Anti-jamming mobile control using QoS-based reinforcement learning

Naomi Kuze\textsuperscript{a)}, Taishi Nakashima, Toshimitsu Ushio\textsuperscript{b)}
Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama-cho, Toyonaka, Osaka 560-8531, Japan
\textsuperscript{a)} kuze@sys.es.osaka-u.ac.jp, \textsuperscript{b)} ushio@sys.es.osaka-u.ac.jp

Abstract: In cyber-physical systems, the reliable data communications between the physical systems and computer systems are inevitable, and it is an essential issue to protect the systems from jamming attacks that obstruct wireless communication. Moreover, for practical uses of the systems, Quality of Service (QoS) is also important. In this paper, we consider a mobile robot and propose an anti-jamming mobile control system that uses QoS-based reinforcement learning. We show that the control system can find a policy that provides planning with a short path while avoiding the jamming attack.

Keywords: Cyber physical system, anti-jamming, mobile control, Quality of Service (QoS), reinforcement learning

Classification: Navigation, guidance and control systems

References

[1] L. Xiao, et al., “Two-dimensional anti-jamming mobile communication based on reinforcement learning,” IEEE Transactions on Vehicular Technology, vol. 67, no. 10, pp. 9499–9512, October 2018. DOI:10.1109/TVT.2018.2856854
[2] V. Saritha, et al., “Efficient multipath routing protocol with quality of service for mobile ad hoc networks,” Proc. IEEE International Conference on Communications (ICC), Kansas City, MO, USA, pp. 1–6, May 2018. DOI:10.1109/ICC.2018.8422385
[3] R. S. Sutton, and A. G. Barto, Reinforcement learning: an introduction, MIT press, Cambridge, 2018.
[4] C. J. C. H. Watkins, and P. Dayan, “Q-learning,” Machine Learning, vol. 8, no. 3, pp. 279–292, May 1992. DOI:10.1007/BF00999269

1 Introduction

Due to the growth of cyber-physical systems, wireless communication devices and services that collect information of physical systems are developed rapidly in recent years. Especially, mobile robots (referred to as ”robots” in this paper) that move for collecting information with wireless communication are used in various services. However, the increment of jamming attacks that
obstruct wireless communication has become a serious problem, and therefore, it is important to develop systems with a high robustness against such attacks. In the conventional systems, wireless communication devices avoid jamming attacks by using spread spectrum technologies, which are not an essential solution. On the contrary, robots can avoid jamming attacks by moving to positions where the effect of jamming attacks is low.

In previous work [1], the authors have proposed a communication system with a robot for avoiding jamming attacks by using deep reinforcement learning. In the system, a controller finds a mobile policy of a robot based on the reward that is calculated in accordance with signal-to-noise ratio (SNR). However, for practical uses of the systems, it is essential to consider their performance in an application level and to satisfy users’ requirements. Therefore, we propose an anti-jamming mobile control system that uses reinforcement learning based on Quality of Service (QoS) [2]. QoS is the metric of the performance of the systems in the application level, which is used for measuring the performance from the users’ perspective. By calculating the reward of the robot based on QoS, the controller can find a policy that the robot moves and communicates while not only avoiding jamming attacks but keeping the system performance high. In this paper, we demonstrate the usefulness of the proposed system through simulation evaluation.

2 Proposal

2.1 Overview

We propose an anti-jamming mobile control system that uses QoS-based reinforcement control. The architecture of the proposed system is shown in Fig. 1.

We assume that robots move in a field where there are attackers. Moreover, there are home positions (HPs) of robots, sensing points (SPs), and access points (APs) in the field. A robot starts from its corresponding HP, collects data in SPs, sends collected data to an AP, and returns to the HP. An attacker generating interference electric waves interferes wireless communication between a robot and an AP. In this paper, we assume that there are a single robot and a single attacker for simplicity. Moreover, we also assume
that there are a single HP, and a single SP.

In the central server, several applications are installed and it calculates the reward of the robot based on QoS of each application. In this paper, we assume that a single application is installed for simplicity.

In the proposed system, the planner receives the reward of the robot from the central server and selects the next task of the robot. Then, the controller finds the mobile policy of the robot according to the selected task by using reinforcement learning. See Section 2.2.1 for the detail of their roll.

### 2.2 System model

We assume that the behavior of the robot is modeled as a Markov decision process $\mathcal{M} = (\mathcal{S} = \mathcal{X} \times \mathcal{T}, \mathcal{A}, P_0, P, r)$, where $\mathcal{S} = \mathcal{X} \times \mathcal{T}$ is a set of the robot’s states. $\mathcal{X}$ is a set of positions that the robot can move to, and $\mathcal{T} = \{0, 1, 2\}$ is a set of task complement flags (TCFs) (see Section 2.2.1 for details). $\mathcal{A} = \{\text{up, down, left, right, send}\}$ is a set of the actions that the robot can select and we define $\mathcal{A}(s) = \{a \in \mathcal{A} | \exists s' \in \mathcal{S}, P(s'|s, a) \neq 0\}$ as a set of actions that the robot can select at state $s \in \mathcal{S}$. $P_0 : \mathcal{S} \mapsto [0, \infty)$ is a probability density function that determines the initial state of the robot. $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, \infty)$ is a probability density function, that is, $P(s'|s, a)$ represents the probability of the next-step state $s'$ given the current state $s$ and the action $a$. $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto (-\infty, \infty)$ is a function that determines a reward, that is, $r(s, a, s')$ is a reward for the current state $s$, the action $a$, and the next-step state $s'$. A state, the action, and the reward of the robot at time $k \in \{0, 1, \cdots\}$ are denoted by $s_k$, $a_k$, and $r_k$, respectively.

#### 2.2.1 Policy

In the proposed system, the planner firstly selects the task of the robot, and the controller secondly finds the mobile policy of the robot according to the selected task.

The planner selects a task from three tasks that the robot has to achieve. The tasks of the robot are

1. to move from the HP to the SP and collect data from the field (HP-SP task),
2. to move from the SP to an AP and send the collected data to the access point (SP-AP task), and
3. to return to the HP from the AP (if the robot cannot receive any ACK packet from the AP, it sends again the collected data to the access point) (AP-HP task).

TCF 0 indicates that the robot does not achieve any task, and the HP-SP task is selected as the next task. TCF 1 indicates that the robot achieves the HP-SP task, and the SP-AP task is selected as the next task. TCF 2 indicates that the robot achieves the HP-SP and the SP-AP tasks, and the AP-HP task is selected as the next task. When the robot achieves all tasks, the TCF is reset to 0.
Based on reinforcement learning [3], the controller determines a policy of the robot according to the selected task. In this paper, the controller finds a policy by using the Q-learning algorithm [4], a model-free reinforcement learning algorithm.

The controller selects action $a_k$ of the robot at time $k$ with the current state $s_k$ in accordance with the policy $\pi(a_k|s_k)$. The purpose of the reinforcement learning is to find the optimal policy $\pi^*$ that maximizes the total reward $G$ in the future. At time $k$, $G$ is given by

$$G_k = \sum_{l=0}^{\infty} \gamma^l r_{k+l+1} = r_{k+1} + \gamma r_{k+2} + \gamma^2 r_{k+3} + \cdots,$$

where $\gamma \in [0, 1]$ is a discount factor, which values earlier rewards higher than later ones.

For finding the optimal policy, the controller uses the Bellman equation. Given a policy $\pi$, the Bellman equation is

$$V^\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s' \in \mathcal{S}} P(s'|s,a) \left( r(s,a,s') + \gamma V^\pi(s') \right),$$

where $V^\pi(s) = E^\pi[G_k|s_k = s]$ is the state value function. Likewise, the action value function $Q^\pi(s,a) = E^\pi[G_k|s_k = s, a_k = a]$ satisfies

$$Q^\pi(s,a) = \sum_{s' \in \mathcal{S}} P(s'|s,a) \left( r(s,a,s') + \sum_{a' \in \mathcal{A}(s')} \gamma P(a'|s') Q^\pi(s',a') \right).$$

The optimal state value function $V^*(s) = \max_\pi V^\pi(s)$ and the optimal action value function $Q^*(s,a) = \max_\pi Q^\pi(s,a)$ with the optimal policy $\pi^*$ satisfy the following equations.

$$V^*(s) = \max_{a \in \mathcal{A}} \left( \sum_{s' \in \mathcal{S}} P(s'|s,a) \left( r(s,a,s') + \gamma V^*(s') \right) \right),$$

$$Q^*(s,a) = \sum_{s' \in \mathcal{S}} \left( P(s'|s,a) \left( r(s,a,s') + \gamma \max_{a' \in \mathcal{A}} Q^*(s',a') \right) \right).$$

For finding the optimal policy, with Q-learning algorithm, the controller estimates the action value function by updating Q-function $Q : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ by Eq. (6).

$$Q(s_k,a_k) \leftarrow Q(s_k,a_k) + \alpha \left( r_k + \gamma \max_{a_{k+1} \in \mathcal{A}} Q(s_{k+1},a_{k+1}) - Q(s_k,a_k) \right),$$

where $r_k$ corresponds to a reward function that is defined according to the selected task.

- **HP-SP task**: if the mobile robot arrives at the SP, the reward is +1. Otherwise, the reward is 0.
- **SP-AP task**: if the mobile robot successfully sends data to the access point, the reward is given by the amount of QoS. In this paper, we define the amount of QoS as the communication success rate with the current state and the selected action. Otherwise, the reward is 0.
Fig. 2. An example of trajectories of the robot. The yellow, green, and blue arrows show the trajectories of the HP-SP task, the SP-AP task, and the AP-HP task, respectively.

- AP-HP task: if the mobile robot returns to the HP, the reward is +1. Otherwise, the reward is 0.

Then, the controller determines the policy $\pi$ of the robot in accordance with the estimated action value function by Eq. (7) ($\varepsilon$-greedy algorithm).

$$\pi(s_k|a_k) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A|}, & \text{if } a_k = \arg \max_{a\in A} Q(s_k, a_k), \\ \frac{\varepsilon}{|A|}, & \text{otherwise}, \end{cases}$$

(7)

where $\varepsilon \in (0, 1)$ is the probability that the robot selects its action randomly.

3 Performance evaluation

In this section, we conduct simulation evaluation for demonstrating the adaptivity of the proposed system to the jamming attacks. We use a field consisting of $5 \times 5$ cells (Fig. 2). The right-bottom cell (S20) is the HP and the left-bottom cell (S24) is the SP. The robot sends collected data to the access point in cells S6, S8, 11, 13, and 16, i.e., these cells are APs.

In each episode, the robot starts from the HP cell, moves to the SP cell for collecting data, moves to an AP cell for sending data to the access point, and then returns to the HP cell. In these AP cells, the robot fails to send data to the access point due to jamming attacks by attackers.

3.1 Results and Discussion

First, we consider the case that an attacker conduct jamming attacks. The communication success rates of cells ($S_8$, $S_{7, 13}$, $S_{6, 12, 18}$, $S_{11, 17}$, $S_{16}$) are (1.0, 0.6, 0.4, 0.2, 0.0). Fig. 2 shows an example of trajectories of the robot at 100 and 1,000 episodes. At 100 episode, the robot sends data to the access point at cell S12 whose communication success rate is low and the trajectory of the robot is not optimal. On the contrary, at 1,000 episode, the robot sends data to the access point at cell S8 whose communication success rate is the highest as a result of the QoS-based reward design. Moreover, the trajectory of the robot is optimal. This result indicates
that the agent can find the mobile policy of the robot that avoids jamming attacks by using QoS-based reinforcement learning.

Next, we consider the case that the position of the jamming attacks is changed because jamming attackers often change their strategies for achieving their attacks. At the beginning of simulation, the communication success rates of cells \((\{S8\}, \{S7, S13\}, \{S6, S12, S18\}, \{S11, S17\}, \{S16\})\) are \((1.0, 0.6, 0.4, 0.2, 0.0)\). At 500 episode, the communication success rate of cell change to \((0.0, 0.2, 0.4, 0.6, 1.0)\). Fig. 3 shows the changes of the trajectory length from the HP to the SP, from the SP to a AP, and from the AP to the HP. The figure shows that the trajectory lengths once converge in about 200 episodes, which indicates the agent finds the mobile policy that maximizes the reward. When the position of the jamming attacker changes at 500 episode, the trajectory lengths temporarily become large. This is because the agent tries to find another trajectory that maximizes the reward according to the changes of communication success rates. Then, the trajectory lengths converge again soon with QoS-based reinforcement learning. In conclusion, the proposed system can adapt to the changes of attacker’s strategies.

4 Conclusion

We have proposed an anti-jamming mobile control system using QoS-based reinforcement learning, and have shown that the controller can find a trajectory with a higher communication success rate and a shorter path length.

As future work, we will consider the application of the proposed system to more complicated and real situations. In details, we consider situations with several robots in a larger and more complicated field. For that purpose, we will introduce the deep reinforcement learning to the proposed system.

Acknowledgments

This research was supported by Grant-in-Aid for Early-Career Scientists No. 18034267 from the Japan Society for the Promotion of Science (JSPS).