An Optimal Control Approach to the Multi-agent Persistent Monitoring Problem [1]

Written by Christos.G. Cassandras, Xuchao Lin and Xu Chu Ding

Reported by Liangyu Chen, Wenxin Zu, Zhiao Zhou

1. Motivation

By recent technology development, autonomous agents can cooperate with each other to accomplish more complex missions. For example, the unmanned vehicles for border patrol missions and some environmental applications that perform area sampling.[2] Specifically, Autonomous vehicles, such as construction machines, operate in hazardous environments, while being required to function at high productivity. To meet both safety and productivity, planning obstacle-avoiding routes in an efficient and effective manner is of primary importance, especially when relying on autonomous vehicles to safely perform their missions. This work explores the use of model checking for the automatic generation of mission plans for autonomous vehicles, which are guaranteed to meet certain functional and extra-functional requirements, e.g., timing ones.[3] In a constant temperature and constant humidity Greenhouse, we need to monitor the temperature and the humidity, while some of them can't reach the setting standard, the monitoring agent will know, then let the corresponding device fix it. Reference [4] presents and experimentally test the framework used by our context-aware, distributed team of small Unmanned Aerial Systems (SUAS) capable of operating in real-time, in an autonomous fashion, and under constrained communications. Their framework relies on three layered approaches: (1) Operational layer, where fast temporal and narrow spatial decisions are made; (2) Tactical Layer, where temporal and spatial decisions are made for a team of agents; and (3) Strategical Layer, where slow temporal and wide spatial decisions are made for the team of agents.

2. Introduction

The main object of this paper is to control the movement of multi-agents to minimize an uncertainty metric in each mission space. For one dimension space, the optimal solution for agents is to move at maximum speed from switching point to others, possibly waiting some time at each point. The solution can be reduced as a parametric optimization problem. A parametric optimization problem is to determine a sequence of switching locations and associated waiting times at these switching points for each agent. We can analyze the hybrid system problem using IPA (Infinitesimal Perturbation Analysis) to obtain the solution through a gradient-based algorithm. [5] Reference [6] presents an algorithm for the complete coverage of free space by a team of mobile robots. But the robots operate under the restriction that communication between two robots is available only when they are within line of sight of each other. Reference [7] proposes gradient descent algorithms for a class of utility functions which encode optimal coverage and sensing policies. The resulting closed-loop behavior is adaptive, distributed, asynchronous, and verifiably correct. Reference [8] propose a receding horizon (RH) controller suitable for dynamic and uncertain environments, where combinatorically complex as-segment algorithms are infeasible. The control scheme dynamically determines vehicle trajectories by solving a sequence of optimization problems over a planning horizon and executing them over a shorter action horizon.

3. Problem description

The uncertainty grows in time if not covered by any agents. Also define the probability of detecting events at each point of the environment by agent sensors. The higher the sensing effectiveness, the uncertainty is reduced faster. We first define the problem as follows: Each agent's optimal trajectory can be described by a set of switch points $\{\theta_1, \ldots, \theta_K\}$, associated waiting times at these points $\{\omega_1, \ldots, \omega_k\}$. The one-dimension space is an interval $[0, L] \in R$. $S_n(t)$is the position of agent *n* at time *t*. $S_n(t) \in [0, L], n = 1, \ldots, N$.

The first constraint is the dynamics of each agent.

$$\dot{S_n}(t) = U_n(t) \tag{1}$$

Assume agents can control its direction and speed. For rescaling size of mission space. Let $|U_n(t)| \leq 1$, $a \leq S(t) \leq b$

$$a \geq 0, b \leq L \tag{2}$$

This constraint let the agents may not reach end points due to obstacles.

$$S_n(t) - S_{n+1}(t) \leq 0 \tag{3}$$

Prevent agents from subsequently crossing each other over all *t*.

Define a function $p_n(x, s_n)$ measures the probability that an event at location x is detected by agent n. $p_n(x, s_n) = 1$ if $x = s_n$. Then denote,

$$p_n(x, s_n) = \begin{cases} 1 - \dfrac{|x - s_n|}{r_n}, & if\ |x - s_n| \leq r_n \\ 0, & if\ |x - s_n| > r_n \end{cases} \tag{4}$$

where r_n is the effectiveness of a sensor.

Define a set of points $\alpha_i, i = 1, \ldots, M, \alpha_i \in [0, L]$. α_i is a time-varying measure of uncertainty with each point. We set $p_n(x, s_n(t)) = p_{n,i}(s_n(t))$

$$P_i(s(t)) = 1 - \prod_{n=1}^{Q} [1 - p_{n,i}(s_n(t))] \tag{5}$$

where $s(t) = [s_1(t), \ldots, s_N(t)]^T$

Denote $R_i(t)$ to be the uncertainty function. The dynamics of $R_i(t), i = 1, \ldots, M$ as follows: A_i is the prespecified increasing rate. B is the decreasing rate of $R_i(t)$. Given that $B > A_i > 0$ (uncertainty strictly decreases when there is perfect sensing $P_i(s(t)) = 1$)

$$\dot{R_i}(t) = \begin{cases} 0, & if\ R_i(t) = 0, A_i \leq BP_i(s(t)) \\ A_i - BP_i(s(t)), & otherwise \end{cases} \tag{6}$$

Our goal now is to control $U_n(t)$ so that cumulative uncertainty over θ_i is minimized over fixed time T. Set $u(t) = [u_1(t), \ldots, u_N(t)]$. Our optimal control problem as following:

$$\min_{u(t)} J = \frac{1}{T}\int_0^T \sum_{i=1}^{M} R_i(t)dt \tag{7}$$

subject to (1), (2), (3), (6), $|U_n(t)| \leq 1$

We characterize the optimal control solution can be reduced to a parametric optimization problem. Due to parametric optimization problem, we can use IPA gradient to find optimal solution. We define the state vector $x(t) = [s_1(t), \ldots, s_N(t), R_1(t), \ldots, R_M(t)]^T$ and costate vector $\lambda(t) = \left[\lambda_{s_1}(t), \ldots, \lambda_{s_N}(t), \lambda_1(t), \ldots, \lambda_M(t)\right]^T$. The Hamiltonian is

$$H(x, \lambda, u) = \sum_{i=1}^{M} R_i(t) + \sum_{n=1}^{N} \lambda_{s_n}(t) u_n(t) + \sum_{i=1}^{M} \lambda_i(t) \dot{R}_i(t) \tag{8}$$

Costate equations:

$$\dot{\lambda}_i(t) = -\frac{\partial H}{\partial R_i(t)} = -1, i = 1, \ldots, M \tag{9}$$

$$\begin{aligned} \dot{\lambda}_{s_n}(t) &= -\frac{\partial H}{\partial S_n(t)} \\ &= -\frac{B}{r_n} \sum_{i \in F_n^-(t)} \lambda_i(t) \prod_{d \neq n} \left[1 - p_{d,i}\big(s_d(t)\big)\right] \\ &\quad + \frac{B}{r_n} \sum_{i \in F_n^+(t)} \lambda_i(t) \prod_{d \neq n} \left[1 - p_{d,i}\big(s_d(t)\big)\right] \end{aligned} \tag{10}$$

with boundary conditions $\lambda_i(T) = 0, i = 1, \ldots, M$ and $\lambda_{s_n}(T) = 0, n = 1, \ldots, N$.

Use PMP to (8), we have $H(x^*, \lambda^*, u^*) = min_{u_n \in [-1,1], n=1,\ldots,N} H(x, \lambda, u)$

The necessary for optimal control:

$$u_n^*(t) = \begin{cases} 1 \ if \ \lambda_{s_n}(t) < 0 \\ -1 \ if \ \lambda_{s_n}(t) > 0 \end{cases} \tag{11}$$

This condition excludes the possibility that $\lambda_{s_n}(t) = 0$ over some finite singular intervals

We can then conclude some propositions:

(i) For any $n = 1, \ldots N, i = 1, \ldots, M, t \in (0, T)$, and any $\varepsilon > 0$, if $s_n(t) = 0, s_n(t - \varepsilon) > 0$, then neither $R_i(\tau) > 0$ for all $\tau \in [t - \varepsilon, t]$ or $R_i(\tau) = 0$ for all $\tau \in [t - \varepsilon, t]$. Under (i), if $a = 0, b = L$, then on an optimal trajectory: $s_n^* \neq 0$ and $s_n^* \neq L$ for all $t \in (0, T), n \in \{1, \ldots, N\}$.

(ii) If $a > 0$ and (or) $b < L$, then on an optimal trajectory there exist finite length intervals $[t_0, t_1]$ such that $s_n(t) = a$ and (or) $s_n(t) = b$, for some $n \in 1, \ldots, N, t \in [t_0, t_1], 0 \leq t_0 < t_1 \leq T$.

(iii) On an optimal trajectory, either $u_n^*(t) = \pm 1$ if $\lambda_{s_n}^*(t) \neq 0$, or $u_n^*(t) = 0$ if $\lambda_{s_n}^*(t) = 0$ for $t \in [0, T], n = 1, \dots, N$.

(iv) If the constraint (3) is included in problem (7), then on an optimal trajectory, $s_n^*(t) \neq s_{n+1}^*(t)$ for $t \in (0, T], n = 1, \dots, N-1$.

4. Methodology

Based on the propositions above, the solution of equation (7) can be characterized by two parameter vectors for each agent $\theta_n = [\theta_{n,1}, \dots, \theta_{n,\Gamma_n}]^T$ and $\omega_n = [\omega_{n,1}, \dots, \omega_{n,\Gamma_n}]^T$. We can now apply the IPA approach.

Previously, this paper showed that the agent moves at full speed on the optimal trajectory, may also stay at the point where time is zero at the switching point, and never reaches any of the boundary points, i.e., $0 < S_n^*(t) < L$. Thus, the $n-th$ agent's movement can be parameterized through $\theta_n = [\theta_{n,1}, \dots, \theta_{n,\Gamma_n}]^T$ and $\omega_n = [\omega_{n,1}, \dots, \omega_{n,\Gamma_n}]^T$.

Further, the solution to equation (7) can be summarized as determining the optimal parameters vectors θ_n^* and ω_n^*, $n = 1, \dots, N$. Therefore, a hybrid system is defined as the optimal behavior of the agent, and the switching position can be regarded as the switching time between specific modes of the system, thus approximating a time optimization problem. The application of IPA and the robustness of the resulting gradients demonstrate that they do not depend on the parameters of the uncertainty model A_i, $i = 1, \dots, M$.

This paper applies IPA to the single-agent case and the multi-agent case, respectively, and the goal is to determine $\nabla J(\theta, \omega)$, which can be obtained from IPA. Then $\nabla J(\theta, \omega)$ is used in the gradient-based algorithm to obtain the optimal parameter vectors θ^* and ω^*,$n = 1, \dots, N$. IPA shows how changes in θ affect state $x(\theta, t)$ and event time $\tau k(\theta)$. The objective function in equation (7) be written as

$$J(\theta, \omega) = \frac{1}{T} \sum_{i=1}^{M} \sum_{k=0}^{K} \int_{\tau_k(\theta)}^{\tau_{k+1}(\theta,\omega)} R_i(t, \theta) dt \tag{12}$$

Proceeding with the gradient descent they got Objective Function Gradient Evaluation

$$\nabla J(\theta, \omega) = \frac{1}{T} \sum_{i=1}^{M} \sum_{k=0}^{K} \int_{\tau_k(\theta)}^{\tau_{k+1}(\theta,\omega)} \nabla R_i(t) dt \tag{13}$$

which depends on $\nabla R_i(t)$.

Finally, use IPA-based optimization algorithm to find θ^* and ω^*.

5. Simulation result

We break the results into two parts, the first part is to compare the difference between the paper and our simulation, and the second part is some extensions we found that are interesting to develop. The following figures are the paper result and our simulation result. We apply the same parameters shown in figure (1).

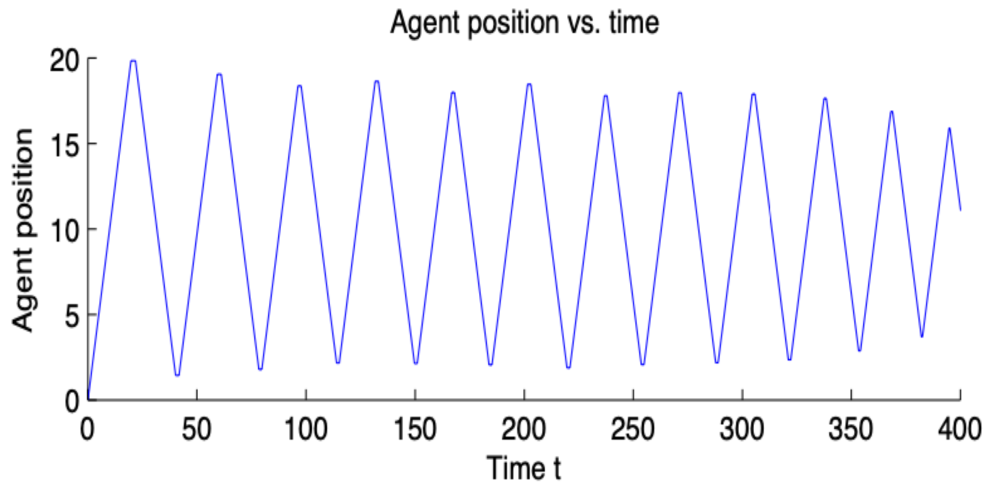


Figure (1): Position of single agent

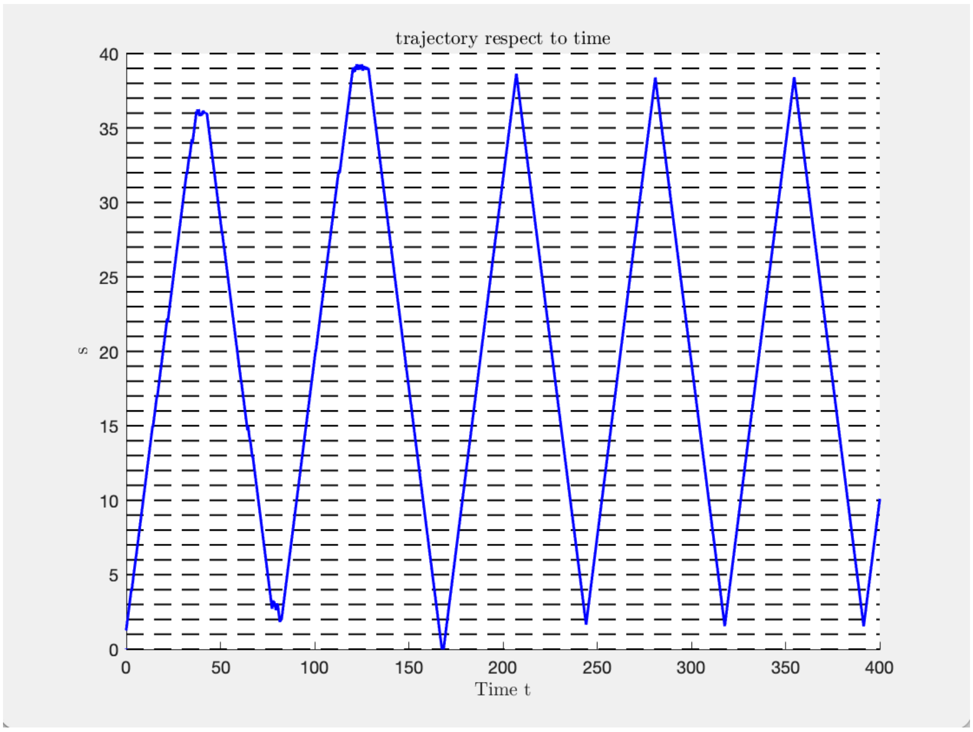
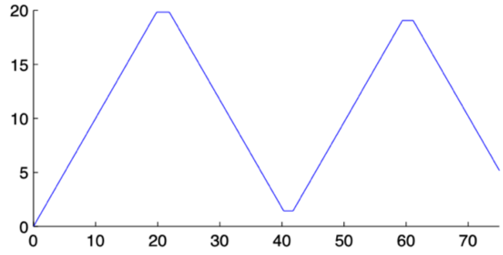


Figure (2): Simulation result

We found that the agent's position peak decreased as time increased in figure (1), which is quite surprising because the uncertainty increasing and decreasing rate is a constant in this case, the agent should move back and forth periodically. Except for that, our simulation quite matches the paper result. And if we zoom in the figure, we can actually see the waiting time as the paper mentioned shows in figure (3) and figure (4).

$$a = 0, b = 20. A_i = 0.1, i = 0, \dots, 20. J^* = 17.77$$



(a) $a = 0, b = 20.$

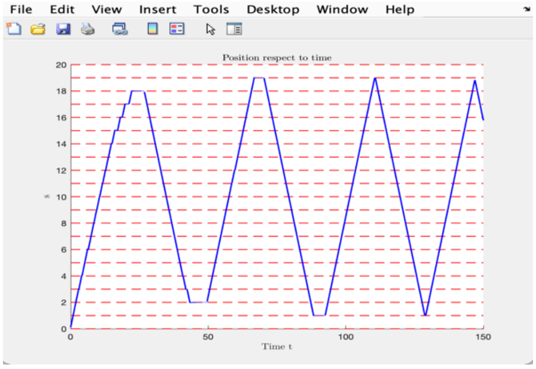
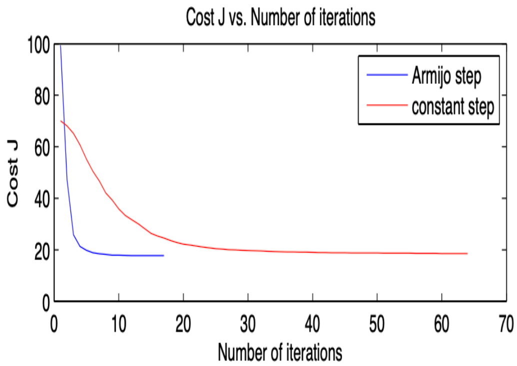


Figure (3): Waiting time

Figure (4) : Simulation result

In theory, the waiting point should be located at the end of the boundary. However, our simulation shows that the waiting time is close to the end boundary, not exactly located at the boundary. We discussed the result and concluded that the sensing range of the agent could cover the end of the boundary and reduce the uncertainty to the setting standard without reaching the end of the boundary. Yet, we still showed the existence of waiting time.

Then we compared the cost function in figure (5) and figure (6).



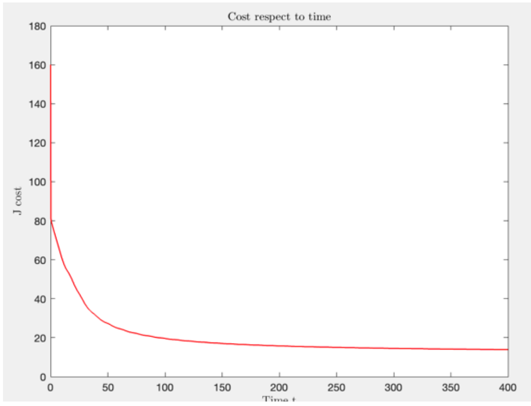


Figure (5): Cost respect to time

Figure (6): Simulation result

In the reference paper, they showed the cost with respect to iteration. Our x axis is respected to time because we didn't ensure how they defined one iteration. Besides, we didn't apply Armijo step, instead, we tried to use different constant step size to get optimal cost close to the reference paper which is J* = 17.77. The step we picked that is closest to the optimal J is 0.0051 and the corresponding J = 18.03.

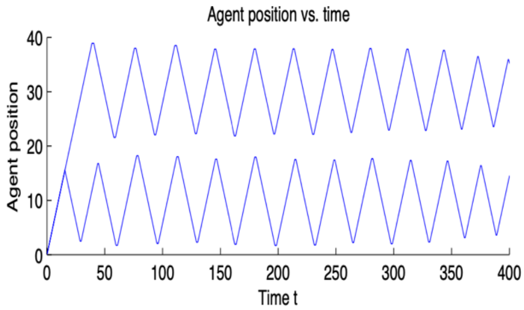


Figure (7): Position of double agents

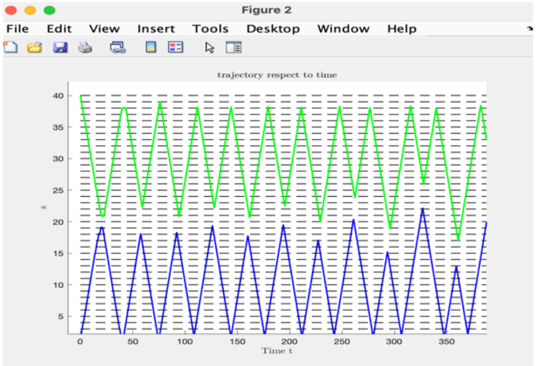


Figure (8): Simulation result

It is obvious to observe that the initial positions of the agents are different. The reason we separated the two agents is due to constraint (3). During the problem formulating process, the constraint has already claimed that the agents' position should not overlap with each other. The rest trajectories are similar to the paper.

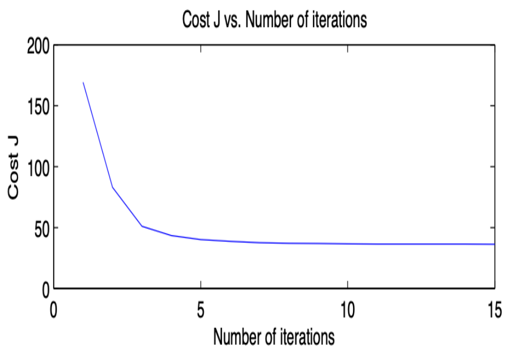


Figure (9): Cost of double agents

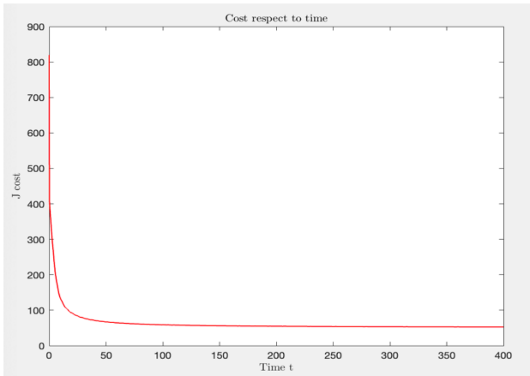
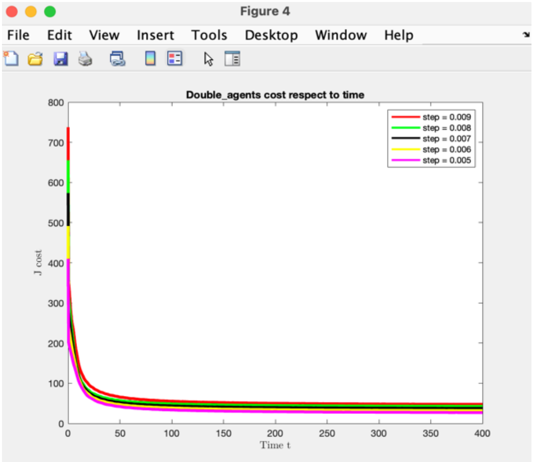


Figure (10): Simulation result

The accumulated uncertainty in our simulation J is = 42.65. However, due to the typo, the paper didn't demonstrate their actual cost. Their J for double agents in L = 40 is 17.77, which is exactly the same as the single agent case in L = 20. Moreover, from figure (9), the cost J is roughly lower than 50. As mentioned in the single agent case, we didn't know the definition for one iteration in the paper, so we couldn't compare the coverage rate of the cost. But it seems to converge at a similar position. Except for simulating the case in paper, we did some extension to analyze how step size affects the cost. As learned from the course SE524, while applying gradient descent method, big step size may not find the optimal value, but small step size consumes too much computational cost. The following figures compared different sizes of steps. We classified into two classes, small scale step size and large-scale step size with respect to different numbers of agents.



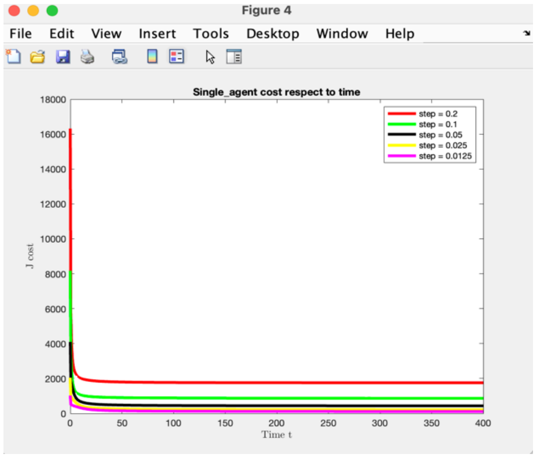
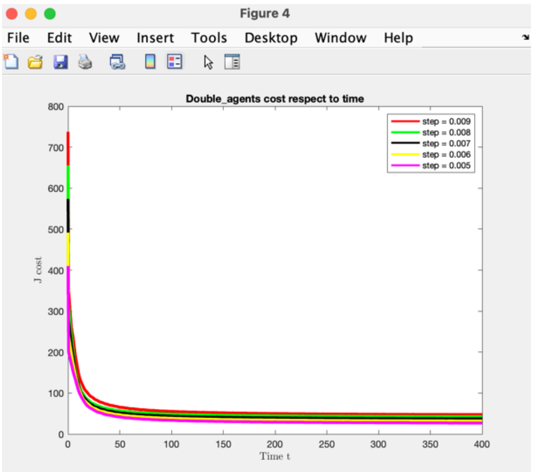


Figure (11): Large scale step size

Figure (12): Small scale step size

It can be observed that large step sizes converge much faster than small step sizes, but the converged value J is also bigger. With large step sizes, the convergence time is at about 50 units, while small step sizes converge mostly after 150 units with respect to the time axis.



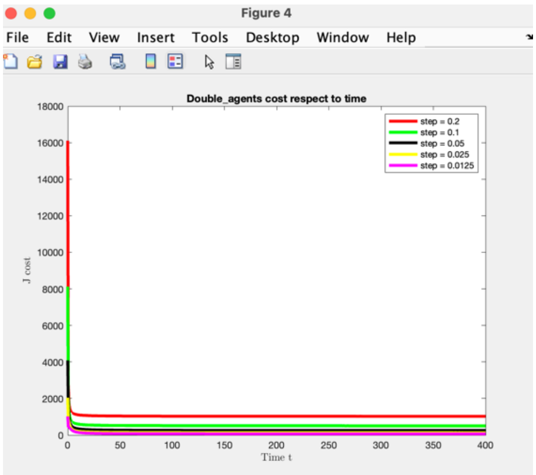


Figure (13): Large scale step size

Figure (14): Small scale step size

Then, we compared single and double agents with respect to same time units. As expected, the cost of double agents is lower than the single case because within the fixed mission space, each agent takes less responsibility to reduce the uncertainty. The uncertainty has less time to grow highly in double agent's case.

Another extension point we made is to have three agents evenly distributed in the mission space to observe the result. Figure (15) and table (1) shows the position and cost.

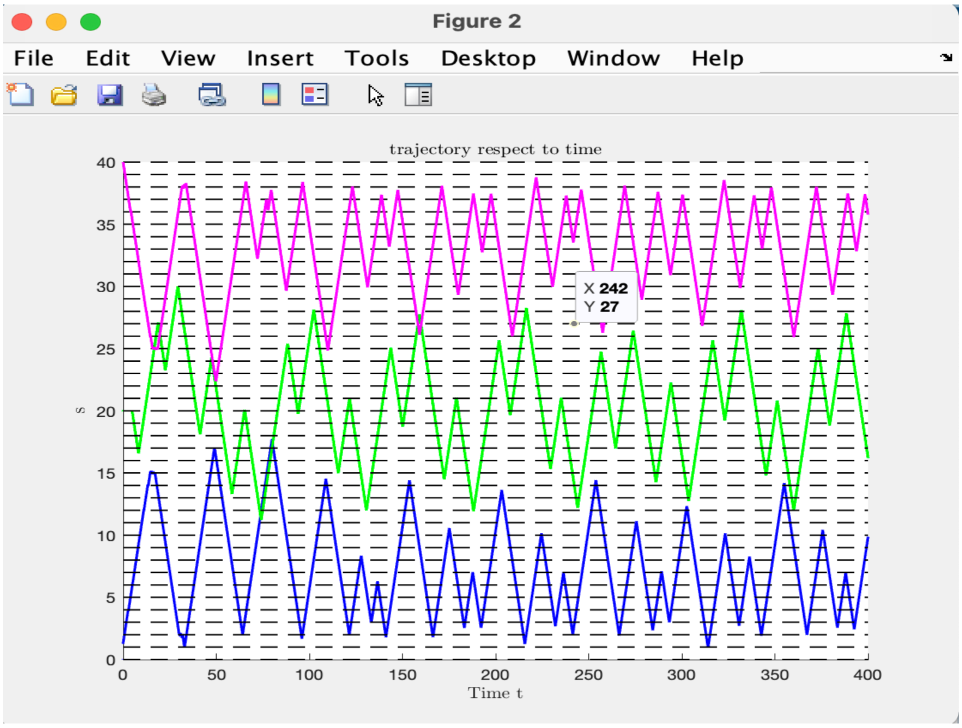


Figure (15): Three agents position with respect to time

Number of agents	Mission space	Sensing range(r)	Initial increasing rate (Ai)	Decreasing rate(B)	T	dt	Cost
1	40	2	0.1	3	400	0.05	432.84
2	40	2	0.1	3	400	0.05	253.20
3	40	2	0.1	3	400	0.05	176.08

Table (1): The cost with respect to time

We found that sometimes the agents' trajectories would cross each other even though we have already imposed the constraint to prevent them from overlapping. We still try to figure out what went wrong and fix it. We look at the cost with different numbers of agents, and the result is quite reasonable due to the same reason as the previous case.

6. Conclusion & Future work

After we went through the project, we started to think about what an optimal control for this kind of problem is. It depends on the application itself. For example, to control multi-agent to rescue the disaster scene, we expect the agents to make decisions as soon as possible because of the emergency. However, once the scenario changes to greenhouse irrigation, we now don't concern about the computational cost and time. Instead, we care more about the accuracy of the humidity control due to the vulnerability of the plants. Other concerns are what constraints are needed if we extend to higher dimensions and deal with the disturbances in reality.

Reference

[1] C. G. Cassandras, X. Lin and X. Ding, "An Optimal Control Approach to the Multi-Agent Persistent Monitoring Problem," in IEEE Transactions on Automatic Control, vol. 58, no. 4, pp. 947-961, April 2013, doi: 10.1109/TAC.2012.2225539.

[2] Cassandras, Christos G. et al. “An optimal control approach for the persistent monitoring problem.” *IEEE Conference on Decision and Control and European Control Conference* (2011): 2907-2912.

[3] Gu, R., Enoiu, E. and Seceleanu, C., 2020, March. TAMAA: UPPAAL-based mission planning for autonomous agents. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing* (pp. 1624-1633).

[4] Martinez-Lorenzo, J., Hudack, J., Jing, Y., Shaham, M., Liang, Z., Bashit, A.A., Wu, Y., Zhang, W., Skopin, M., Heredia-Juesas, J. and Ma, Y., 2021. Preliminary Experimental Results of Context-Aware Teams of Multiple Autonomous Agents Operating under Constrained Communications. *arXiv preprint arXiv:2103.14123*.

[5] Cassandras, Christos G. et al. “An optimal control approach for the persistent monitoring problem.” *IEEE Conference on Decision and Control and European Control Conference* (2011): 2907-2912.

[6] I. Rekleitis, V. Lee-Shue, A. New, and H. Choset, “Limited communication, multi-robot team-based coverage,” in Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on, vol. 4. IEEE, 2004, pp. 3462–3468.

[7] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” Robotics and Automation, IEEE Transactions on, vol. 20, no. 2, pp. 243–255, 2004.

[8] W. Li and C. Cassandras, “A cooperative receding horizon controller for multivehicle uncertain environments,” IEEE Transactions on Automatic Control, vol. 51, no. 2, pp. 242–257, 2006.