Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling (ICAPS 2017)

Multi-Agent Ergodic Coverage with Obstacle Avoidance

Hadi Salman, Elif Ayvali, Howie Choset
{hadis@andrew, elifa@andrew, choset@cs}.cmu.edu
Robotics Institute at Carnegie Mellon University
Pittsburgh, PA 15213, USA.

Abstract

Autonomous exploration and search have important applications in robotics. One interesting application is cooperative control of mobile robotic/sensor networks to achieve uniform coverage of a domain. Ergodic coverage is one solution for this problem in which control laws for the agents are derived so that the agents uniformly cover a target area while maintaining coordination with each other. Prior approaches have assumed the target regions contain no obstacles. In this work, we tackle the problem of static and dynamic obstacle avoidance while maintaining an ergodic coverage goal. We pursue a vector-field-based obstacle avoidance approach and define control laws for idealized kinematic and dynamic systems that avoid static and dynamic obstacles while maintaining ergodicity. We demonstrate this obstacle avoidance methodology via numerical simulation and show how ergodicity is maintained.

Keywords— Multi-agent planning, centralized robot control, ergodic theory, uniform coverage, obstacle avoidance.

1 INTRODUCTION

Ergodic theory is the statistical study of time-averaged behavior of dynamical systems (Petersen 1989). An agent exhibits ergodic dynamics if it uniformly explores all of its possible states over time. This useful notion has been previously used to distribute multiple-agents in an exploration domain such that agents' trajectories uniformly cover the domain[1]. Formally, ergodicity is measured with respect to a given target probability distribution. For example, to visit all the possible states in the exploration domain, one needs to specify the target probability distribution as a uniform distribution.

What makes ergodic coverage interesting is that it overcomes the drawbacks of alternative uniform coverage algorithms, such as the lawnmower algorithm in which agents scan an area by going back and forth in parallel lines (Hubenko et al. 2011); ergodic coverage can be easily implemented for irregular domains and non-uniform target probability distributions. Also, it assumes perfect communication between agents. Agents' motions are planned sequentially by taking into account the motion of every other agent. Therefore, it is flexible to agent failure and to adding new agents. Furthermore, ergodic coverage is naturally multi-scale i.e., covers large scale regions followed by smaller regions (Mathew and Mezić 2011).

Methods that use ergodicity to drive multi-agents do not consider obstacles (Mathew and Mezic 2009; Miller and Murphey 2013a; 2013b). To exclude the coverage of particular regions, Mathew and Mezić (2011) set the target probability distribution to zero in those regions. Although this method often steers the agents away from the regions they should exclude, it does not guarantee that agents fully avoid them. Therefore, such an approach cannot be used for obstacle avoidance.

Our goal in this paper is to design control laws for multiple agents such that they achieve uniform ergodic coverage of a domain occluded with obstacles. To the best of our knowledge, no one has tackled the problem of uniform ergodic coverage of a domain while avoiding obstacles.

This paper is organized as follows. In Section 2, we give background on ergodic coverage and vector-field-based obstacle avoidance frameworks (Panagou 2014). In Section 3, we tackle the above problem by redefining the optimal feedback law derived by Mathew and Mezić (2011) for first order and second order dynamical systems such that static and dynamic obstacles are avoided. We show that the ergodic metric is driven to zero, hence, ergodicity is maintained. In Section 4, we validate our algorithms via simulation with tuned parameters. Finally, Section 5 concludes by stating the importance of the presented method and paves the way for related future work.

2 Background

A. Ergodic Coverage

In a multi-agent system with N agents, the spatial time average statistics of an agent's trajectory $\gamma_j : [0,t] \rightarrow X$, quantifies the fraction of the time spent at a position $\mathbf{x} \in X$, where $X \subset \mathbb{R}^n$ is an n-dimensional coverage domain defined by $[0,L_1] \times ... \times [0,L_n]$, and $j \in \{1,2,..N\}$ is the index of an agent. The time-averaged statistics of the agents' trajecto-

[1]The notion of 'uniform coverage' implies that points on the agent trajectories must be as uniformly distributed or evenly spaced throughout the domain (Mathew and Mezić 2011).

ries is defined as

$$C^t(\mathbf{x}) = \frac{1}{Nt}\sum_{j=1}^{N}\int_0^t \delta(\mathbf{x}-\gamma_j(\tau))d\tau \tag{1}$$

where δ is the Dirac delta function.

Ergodicity is measured by comparing the spectral decompositions of the spatial distribution of agents' trajectories and a given probability distribution $\mu(\mathbf{x})$ defined over the coverage domain. The metric of ergodicity, $\phi(t)$, quantifies the difference between the time-average statistics of a trajectory and the probability distribution $\mu(\mathbf{x})$, and is defined as

$$\phi(t) = \sum_{K=0}^{m}\Lambda_k |s_k(t)|^2, \tag{2}$$

where m is the number of Fourier basis functions (we use $m = 50$) and

$$s_k(t) = c_k(t) - \mu_k, \quad \Lambda_k = \frac{1}{(1+\|k\|^2)^{\frac{n+1}{2}}} \tag{3}$$

where $c_k(t)$ and μ_k are the Fourier coefficients of the distributions $C^t(\mathbf{x})$ and $\mu(\mathbf{x})$ respectively, i.e.,

$$c_k(t) = \langle C^t, f_k\rangle = \frac{1}{Nt}\sum_{j=1}^{N}\int_0^t f_k(\gamma_j(\tau))d\tau \tag{4}$$

$$\mu_k = \langle \mu, f_k\rangle = \int_X \mu(\mathbf{x})f_k(\mathbf{x})d\mathbf{x} \tag{5}$$

with $f_k = \frac{1}{h_k}\prod_{i=1}^{n}\cos(\frac{k_i\pi}{L_i}x_i)$ being the Fourier basis functions that satisfy Neumann boundary conditions on the domain X, $k \in \mathbb{Z}^n$ is the wave-number vector, and $\langle\cdot,\cdot\rangle$ is the inner product with respect to the Lebesgue measure. The term h_k is a normalizing factor. The metric of ergodicity $\phi(t)$, defined in (2), quantifies the deviation of the time averages of the Fourier basis functions from their spatial averages, and gives more weight to large-scale (low frequency) modes than small-scale (high frequency) modes because of the term Λ_k.

The goal of ergodic coverage is to generate optimal controls $\mathbf{u}^{opt}(t)$ for an agent, whose dynamics are described by a function $g\colon Q\times U \to TQ$, such that

$$\begin{aligned} \mathbf{u}^{opt}(t) &= \arg\min_{\mathbf{u}} \phi(t), \\ \text{subject to } \dot{\mathbf{x}} &= g(\mathbf{x}(t),\mathbf{u}(t)), \\ \|\mathbf{u}(t)\|_2 &\le u_{max} \end{aligned} \tag{6}$$

where $\mathbf{x}\in Q$ is the state of the agent and $\mathbf{u}\in U$ denotes the set of controls. Equation (6) can be solved using sequential quadratic programming (Nocedal and Wright 2006) or trajectory optimization (Miller and Murphey 2013b) for systems with nonlinear dynamics, and can be solved for systems with simple first-order or second-order dynamics using greedy optimization methods (Mathew and Mezić 2011). Similar to Mathew and Mezić (2011), we minimize the metric of ergodicity at the end of a short time horizon. We use this approach because of the algorithm's online nature, ease of implementation, and good empirical performance.

First Order Dynamics We first consider an agent j with first order dynamics of the form,

$$\dot{\mathbf{x}}_j(t) = \mathbf{u}_j(t). \tag{7}$$

The objective is to design feedback laws $\mathbf{u}_j(t) = \mathbf{F}_j(\mathbf{x}_j)$ so that the agents have ergodic dynamics, i.e., the metric of ergodicity $\phi(t)$ is driven to zero. We adopt the notations used in Mathew and Mezić (2011) and define the following auxiliary variables, which are used in the derivation of the optimal feedback law,

$$S_k(t) := Nts_k(t) \tag{8}$$

$$\Phi(t) := \frac{1}{2}\sum_k \Lambda_k |S_k(t)|^2 = \frac{1}{2}N^2t^2\phi(t) \tag{9}$$

$$W_k(t) := \dot{S}_k(t) = \sum_{j=1}^{N} f_k(\gamma_j(t)) - N\mu_k \tag{10}$$

$$\dot{W}_k(t) = \sum_{j=1}^{N}\nabla f_k(\gamma_j(t))\,.\,\mathbf{u}_j(t) \tag{11}$$

where ∇f_k is the gradient vector field of the Fourier basis functions f_k.

The problem of choosing $\mathbf{u}_j(t)$ at time t, such that it minimizes $\phi(t+dt)$ for small dt, is the same as that of minimizing $\Phi(t+dt)$ by referring to (9). Using Taylor series expansion of $\Phi(t)$ around t, we get

$$\Phi(t+dt) \approx \Phi(t) + \dot{\Phi}(t)dt + \frac{1}{2}\ddot{\Phi}(t)dt^2 =: \tilde{\Phi}(t,dt). \tag{12}$$

The first time-derivative of $\Phi(t)$ is given by

$$\dot{\Phi}(t) = \sum_K \Lambda_k S_k(t)W_k(t). \tag{13}$$

Note that since the cost function $\Phi(t)$ involves time-integrals of function of the agents' positions, the current values of the controls $\mathbf{u}_j(t)$ do not directly affect the current value of the first derivative $\dot{\Phi}(t)$. This can be seen by referring to equations (8), (10), and (13) [2].

The second time-derivative of $\Phi(t)$ is given by

$$\ddot{\Phi}(t) = \sum_K \Lambda_k (W_k(t))^2 + \sum_K \Lambda_k S_k(t)\dot{W}_k(t) \tag{14}$$

Substituting equation (11) in equation (14), we get

$$\begin{aligned} \ddot{\Phi}(t) &= \sum_K \Lambda_k (W_k(t))^2 + \sum_K \Lambda_k S_k(t)\left[\sum_{j=1}^{N}\nabla f_k(\gamma_j(t))\,.\,\mathbf{u}_j(t)\right] \\ &= \sum_K \Lambda_k (W_k(t))^2 + \sum_{j=1}^{N}\mathbf{B}_j(t)\,.\,\mathbf{u}_j(t) \end{aligned} \tag{15}$$

where

$$\mathbf{B}_j(t) = \sum_K \Lambda_k S_k(t)\nabla f_k(\gamma_j(t)) \in \mathbb{R}^2 \tag{16}$$

[2] The controls at the current time affect the values of the first time-derivative $\dot{\Phi}(t)$ at a later time (since they affect positions of the agents).

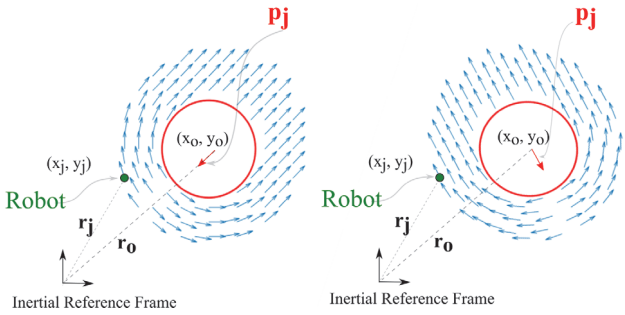


Figure 1: Two examples of a repulsive vector field (blue vectors) around the obstacles (red circles). This vector field is oriented depending on a vector $\mathbf{p}_j$, which is different in each of these examples. The shape of the vector field is constant in this paper, and its orientation depends solely on $\mathbf{p}_j$.

Since the first term in (15) is a quadratic term that is always greater than or equal to 0, the choice of $\mathbf{u}_j(t)$ that minimizes $\tilde{\Phi}(t,dt)$ subject to the constraint $\|\mathbf{u}_j(t)\|_2 \leq u_{max}$ is,

$$\mathbf{u}_j^{opt}(t) = -u_{max}\frac{\mathbf{B}_j(t)}{\|\mathbf{B}_j(t)\|_2} \tag{17}$$

where u_{max} is the maximum speed of the agents.

We define

$$\mathbf{V}_j^{opt}(t) = -\frac{\mathbf{B_j}(t)}{\|\mathbf{B_j}(t)\|_2} \tag{18}$$

to be the direction of velocity vector of agent j at time t (we will use this term later in Section 3). This is the direction of steepest descent that minimizes $\Phi(t)$ at each time step. Thus, the optimal feedback law given by (17) drives the agent at each time step along this steepest descent direction.

Second Order Dynamics Now we consider an agent j with second order dynamics of the form,

$$\ddot{\mathbf{x}}_j(t) = \mathbf{u}_j(t). \tag{19}$$

This can be written in state space as,

$$\begin{cases} \dot{\mathbf{x}}_j(t) = \mathbf{v}_j(t) \\ \dot{\mathbf{v}}_j(t) = \mathbf{u}_j(t) \end{cases}. \tag{20}$$

In a derivation similar to that obtained for first order dynamics, the optimal feedback law that ensures ergodic dynamics for this second order system is given by (Mathew and Mezić 2011) as,

$$\mathbf{u}_j^{opt}(t) = -F_{max}\frac{c\mathbf{v}_j(t)+\mathbf{B}_j(t)}{\|c\mathbf{v}_j(t)+\mathbf{B}_j(t)\|_2} \tag{21}$$

where $\mathbf{B}_j(t)$ is defined in (16), c is a parameter representing how much damping the agent has, and F_{max} is the maximum acceleration of the agent.

B. Vector Field Based Obstacle Avoidance

In the presence of obstacles, repulsive vector fields (Panagou 2014) are one solution to avoid obstacles by creating an artificial vector field in the vicinity of an obstacle to drive the robot around the obstacle and prevent it from collision.

Panagou (2014) defines the following family of vector fields,

$$\mathbf{F}(\mathbf{r}) = \lambda(\mathbf{p}^T\mathbf{r})\mathbf{r} - \mathbf{p}(\mathbf{r}^T\mathbf{r}) \tag{22}$$

where $\lambda \in \mathbb{R}$ is a parameter that specifies the "shape" of the vector field, $\mathbf{p} \in \mathbb{R}^2$ specifies the "overall orientation" of the vector field, and $\mathbf{r} = \mathbf{r}_j - \mathbf{r}_o$, where $\mathbf{r}_j = (x_j, y_j)$ is the position of the agent, and $\mathbf{r}_o = (x_o, y_o)$ is the position of the obstacle with respect to an inertial reference frame.

3 Ergodic Coverage with Obstacles

In this section, we present our derivation of feedback laws for first order and second order dynamical systems of the form given in (7) and (19) respectively, that avoid obstacles while maintaining ergodicity. In this paper, we assume that the obstacles are all circular in shape. Without loss of generality, we assume there is only one obstacle in the domain of coverage.

The repulsive vector field that we use around an obstacle is picked from the family of vector fields (22) as follows,

$$\mathbf{F}_j^o(\mathbf{r}_j) = \begin{cases} \mathbf{F}\big|_{\lambda=1}(\mathbf{r}_j - \mathbf{r}_o) & \text{if } \mathbf{p}_j^T(\mathbf{r}_j - \mathbf{r}_o) \geq 0 \\ \mathbf{F}\big|_{\lambda=0}(\mathbf{r}_j - \mathbf{r}_o) & \text{if } \mathbf{p}_j^T(\mathbf{r}_j - \mathbf{r}_o) < 0 \end{cases} \tag{23}$$

An example of this vector field is shown in Fig. 1 for two different choices of $\mathbf{p}_j$[3], which shows the effect of the latter on the orientation of the vector field.

A. First Order Systems

For first order systems of the form given in equation (7), we define the feedback law that drives an agent j, while avoiding collisions and maintaining ergodicity, to be,

$$\mathbf{u}_j^*(t) := u_{max}\frac{\mathbf{V}_j^*(t)}{\|\mathbf{V}_j^*(t)\|_2}. \tag{24}$$

where $\mathbf{V}_j^*(t) \in \mathbb{R}^2$ is defined as,

$$\mathbf{V}_j^*(t) := \alpha\mathbf{V}_j^{opt}(t) + (1-\alpha)\mathbf{F}_j^o(\mathbf{r}_j) \tag{25}$$

where $\mathbf{V}_j^{opt}(t)$ and $\mathbf{F}_j^o(\mathbf{r}_j)$ are defined in (18) and (23) respectively. Furthermore, $\alpha \in [0,1]$ is a bump function; it takes the value of 1 far from the obstacle and decreases as the agent approaches the obstacle till it reaches 0 at the boundary of the obstacle (or at a safety offset from the boundary of the obstacle). The parameter α can be defined in various ways (linear, quadratic, cubic, (Panagou 2014), or exponentially-decaying (Lindemann and LaValle 2007) bump functions). In our implementation, we use a linear bump function.

Claim 1 *When an agent is near an obstacle, $\mathbf{u}_j^*(t)$, given by equation (25), minimizes $\Phi(t)$, given by equation (9), along a descent direction at each time step. Far from obstacles, $\mathbf{u}_j^*(t)$ is equivalent to the original feedback law (17), so descent direction is steepest.*

[3] Usually, this vector is along the line that connects the center of the obstacle to the target point that the robot is supposed to reach, and thus is usually constant (in the case of static obstacles). This is not true in our case; $\mathbf{p}_j$ changes at each time step as will be shown later.

We set $\mathbf{p}_j = \frac{\mathbf{B_j}(t)}{\|\mathbf{B_j}(t)\|_2}$. Notice that the value of $\mathbf{p}_j$ changes at each time step, thus the vector field $\mathbf{F}_j^o(\mathbf{r}_j)$, around the obstacle o, changes at each time step. This choice of $\mathbf{p}_j$ prevents the agent from colliding with the obstacle, and at the same time is primary in the proof of **Claim 1** that ergodicity is maintained as will be shown later.

As shown in Fig. 2, $\mathbf{V}_j^{opt}(t)$ and $\mathbf{F}_j^o(\mathbf{r}_j)$ form an acute angle wherever the robot is in the vicinity of the obstacle. This fact is true by the construction of the repulsive vector field and by our choice of $\mathbf{p}_j$. Thus, $\mathbf{V}_j^{opt}(t)^T\mathbf{F}_j^o(\mathbf{r}_j) \geq 0$. Equivalently, $\mathbf{p}_j^T\mathbf{F}_j^o(\mathbf{r}) \leq 0$ since $\mathbf{p}_j = -\mathbf{V}_j^{opt}(t)$. Therefore,

$$\mathbf{B}_j(t)^T\mathbf{F}_j^o(\mathbf{r}_j) \leq 0. \tag{26}$$

Moreover, far from the obstacle, $\alpha = 1$, then (24) simplifies to

$$\mathbf{u}_j(t) = u_{max}\mathbf{V}_j^{opt}(t) = -u_{max}\frac{\mathbf{B}_j(t)}{\|\mathbf{B}_j(t)\|_2} \tag{27}$$

which is nothing except the optimal feedback given by equation (17).

Now that we have defined the feedback law (24) that guarantees that the robot avoids obstacles, we have to prove **Claim 1**. In order to do this, we plug (24) in (15), also we replace $\mathbf{V}_j^{opt}$ by its expression given by (18), and we get,

$$\begin{aligned}
\ddot{\Phi}(t) &= \sum_K \Lambda_k(W_k(t))^2 + \sum_{j=1}^N \mathbf{B}_j(t)\,.\,\mathbf{u}_j^*(t) \\
&= \sum_K \Lambda_k(W_k(t))^2 \\
&\quad + \sum_{j=1}^N \mathbf{B}_j(t)\,.\,\Big[u_{max}\frac{\alpha\mathbf{V}_j^{opt}(t) + (1-\alpha)\mathbf{F}_j^o(\mathbf{r}_j)}{\|\mathbf{V}_j(t)\|_2}\Big] \\
&= \sum_K \Lambda_k(W_k(t))^2 + \frac{u_{max}}{\|\mathbf{V}_j(t)\|_2}\Big[\sum_{j=1}^N \mathbf{B}_j(t)\,.\,\alpha\mathbf{V}_j^{opt}(t) \\
&\quad + \sum_{j=1}^N \mathbf{B}_j(t)\,.\,(1-\alpha)\mathbf{F}_j^o(\mathbf{r}_j)\Big] \\
&= \sum_K \Lambda_k(W_k(t))^2 + \frac{u_{max}}{\|\mathbf{V}_j(t)\|_2}\Big[\alpha\sum_{j=1}^N -\|\mathbf{B}_j(t)\|_2 \\
&\quad + (1-\alpha)\sum_{j=1}^N \mathbf{B}_j(t)\,.\,\mathbf{F}_j^o(\mathbf{r}_j)\Big]
\end{aligned} \tag{28}$$

Notice that the first term in equation (28) is always positive and does not depend on $\mathbf{u}_j(t)$. Besides, we have shown in (26) that $\mathbf{B}_j(t)^T\mathbf{F}_j^o(\mathbf{r}_j) \leq 0$, then the term in brackets is always negative. Therefore, our choice of input decreases the second time-derivative of $\Phi(t)$ subject to the constraint $\|\mathbf{u}_j(t)\|_2 \leq u_{max}$ at each time step, thus driving $\Phi(t)$ to zero, and equivalently guaranteeing that agents will have ergodic dynamics. □

By looking at the term of equation (28) in the brackets, we see that it represents a descent direction which is not the steepest. The steepest descent direction is for $\alpha = 1$, i.e., when the agent is far from the obstacle. This takes us back to the case of no obstacles described in equations (15) and (17).

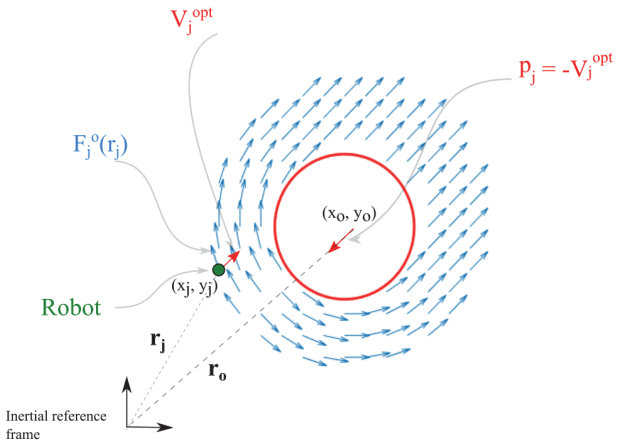


Figure 2: The robot (green dot) is driven near an obstacle (Red circle) via a repulsive vector field (blue vectors). This vector field is oriented depending on a vector $\mathbf{p}_j = -\mathbf{V}_j^{opt}(t)$. The robot is driven along a "blended vector" that is the combination of $\mathbf{F}_j^o(\mathbf{r}_j)$, and the vector $\mathbf{V}_j^{opt}(t) = -\frac{\mathbf{B_j}(t)}{\|\mathbf{B_j}(t)\|_2}$.

In the case of M distinct obstacles, a trivial extension of the vector $\mathbf{V}_j^*(t)$, defined in (25), is as follows,

$$\mathbf{V}_j^*(t) := (\prod_{i=1}^M \alpha_i)\mathbf{V}_j^{opt}(t) + \sum_{i=1}^M (1-\alpha_i)\mathbf{F}_j^{o_i}(\mathbf{r}_j) \tag{29}$$

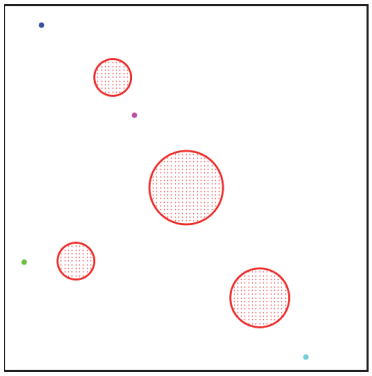
where $\mathbf{F}_j^{o_i}(\mathbf{r}_j)$ is the repulsive vector field (23) around an obstacle o_i, $i \in \{1,2,..,M\}$, and α_i is a linear bump function around o_i. Notice that whenever agent j is close to an obstacle, α_i becomes zero for some $i \in \{1,2,..,M\}$, thus, the first term in (29) vanishes and cancels the effect of $\mathbf{V}_j^{opt}(t)$. Also, whenever agent j is far from all the obstacles, α_i becomes 1 for all $i \in \{1,2,..,M\}$, thus, the second term in (29) vanishes, and (29) simplifies back to (18).

B. Second Order Systems

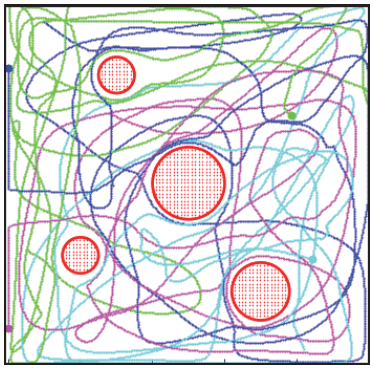
For second order systems of the form given in equation (19), we define the feedback law $\mathbf{u}_j^*$ to be the same as the original optimal feedback law given by equation (21) when the agent is far from the obstacle. In the vicinity of an obstacle, $\mathbf{u}_j^*$ should allow the agent to track the repulsive vector field defined in equation (23) and shown in Fig. 1. Thus, we need a control policy that allows an agent with second order dynamics to track a vector field. Following Rizzi (1998), we define a velocity reference control policy of the form,

$$\mathbf{u}_j^* = K(\mathbf{F}_j^o(\mathbf{x}_j) - \dot{\mathbf{x}}_j) + \dot{\mathbf{F}}_j^o(\mathbf{x}_j) \tag{30}$$

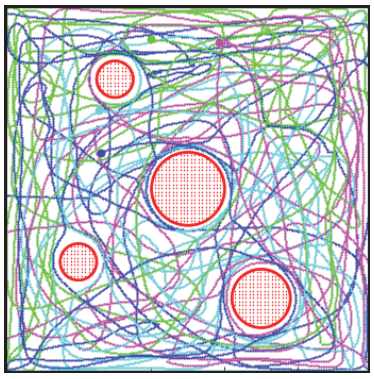
where $K > 0$ is velocity regulation gain that serves in decreasing the error $(\mathbf{F}_j^o(\mathbf{x}_j) - \dot{\mathbf{x}}_j)$, and $\dot{\mathbf{F}}_j^o(\mathbf{x}_j)$ is a feedforward term that ensures that the system tracks the reference velocity vector field $\mathbf{F}_j^o(\mathbf{x}_j)$. Here, $\mathbf{F}_j^o(\mathbf{x}_j)$ is as defined



(a) Time $t = 0$

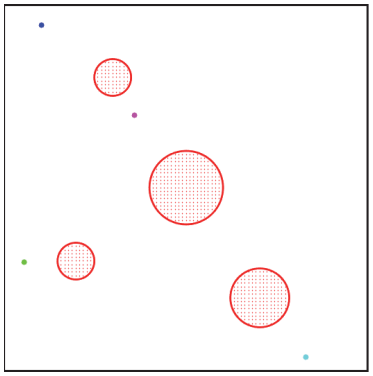


(b) Time $t = 2.5$

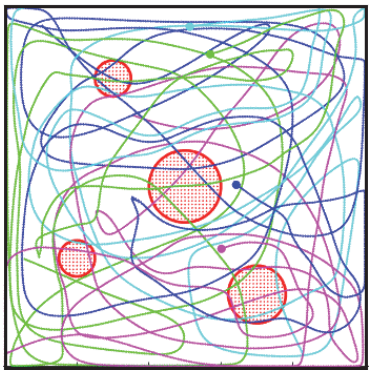


(c) Time $t = 5$

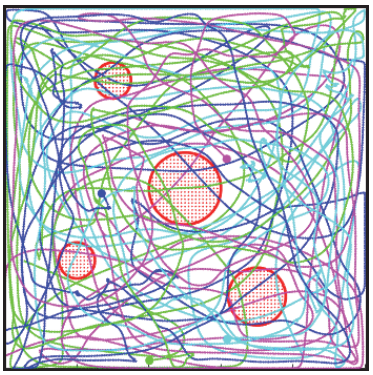
Figure 3: Snapshots at three different time instants of the trajectories of four agents uniformly covering a unit square. The agents are modeled as idealized first order dynamical systems. The red disks are obstacles.



(a) Time $t = 0$



(b) Time $t = 2.5$



(c) Time $t = 5$

Figure 4: Snapshots at three different time instants of the trajectories of four agents uniformly covering a unit square using the algorithm implemented by Mathew and Mezić (2011). The agents are modeled as idealized first order dynamical systems.

in (23). We noticed that by fixing the value of $\mathbf{p}_j$ around an obstacle, we avoid abrupt changes of the reference vector field $\mathbf{F}_j^o(\mathbf{x}_j)$, hence, the agents have better tracking for $\mathbf{F}_j^o(\mathbf{x}_j)$. Thus, we define $\mathbf{p}_j = \frac{\mathbf{B_j}(t_e)}{\|\mathbf{B_j}(t_e)\|_2}$, where t_e is the time instant at which the agent enters into the vicinity of an obstacle. The value of $\mathbf{p}_j$ never changes as long as the agent is in the vicinity of the obstacle. It only changes when the agent exits and then enters again into the vicinity of an obstacle.

The feedback law in (30) may violate the constraint of the bounded acceleration F_{max} of the agents. For example, in regions where the vector field is changing, tracking the vector field with $\dot{\mathbf{x}}_j = \mathbf{F}_j^o(\mathbf{x}_j)$ requires $\|\mathbf{u}_j(t)\|_2 = \|\dot{\mathbf{F}}_j^o(\mathbf{x}_j)\|_2$ (Conner, Choset, and Rizzi 2009). Thus, the control law (30) does not guarantee that the acceleration constraint $\|\mathbf{u}_j(t)\|_2 = \|\ddot{\mathbf{x}}_j(t)\|_2 \leq F_{max}$ is respected.

Therefore, we modify (30) so that the bounded acceleration constraint $\|\mathbf{u}_j(t)\|_2 \leq F_{max}$ is not violated. Conner, Choset, and Rizzi (2009) proved that by scaling down the

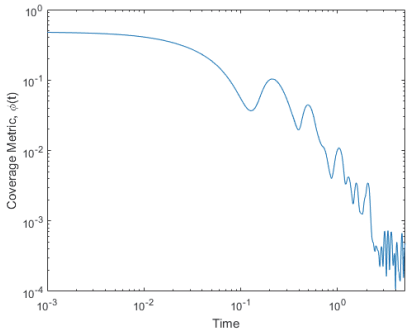


Figure 5: The decay of the metric of coverage $\phi(t)$ with time for first order dynamics using our algorithm.

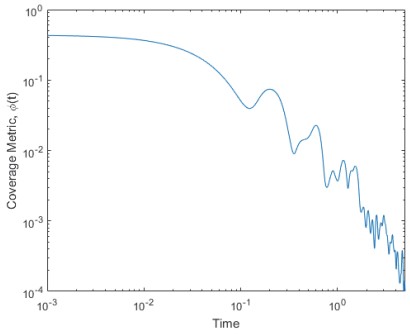


Figure 6: The decay of the metric of coverage $\phi(t)$ with time for first order dynamics using the algorithm implemented by Mathew and Mezić (2011).

velocity reference vector field, the agents are able to track a reference vector field without violating the bounded acceleration constraint. This encodes the idea of "slowing down when turning". We adopt this idea and scale down our vector field $\mathbf{F}_j^o(\mathbf{x}_j)$ by a factor $s(\mathbf{x}_j)$ defined in (Conner, Choset, and Rizzi 2009) as follows,

$$s(\mathbf{x}_j) = \min\left(\frac{s^*}{\|D_{\mathbf{x}}\mathbf{F}_j^o(\mathbf{x}_j)+\varepsilon\|_{sp}}, V_{max}\right) \tag{31}$$

where $\varepsilon > 0$ is an arbitrary constant, $D_{\mathbf{x}}$ is the Jacobian operator with respect to $\mathbf{x}$, $\|\cdot\|_{sp}$ is the spectral norm[4], V_{max} is the maximum speed of the agent in the domain, and s^* is a constant that is chosen offline such that it satisfies,

$$0 \le s^* \le \min_{\mathbf{x}} \frac{\sqrt{F_{max}}\left(\|D_{\mathbf{x}}\mathbf{F}_j^o\|_{sp}+\lambda\right)}{\sqrt{\|D_{\mathbf{x}}\mathbf{F}_j^o - \frac{\mathbf{F}_j^o D_{\mathbf{x}}\|D_{\mathbf{x}}\mathbf{F}_j^o\|_{sp}}{\|D_{\mathbf{x}}\mathbf{F}_j^o\|_{sp}+\lambda}\|_{sp}}} \tag{32}$$

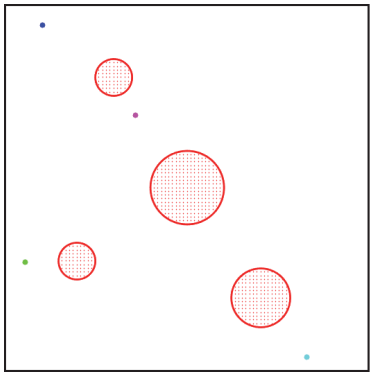
4 Simulation Results

A. Static Obstacles

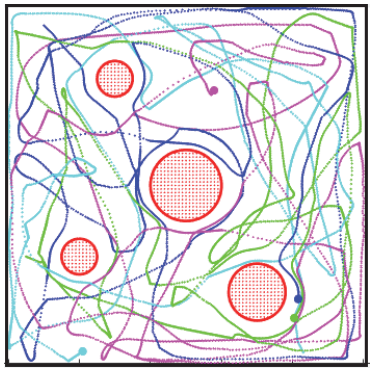
As a demonstration of our algorithm, we perform numerical simulations. Our goal in these simulations is to uniformly cover a unit square having multiple stationary circular obstacles using four agents. For the two scenarios that we simulate, the agents either have first order dynamics (equation (7)) with a maximum velocity u_{max}, or second order dynamics (equation (19)) with a maximum acceleration of F_{max}. For both scenarios, the target probability distribution $\mu(\mathbf{x})$ is defined as,

$$\mu(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ is outside an obstacle} \\ 0, & \text{if } \mathbf{x} \text{ is inside an obstacle} \end{cases} \tag{33}$$

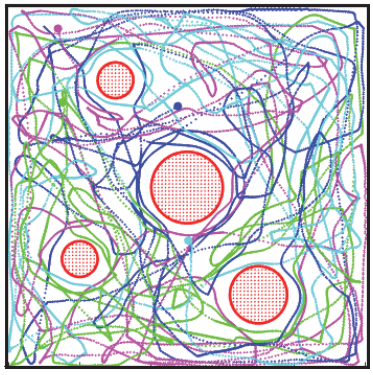
[4]The spectral norm of a matrix M, denoted $\|M\|_{sp}$, is defined as $\|M\|_{sp} = \max_{\|\mathbf{x}\|=1}\|M\mathbf{x}\|_2$



(a) Time $t = 0$



(b) Time $t = 15$



(c) Time $t = 30$

Figure 7: Snapshots at three different time instants of the trajectories of four agents uniformly covering a unit square. The agents are modeled as idealized second order dynamical systems. The red disks are obstacles.

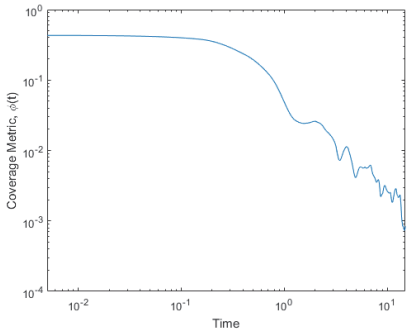


Figure 8: The decay of the metric of coverage $\phi(t)$ with time for second order dynamics.

where $\mathbf{x} \in [0,1] \times [0,1]$ is a point in the unit square.

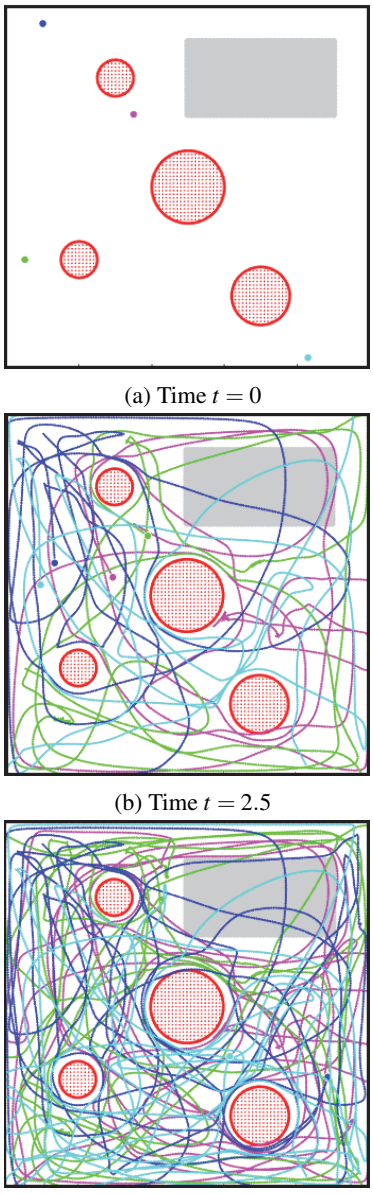
First Order Dynamics We ran a simulation of four agents having first order dynamics with $u_{max} = 5.0$ and a total simulation time $T = 5.0$. The initial positions of the agents are randomly selected. Fig. 3 shows snapshots of the trajectories of the agents at three different time instants. Notice how the agents were able to uniformly cover the free space of the unit square while avoiding four randomly positioned obstacles. Fig. 5 shows a plot of the decay of the metric of uniform coverage $\phi(t)$ as a function of time, which supports our claim that ergodic coverage is maintained. Results can be found in the supplementary video[5].

Comparing our results shown in Fig. 3 with the results of the algorithm implemented by Mathew and Mezić (2011) shown in Fig. 4, we can see that although both methods minimize ergodicity as shown in Fig. 5 and Fig. 6, our method avoids collision with any obstacle, whereas Mathew and Mezić's (2011) algorithm steers trajectories away from the obstacles, but does not guarantee obstacle avoidance.

Second Order Dynamics We ran a simulation of four agents having second order dynamics with $F_{max} = 100.0$, $c = 2.5$, a velocity regulation gain $K = 30.0$, and a total simulation time $T = 30.0$. The initial positions of the agents are randomly selected. Fig. 7 shows snapshots of the trajectories of the agents at three different time instants. Notice how the agents again were able to uniformly cover the free space of the unit square while avoiding four randomly positioned obstacles. Fig. 8 shows a plot of the decay of the metric of uniform coverage $\phi(t)$ as a function of time, which supports our claim that ergodic coverage is maintained. Results can be found in the supplementary video[5].

C. Nonuniform Coverage

We also test our method on a nonuniform target distribution $\mu(\mathbf{x})$. The results are shown in Fig. 9. In this simulation, four agents, having first order dynamics with $u_{max} = 5.0$ and a total simulation time $T = 5.0$, are supposed to cover the unit square shown in Fig. 9-a. The red disks are obstacles. The gray box is a region with low probability $\mu(\mathbf{x})$ over which the agents can pass, but should not spent much time. You can think of this problem as a surveillance problem. The gray region can be thought of areas where no sensor measurements can be taken due to foliage. Thus, agents must spend less time in such regions. The initial positions of the agents are randomly selected. Fig. 9 shows snapshots of the trajectories of the agents at three different time instants. Fig. 10 shows a plot of the decay of the metric of uniform



(a) Time $t = 0$

(b) Time $t = 2.5$

(c) Time $t = 5$

Figure 9: Snapshots at three different time instants of the trajectories of four agents uniformly covering a nonuniform distribution on a unit square. The agents are modeled as idealized first order dynamical systems. The red disks are obstacles, and the gray box is a region with low probability $\mu(\mathbf{x})$ over which agents can pass.

[5] Video at https://youtu.be/fcLh0GUJbtI

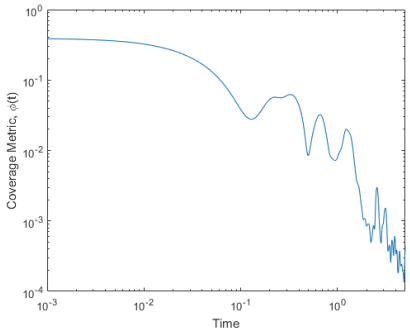


Figure 10: The decay of the metric of coverage $\phi(t)$ with time for the nonuniform coverage distribution scenario.

coverage $\phi(t)$ as a function of time. Results can be found in the supplementary video[5].

C. Dynamic Obstacles

For dynamic obstacles, our formulation still applies. We demonstrate our method on dynamic obstacles in a simulation of four agents having first order dynamics and covering a unit square containing three obstacles that are moving in a random manner. We use $u_{max} = 5.0$ and a total simulation time $T = 5.0$. The results are shown in the supplementary video[5].

5 Conclusion and Future Work

In this paper, a methodology that enables multi-agent systems to uniformly cover a domain while avoiding static and dynamic obstacles is introduced. We use an ergodic metric to ensure uniform coverage of the domain, and we leverage tools from vector-field-based obstacle avoidance community. This enables us to define feedback control laws that drive the agents in the coverage domain safely without colliding with any obstacle, and at the same time ensures the agents uniformly cover the free space of the domain. Our methodology is demonstrated via simulation of first order and second order dynamical multi-agent systems.

Future work will focus on extending this approach to obstacles with arbitrary shapes, and evaluating it via physical experiments on quadrotors.

References

Conner, D. C.; Choset, H.; and Rizzi, A. A. 2009. Flow-through policies for hybrid controller synthesis applied to fully actuated systems. *IEEE Transactions on Robotics* 25(1):136–146.

Hubenko, A.; Fonoberov, V. A.; Mathew, G.; and Mezic, I. 2011. Multiscale adaptive search. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41(4):1076–1087.

Lindemann, S. R., and LaValle, S. M. 2007. Smooth feedback for car-like vehicles in polygonal environments. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 3104–3109. IEEE.

Mathew, G., and Mezic, I. 2009. Spectral multiscale coverage: A uniform coverage algorithm for mobile sensor networks. In *CDC*, 7872–7877. Citeseer.

Mathew, G., and Mezić, I. 2011. Metrics for ergodicity and design of ergodic dynamics for multi-agent systems. *Physica D: Nonlinear Phenomena* 240(4):432–442.

Miller, L. M., and Murphey, T. D. 2013a. Trajectory optimization for continuous ergodic exploration. In *2013 American Control Conference*, 4196–4201. IEEE.

Miller, L. M., and Murphey, T. D. 2013b. Trajectory optimization for continuous ergodic exploration on the motion group se (2). In *52nd IEEE Conference on Decision and Control*, 4517–4522. IEEE.

Nocedal, J., and Wright, S. 2006. *Numerical optimization*. Springer Science & Business Media.

Panagou, D. 2014. Motion planning and collision avoidance using navigation vector fields. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2513–2518. IEEE.

Petersen, K. E. 1989. *Ergodic theory*, volume 2. Cambridge University Press.

Rizzi, A. A. 1998. Hybrid control as a method for robot motion programming. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, 832–837. IEEE.