

Statement on the Revisions Based on the Reviewers' Comments

Ian Abraham

Todd D. Murphey

We thank the reviewer and editor for processing our paper. Below are the responses to the comments made to our submitted paper.

Comments by Editor

The paper addresses the problem of coordinate agents in a shared environment for coverage purposes in a distributed framework. The proposed method is based on the ergodic control that with consensus based algorithms can be distributed among agents. Simulations for a scenario with obstacles is proposed for three quadrotors.

Before publication authors should address several issues such as those reported by Reviewer 2 and others reported below mainly due to the control flavour of the paper and the lack of realistic aspects. The paper needs some improvement in clarity, formalism, and from the applicability point of view:

We thank the editor and reviewers for their comments and suggestions for the paper. We address the main comments in the following way:

- We rewrote Introduction (Section I) with further motivation and comparison with the references provided by Reviewer 2.
- We added a problem statement (Section II) to address the lack of formalism and clarify a problem we are solving within the context of robotics.
- We added Definitions 1 and 2 to clarify the distinction between the terms distributed and decentralized
- We added a comparison with the work in [17] in Figure 3 of an area coverage example in a closed corridor illustrating that our method can directly control the agents with nonlinear dynamics and compensate for small visibility. We also add an example in the multimedia video of a moving target localization illustrating that our method can also handle time-varying distributions.
- We removed the appendix for space constraints and refer to existing citations

- Actual scalability due to the required stored memory.

A similar point has been brought up by reviewer 2. Since the stored memory is only necessary for calculating the history of the single robot (and not the system of agents), the amount of memory required does not change at all with respect to the number of robots in the system. Moreover, the only information that is communicated is the c_k Fourier coefficient terms of the single robot agent to its neighboring agents in the network. Thus, increasing the number coefficients used for the spatial statistic decomposition (and the robot trajectory) would increase the required data that needed to be communicated to each agent. However, increasing the number of agents in the system does not increase the amount of data (from computing the c_k coefficients) that is needed to be communicated.

We regret that this was not clearly explained in the paper and in response have added Section III-A paragraph 2 which explains the role of the stored memory to the algorithm. In addition, we added Section III-D which discusses the required data needed in storage for the individual agent and what is required of an agent to communicate to their respected adjacent agents.

- Comparison with other methods in the literature

We thank the editor and reviewer 2 for providing us with a list of recent work that is related and relevant to our work. In response to this comment, we updated Section I with an update literature review and motivation for our method in relation to the existing literature. In addition, we added a comparison in Figure 3 with the example used in [17] .

We further explain both the differences in our approach to the problem of area coverage and how our work is similar in ideas (particularly the metrics used) with the prior work. We also include more examples in the attached multimedia for time-varying distributions, that emphasize the contribution of this paper.

-Robustness evaluation with respect to uncertainties and estimation errors

While this is a topic we want to discuss in later work, the filter method (and estimation method) as well as the sensor noise and parameters specified will affect any statement about robustness, but the ergodic control methodology is independent of those choices. We do, however point out that we can guarantee stability with respect to the expected information density, and that notion of stability could be generalized to a robust control framework.

We add a brief mention of the filtering method and the sensor parameters in Section VI-B paragraph 1 and Figure 1 (c) in response to this comment.

Comments by Reviewer #2

The submitted paper proposes an ergodic control scheme to provide decentralized time-varying coverage using multiple robots with nonlinear dynamics. The paper is interesting but there are several major issues that the authors should address. In a nutshell, (i) the paper is poorly written and some important parts of the proposed method are not readable (ii) the contribution is unclear.

We thank the reviewer for their comments and hope that the revisions based on the comments made have improved the readability and contribution clear.

The addressed problem is very similar to area coverage problems that have been extensively studied in the literature. Specifically, see the following works for time-independent target distributions. Similarly, target localization is another well studied topic. These methods are online, distributed, do not require to store the history of the robot positions, and scale well with the size for the network. How is the proposed method compared to such works? How is the proposed method compared to the above methods? Why and when should someone use the proposed ergodic control framework and not one of the existing coverage and localization methods?

We thank the the reviewer for the list of relevant papers in the literature. From reading through the literature, one of the biggest differences is that each of these methods cannot be used to solve the other methods. In contrast, our method is general enough to be specified for the mentioned problems. We illustrate this in our examples for terrain estimation, target localization, area coverage in corridor, and moving target localization with obstacles.

Another difference is the use of Voronoi segmentation and the additional specification of control metrics when the dynamics becomes nonlinear. Moreover, much of the referenced work involves placement of the agents rather than persistent movement in area coverage. Our work uses hybrid systems theory to specify controllers for agents with nonlinear dynamics for area coverage under the ergodic metric. In using the ergodic metric, we enforce that the agents have persistent area coverage. Thus, the agents are always in motion and reactive to time-varying target distributions which are general to a variety of problems that are found in multi-agent coordination.

In response to this comment, we have added to the Introduction in Section I which compares the literature provided by the reviewer and additional references which address nonlinear agent dynamics.

There is no clear problem formulation. Instead in section II, the problem formulation is mixed with the proposed solution. This section needs to be re-organized.

We thank the reviewer for this observation. We have split Section II into Section II and III, with Section II providing a dedicated problem statement for the paper.

What is ‘‘q’’ in (3)?

Here, ‘‘q’’ is a weight on the ergodic metric, similar to a weight on a quadratic objective function. We provide a definition for ‘‘q’’ in the text as the weight on the ergodic metric right after equation (3).

What is λ ? Is it a small time interval? How does it appear in (3)? This is important to understand Proposition 1. The same also holds for τ_* that suddenly appears later in the text.

The value λ is a time duration for which the control is applied. During the calculation of the mode insertion gradient, we treat this as an infinitesimally small time interval for which we measure how sensitive the objective is to a small time perturbation of a control u_* . In application, λ is coupled with a time of application τ . While one can simply choose any time $\tau \in [t_i, t_i + T]$, we typically search for the time that has the most negative mode insertion gradient. We define this point as τ_* after solving the arg min of the mode insertion gradient.

We modified Section III-B paragraphs 2 and 5 in order to include this information that was not previously defined.

How is $\Delta t_{\mathcal{E}}$ selected? Is it time dependent? More details are required.

Any time parameter that is positive can be chosen. The parameter $\Delta t_{\mathcal{E}}$ is typically chosen to be $2 * T$ or twice the time horizon in practice. This parameter can even be time dependent if required. For many other applications, $\Delta t_{\mathcal{E}}$ can be chosen such that the desired behavior of the robot is tuned to a user’s preference. Specifically, having a small $\Delta t_{\mathcal{E}}$ would result in myopic behavior where the robot is likely to revisit regions it has already been because the past memory, dictated by $\Delta t_{\mathcal{E}}$, is small. The reverse is true if $\Delta t_{\mathcal{E}}$ is significantly large. The robot is then less likely to visit regions that it has visited more frequently.

We provide a discussion for choosing $\Delta t_{\mathcal{E}}$ in Section III the last paragraph.

Also, since the history of each robot needs to be stored, this means that the proposed algorithm scales poorly with the size of the network. This needs to be discussed and simulation studies that show how the proposed method performs as the number of robots increases need to be provided.

Since the algorithm is decentralized, the memory of the past trajectory of a single robot is only required to be stored on the single robot agent itself. As a result, given a fixed $\Delta t_{\mathcal{E}}$ for the j^{th} robot's state history $x_j(t) \forall t \in [t_i - \Delta t_{\mathcal{E}}, t_i]$, the memory required remains constant for each agent. Specifically, given a 64 bit memory convention, the memory required is $64 * \Delta t_{\mathcal{E}} / t_s * n$ bits where t_s is the sampling rate and n is the size of the state vector. Since the only information that is shared within the network is the c_k trajectory Fourier coefficients of a single agent to their adjacent neighbors, the only required data that is needed to be sent is the trajectory c_k . Since the number of c_k coefficients is user defined and finite, the memory required to communicate for a single agent is $64 * |k| * \text{number of neighbors}$ bits where $|k|$ is the maximum number of coefficients. If the number of neighbors is fixed then regardless of the number of robots that are in the network, the amount of bits needed to communicate remains the same. Thus, the proposed decentralized algorithm does not scale at all with the size of the network.

We address how the algorithm scales in Section III-D and provide a similar analysis as presented in the response.

If the proof of Proposition 1 exists in [11] why is it included in the Appendix?
Is there any difference?

The proof is simply there so that the reader does not have to reference in the paper in [11]. For space constraints, we have removed the proof to Proposition 1 and just mentioned [11].

Also, equation (6) and the text before it are very vague. What does it capture?
Why is it important?

Equation (6) is an objective that is used to reframe the constrained objective in (4) as a unconstrained objective with a minimizer that has a closed form solution. Without this formalisms we would not be able to construct the distributed controller.

We address this comment by adding a description of (6) in Section III-B paragraph 3 that reflects the purpose and importance of equation (6).

Based on Fig 2, it is unclear why the decentralized method should be used instead of the centralized one. What are the advantages of the decentralized method compared to the centralized one? This is very important to answer to clarify the contribution of the paper.

We thank the reviewer for making this important observation. Figure 2 is provided as comparison of the decentralized and centralized version of the algorithm. This allows us to compare how effective the decentralized algorithm is with respect to the centralized algorithm for the same problem. The advantage of the decentralized algorithm, as opposed to the centralized algorithm, is the ability to solve large scale multi-agent coordination problems without having to compute an action for the large scale system. Instead, using a decentralized framework allows local individual-level coordination problems to be solved by sharing information with neighboring agents and the resulting outcome is solving the large scale coordination much more quickly without worrying about the scale of the problem as the multi-agent network is large.

We have rewritten the Introduction and related work (Section I) in order to best reflect this contribution of our method.

I would suggest Section V appear right after Section II but the authors do need to follow this suggestion.

Thank you for the suggestion. We have made this change.

There are several typos in the paper, such as:

We thank the reviewer for taking the time to look for typos. We hope we fixed all the typos.

Comments by Reviewer #4

The authors present a decentralized formulation of ergodic control for multi-agent systems. The paper extends some previous results resulting in a distributable formulation emptying consensus. The authors provide also a game theoretical analysis of the algorithm and show the best possible control policy for an agent in a multi-agent game.

The paper is well-written and the results are presented well.

We thank the reviewer for their comment!

The paper needs some proofreading for some minor typos or and editorial polishing

We thank the reviewer for taking the time to find some of these typos and hope that we have done a good job at fixing them.