SE 701 Project Report

An Optimal Control Approach to the Multi-Agent Persistent Monitoring Problem

Jingci Li, Qingyang Long, Zekun Wang

04/30/2020

1. Background

From the introduction of concepts to the appearance of solid robots, robot technology has always been a hot research topic and an important development trend of future science and technology. Nowadays, robot technology is widely used in storage, logistics, industrial automation, military and many other fields. In order to accomplish a certain task in these fields, especially to solve some special tasks with distributed space or time, the robot system must have high efficiency and stability. It is obviously unwise to combine the complex function configuration with a robot, so the multi robot system becomes the research core. In short, it is to cooperate with multiple robots to complete a series of efficient tasks.[1]

Robot cooperation scenarios could be divided into two categories: static environment and dynamic environment. The persistent monitoring problem arises in a dynamic scenario, in which cooperating mobile agents must monitor the dynamic changing environment. The paper that we researched for this project aimed at providing an optimal control solution to the persistent monitoring problem in 1-D space through minimizing a metric of uncertainty over the environment.[2]

To simulate the persistent monitoring problem, two crucial aspects are necessary for consideration: how to simplify the environment model due to limited computational capability as well as how to present the complexity of a dynamic environment.[3] For the first aspect, instead of monitoring all discrete points within the environment, sampling points were assigned, and those points would be monitored persistently. And the monitoring process of each individual point could be metaphorized as enqueuing and dequeuing in a “uncertainty queue”. For the second aspect, the complexity of each individual sampling point was presented by a function of space and time. Also, intuitively, such complexity would be inversely proportional to the event detection probability.[4]

2. Problem description

In this project, we mainly focus on one dimensional persistent monitoring problem. There would be N mobile agents and M sampling points.[5] As we can see on figure 1, the blue dot is a mobile agent, and those little crosses on the axis are sampling points.

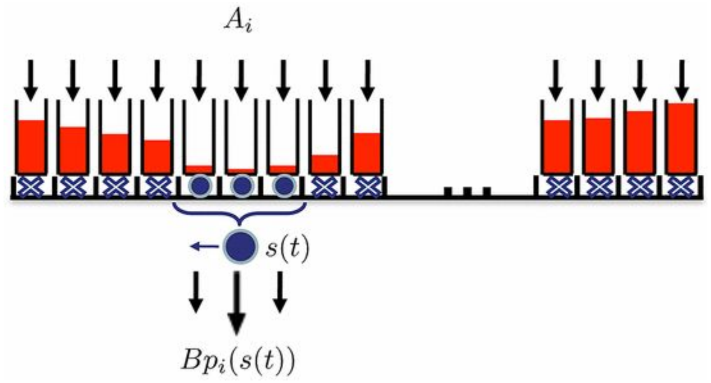

Figure 1 the structure of the persistent monitoring problem model

We assume that the agent can control its direction and speed, so there could only be three kinds of behavior, moving in full speed, waiting, and switching direction. The agents are constrained within the 1-D interval [0, L]. The derivative of agent position s(t) equals the speed u(t) of such agent, and speed is constrained in 1.

We associate with every point $x \in [0, L]$, a function that $P_n(x, s_n)$ measures the probability that an event at location x is detected by agent n . We also assume that $P_n(x, s_n) = 1$ if $x = s_n$, and that $P_n(x, s_n)$ is monotonically non increasing in the distance $|x - s_n|$ between x and s_n, thus capturing the reduced effectiveness of a sensor over its range which we consider to be finite and denoted by r_n .Therefore, $P_n(x, s_n)$ we set =0 when $|x - s_n| > r_n$. Although our analysis is not affected by the precise sensing model $P_n(x, s_n)$, we will limit ourselves to a linear decay model as follows:

$$p_n(x, s_n) = \begin{cases} 1 - \frac{|x - s_n|}{r_n}, & \text{if } |x - s_n| \le r_n \\ 0, & \text{if } |x - s_n| > r_n . \end{cases} \tag{1}$$

Then, we consider a set of points $\alpha_i, i = 1, 2, \ldots, M$, and associate a time-varying measure of uncertainty with each point α_i, which we denote by $R_i(t)$. Without loss of generality, we assume $0 \le \alpha_1 \le \cdots \le \alpha_M \le L$ and, to simplify notation, we set $p_{n,i}(s_n(t)) \equiv p_n(\alpha_i, s_n(t))$. This would help us to simplify the calculation. Alternatively, we may consider a partition of [0,L] into M intervals whose center points are the following equation (2):

$$\alpha_i = \left(\frac{(2i - 1) L}{2M} \right), i = 1, 2, \ldots, M \tag{2}$$

We can then set $p_n(x, s_n(t)) = p_{n,i}(s_n(t))$ for all $x \in [\alpha_i - (L/2M), \alpha_i + (L/2M)]$. Therefore, the joint probability of detecting an event at location $x \in [\alpha_i - (L/2M), \alpha_i + (L/2M)]$ by all the Formula agents simultaneously (assuming detection independence) is

$$P_i(\mathbf{s}(t)) = 1 - \prod_{n=1}^{N} [1 - p_{n,i}(s_n(t))] \tag{3}$$

So basically the agent would first move in full speed until it reaches a switching point, then it would wait here for a while, after that it would switch the direction and start a new round of monitoring.

As for the environment, the complexity at a certain sampling point would decrease at a fixed rate if there is any agent nearby. Otherwise, the complexity would monotonically increase. Then our target is to find the optimal switching points and corresponding waiting time.It can be expressed by the following equation(4):

$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0,\ A_i \le BP_i(\mathbf{s}(t)) \\ A_i - BP_i(\mathbf{s}(t)) & \text{otherwise} \end{cases} \tag{4}$$

Then, the problem can be changed to the following optimal control problem (5)

$$\min_{u(t)} \quad J = \frac{1}{T}\int_0^T \sum_{i=1}^{M} R_i(t)dt \tag{5}$$

St.

$$\dot{s}_n(t) = u_n(t)$$

$$|u_n(t)| \le 1, n = 1, ..., N$$

$$a \le s(t) \le b, a \ge 0, b \le L$$

$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0,\ A_i \le BP_i(\mathbf{s}(t)) \\ A_i - BP_i(\mathbf{s}(t)) & \text{otherwise} \end{cases}$$

After studying the algorithm carefully, we find that there are some places we are interested in and can be improved.

So in this project, we mainly focus on

(1) The influence to monitor motion and loss function when it occurs to multiple agents.

(2) The influence of monitor radius on monitor motion state and loss function

3. Approach

As written in the proposal, we decided to simulate the problem using Matlab, and divided the simulation into 4 steps:

1. Function to get P
2. Function to get u by PMP
3. Function to output simulation and result
4. Function to get u by API

Firstly, we initialized the numerical value of A, B, L, s and r. Then, in order to simplify the simulation process, the value max of (1 - |X - s(j)|/r, 0) was used as p_n , instead of equation (1). Then the sum of p_n is applied to each s to get $P_i(s(t))$.

Secondly, based on PMP, we obtained $\lambda_i(t)$ = -1*t, and transferred the solution of PMP problem into the following equation (6) to get the speed u of $s_n(t)$:

$$\lambda_{sn}dot = -(B/r) * \sum_{i<x} \lambda i * (P - p_{d=n}) + (B/r) * \sum_{i>x} \lambda i * (P - p_{d=n}) \qquad (6)$$

After obtaining the summation of $\lambda_{sn}dot$, we can determine that

u = 1, if $\lambda_{sn}dot < 0$

u = -1, if $\lambda_{sn}dot > 0$ (7)

Thirdly, to output the simulation as a dynamic diagram, we created a new figure window to keep output the location of each R and s. In order to print the cost and path, we imported R_path and s_path to record the cost and path for every timing.

At last, we tried to analyze the API method and simulated the algorithm. We introduced two small positive values σ and ε and defined D_n=a+(2n-1/2N)(b-a), and set

$$\theta_{n,\xi} = D_n - \sigma \qquad if\ \xi even \qquad (8)$$

$$\theta_{n,\xi} = D_n + \sigma \qquad if\ \xi odd$$

We made a circulation to find $s_n(t)$ as the method, however, we cannot get a right number of $\theta_{n,\xi}$ by this method. So we did not apply API for most of our simulations.

4. Result and Analysis

Simulation Detail

The simulation of the one dimensional persistent monitoring problem is shown in figure 2. The x-axis is the one dimensional monitoring interval. Here the length L equals to 30, and the number of sampling points is 31. The red bars simulated the increasing environment complexity. Without detection by any agent, each of them should increase at a rate of A. In our simulation, two types of complexity increasing rate were applied, a fixed rate 10 and an uniform distributed rate. The blue and green balls on the x-axis are the simulated agents. The extended horizontal lines from the balls are the detection radius of the agents. Different radii were also carried out. When one agent moves close to one sampling point (within the detection radius), the complexity would decrease. The fixed complexity decreasing rate B in our simulation is 100.

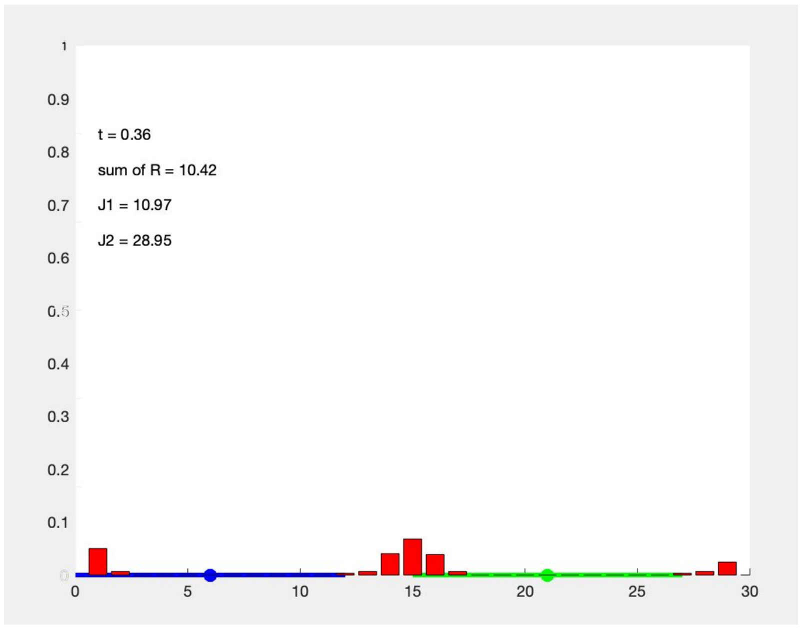

Figure 2 Simulation of 1-D persistent monitoring problem with two agents

Four stages of simulation were carried out:

1. Simulation of single agent persistent monitoring problem
2. Simulation of double agents starting with same and opposite direction
3. Simulation of double agents with different detection radius
4. Simulation of multiple agents

Firstly, single agent simulation for 1-D persistent monitoring problem was performed with different inflow rate A_i, one is with fixed increasing rate and the other one is with uniform distributed environment complexity increasing rate. The random function 'randi' in Matlab was applied. The optimized paths and the cost function are shown in figure 3. The left two charts are the results of fixed complexity increasing rate, and the right two are those of random increasing rate. We noticed that the cost versus time curve of the fix rate trail is relatively smoother. Though the result suggests that different environments could influence the effectiveness of this algorithm, both the optimized paths are similar in a periodic motion. Hence we conclude that the environment could influence the result of this approach, however the influence is relatively small and could be ignored. For the rest simulation stages, a fixed increasing rate was performed.

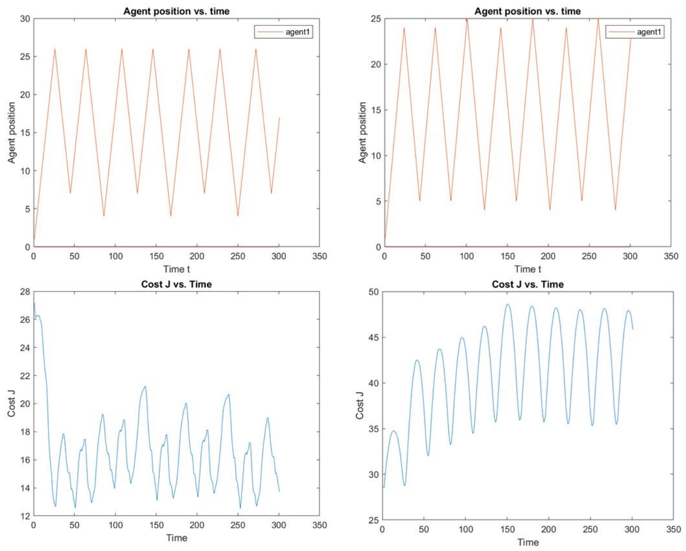

Figure 3 The simulation of the one dimensional persistent monitoring problem

Secondly, double agent simulations with different starting modes were carried out. We aimed at discussing the relationship between starting positions and the final optimized path pattern. Both agents started at the origin in the first trial. And in the second trial, the agents started from both ends of the interval. The results are shown in figure 4. The slight difference between these two groups is that the cost chart in the second trial converges to zero immediately, while it took the first group around one second to reach such a state. However, the similar optimized movement pattern between these two groups suggest that the starting point should not be a crucial factor of influence in this problem.

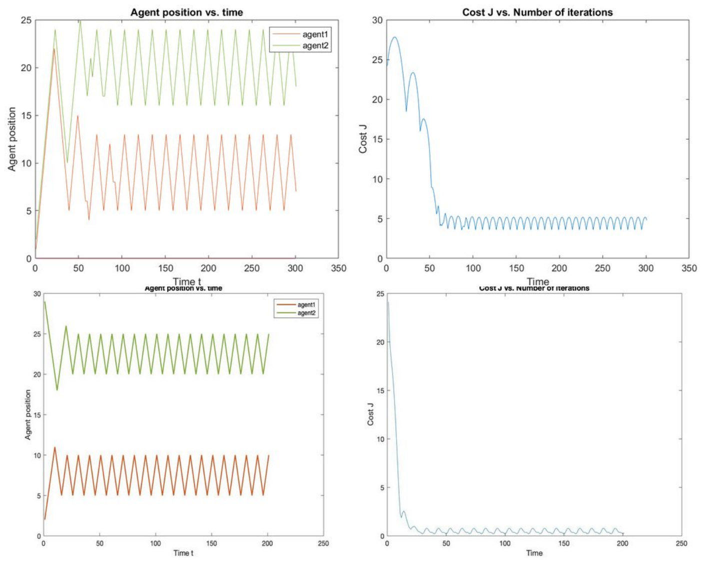

Figure 4 The relationship between starting positions and the final optimized path pattern

In the third stage, we focused on discussing the influence of detection radius in this problem. Two-agent simulations with different detection radii were performed. For the interval L = 30, three radii were applied in the simulation, four, eight and twelve (figure 5). We noticed that cost would increase linearly with small radius, as the detection range was too small to reduce the total complexity of this system. The cost would eventually converge with a larger detection radius. The radius threshold is L/2N.

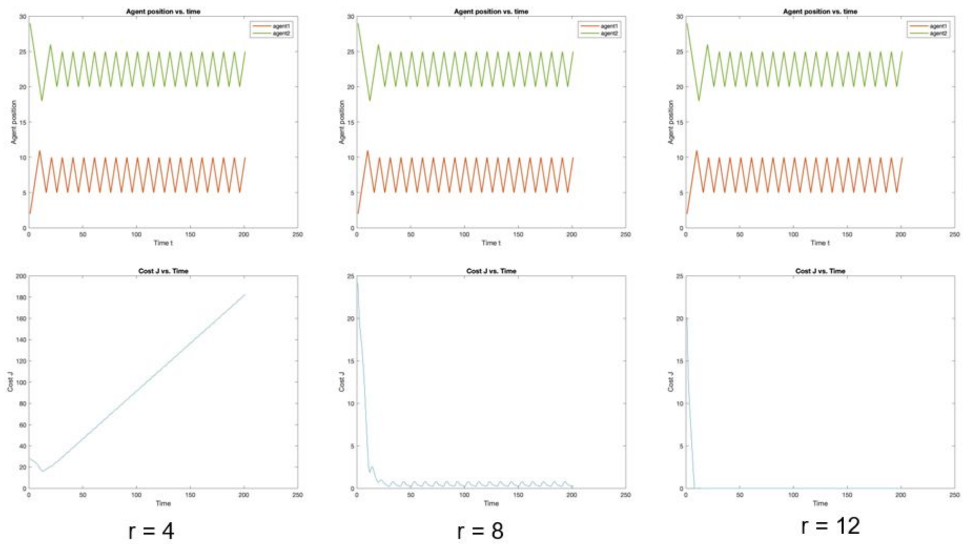

Figure 5 The influence of detection radius

For the fourth simulation, we added more agents into the system in order to test the robustness of this algorithm. As the number of agents increased, the pattern of movement remained unchanged.

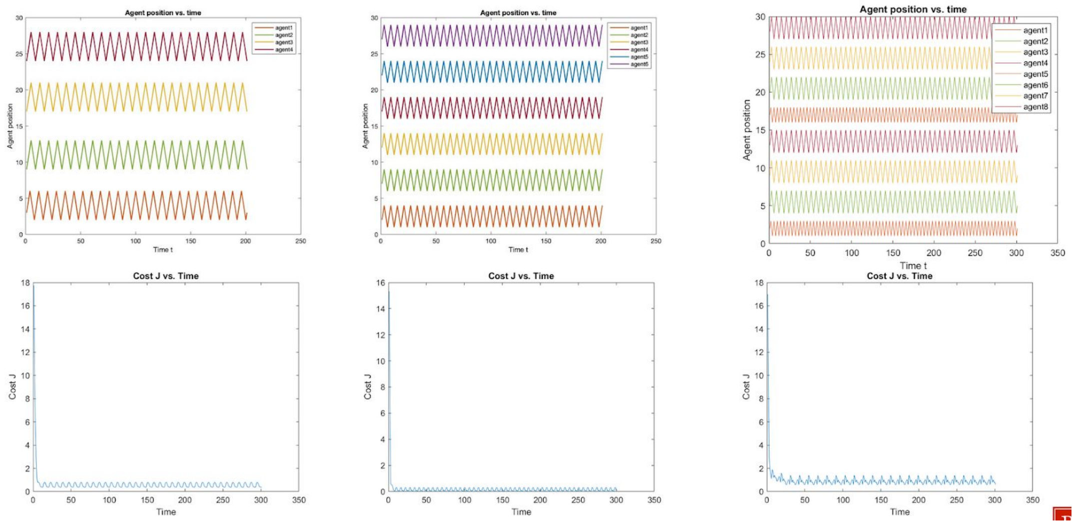

Figure 6 The influence to the curve between different number of agents

5. Conclusion

For this project, we simulated the algorithm in the research paper, which provides an optimal control approach to the multi-agent persistent monitoring problem. And we mainly focused on and explored the following two aspects: The influence in the number of agents and the detection radius.

Number of agents

The numerical result part of the paper from professor Cassandras provided simulation result of one and two agents. In our simulation, we pushed this limit to 8 agents. We can see that though more agents appear in this system, the result remains unchanged, especially the movement pattern of the agents. And the final result also follows the unmentioned constraints in the problem setup, that each agent moves without intersection and collision in the optimized path. In this way, the algorithm is robust in the multi-agent aspect.

Detection Radius

During simulation, we noticed that the effectiveness of the cost reduction is proportional to the size of detection radius for each and every robot. The larger the monitoring radius is, the faster the total complexity of this system will decrease. And such behavior accords with our life experience. Besides, a threshold for the radius is obvious. If the radius is larger than half the length of interval over the number of agents, the cost of the entire system would decrease, otherwise it might increase. It is not hard to imagine, as if the radius is larger than the threshold, the detection range could cover the entire interval. In this case, the cost of the system would eventually converge to zero, and the agents would stand still.

6. Limitation and Further Research

The main idea of this paper is the optimal control approach to the persistent monitoring problem in one dimension. And we tried to discuss how the radius of detection and number of agents would influence the cost during the project. This problem could be used in real life such as smart city control, in which case we need to consider the price for different types of agents.

Moreover, what we usually need to do is 2-D and 3-D modeling. So if for future research, one path is to explore the result when it comes to a team of cooperating agents in the higher dimensional environment. And that is actually what the authors did after publishing this paper.

However, there are some differences between the simulation we do and the paper did, such as the curve of the loss function. It is mainly because we did not consider some special case for A_i, that have an extremely big single point, as the result, the agent should stay there for a while to wait the R_i return to zero. We finish the function after the presentation as figure 7, the agent stays at s=1 for a while at T = 1.2 and T = 1.7.

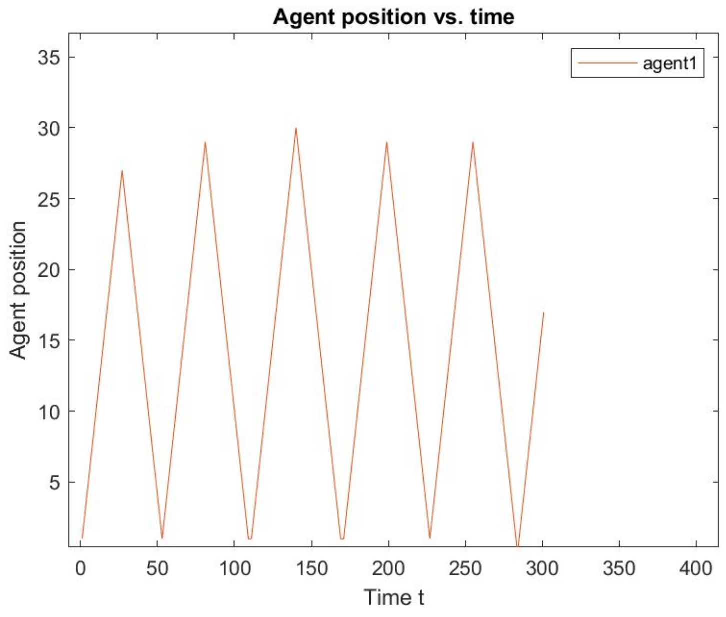

Figure 7 Special case for big A near zero point

Reference

1. I. Rekleitis , V. Lee-Shue , A. New and H. Choset "Limited communication, multi-robot team based coverage" Proc. IEEE Int. Conf. Robot. Autom., vol. 4, pp. 3462-3468, 2004
2. S. Smith , M. Schwager and D. Rus "Persistent robotic tasks: Monitoring and sweeping in changing environments" *IEEE Trans. Robot., vol. 28, no. 2, pp. 410-426, 2012*
3. Cassandras, C., Lin, X., & Ding, X. (2012). An Optimal Control Approach to the Persistent Monitoring Problem.
4. Jingjin Yu, M., Schwager, & Rus. (2016). Correlated Orienteering Problem and its Application to Persistent Monitoring Tasks. *IEEE Transactions on Robotics, 32*(5), 1106-1118.
5. Cassandras, C., Xuchao Lin, & Xuchu Ding. (2013). An Optimal Control Approach to the Multi-Agent Persistent Monitoring Problem. *IEEE Transactions on Automatic Control, 58*(4), 947-961.
6. Xuchao Lin, C., & Cassandras. (2015). An Optimal Control Approach to the Multi-Agent Persistent Monitoring Problem in Two-Dimensional Spaces. *IEEE Transactions on Automatic Control, 60*(6), 1659-1664.