks. IEEE Netw. **30**(1), 6–10 (2016)
17. Doya, K.: Reinforcement learning in continuous time and space. Neural Comput. **12**(1), 219–245 (2000)
18. Liang, J., Yu, X., Li, H.: Collaborative energy-efficient moving in internet of things: genetic fuzzy tree versus neural networks. IEEE Internet Things J. **6**(4), 6070–6078 (2018)

19. Liu, C., Lonsberry, A.G., Nandor, M.J., Audu, M.L., Lonsberry, A.J., Quinn, R.D.: Implementation of deep deterministic policy gradients for controlling dynamic bipedal walking. In: Conference on Biomimetic and Biohybrid Systems, vol. 4, no. 1, p. 28 (2018)
20. Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. In: 31st International Conference on Machine Learning, ICML (2014)
21. Pillutla, L.S.: Network coding based distributed indoor target tracking using wireless sensor networks. Wirel. Pers. Commun. **96**(3), 3673–3691 (2017). https://doi.org/10.1007/s11277-017-4069-7
22. Huang, Y., Liang, W., Yu, H.B., Xiao, Y.: Target tracking based on a distributed particle filter in underwater sensor networks. Wirel. Commun. Mob. Comput. **8**(8), 1023–1033 (2008)
23. Sozer, E.M., Stojanovic, M., Proakis, J.G.: Underwater acoustic networks. IEEE J. Oceanic Eng. **25**(1), 72–83 (2000)
24. Rault, T., Bouabdallah, A., Challal, Y.: Energy efficiency in wireless sensor networks: a top-down survey. Comput. Netw. **67**, 104–122 (2014)