

Decentralized Ergodic Control: Distribution-Driven Sensing and Exploration for Multi-Agent Systems

Ian Abraham and Todd D. Murphey

Abstract—We present a decentralized ergodic control policy for time-varying area coverage problems for multiple agents with nonlinear dynamics. Ergodic control allows us to specify distributions as objectives for area coverage problems for nonlinear robotic systems as a closed-form controller. We derive a variation to the ergodic control policy that can be used with consensus to enable a fully decentralized multi-agent control policy. Examples are presented to illustrate the applicability of our method for multi-agent terrain mapping as well as target localization. An analysis on ergodic policies as a Nash equilibrium is provided for game theoretic applications.

Index Terms—List of keywords (from the RA Letters keyword list)

I. INTRODUCTION

IN the task of exploration and area coverage, decentralized robot networks have been shown to improve the sensing capacity of mobile robots [1], [2], [3] while minimizing computation for individual robotic agents. Shifting the computation to the individual robotic agent becomes necessary as the size of the multi-agent network becomes large and coordination of the network becomes costly for a single computing unit to calculate [3], [4], [5], [6]. This is more of an issue if the coordination algorithm becomes more complex with each agent, making it more desirable to have local, decentralized computation that only relies on neighboring information. This is also true as the underlying task becomes complex and additional environmental considerations must then be taken into account. In this paper, we present an algorithm for dynamic decentralized area coverage derived from ergodic control [7] that admits nonlinearities in the dynamics of robot and is general to many applications of multi-agent coordination.

Ergodic control [8], [9], [10], [7], [11] enables area coverage for robotic agents with nonlinear dynamics that is general to many applications. By specifying the ergodic metric for area coverage, it was shown that one can synthesize trajectories that maximally optimize the ergodic metric, resulting in persistent coverage,¹ visitation of the entire exploration domain [8], [9], [12], and resilience to distractors in localization tasks [13],

[7]. In [7] it was shown that one can formulate the ergodic control algorithm as a centralized ergodic controller for multiple agents. However, it has yet to be shown how one can decentralize the algorithm for use in larger, more complex, multi-agent systems where control decisions are made on an individual basis. Thus, the contribution of this work is a formulation of ergodic control for a multi-agent network as a decentralized algorithm that, through consensus, solves various forms of persistent area coverage problems using the ergodic metric for agents with nonlinear dynamics.

Existing work in multi-agent coordination addresses the problems of area coverage [14], [15], [16], inclusion of sensor constraints [17], [18], and localization and estimation [19], [18]. While these methods address specific problems in decentralized coordination, none of these methods have been shown to be flexible enough to solve all the problems. While we initially frame our algorithm for area coverage, we provide additional examples for target localization, terrain estimation, and coverage in corridors to show that our method can be generalized to other tasks seen in multi-agent coordination [18], [19] without the need to change the specification of our algorithm. Moreover, our method is distinct from coverage algorithms that rely on Voronoi segmentation of the environment to make coordinated decisions [17], [14], [16], [15], [20]. Voronoi segmentation requires the specification of a metric for generation of the segmentation in addition to a metric for control and area coverage of each individual robotic agent. When the dynamics of the robot are nonlinear, control synthesis requires additional assumptions or metrics [20], [15]. Our method only uses the ergodic metric to formulate control for nonlinear dynamics [7]. Moreover, one can specify the ergodic metric with respect to information densities based on measurement models that include sensor physics/constraints [7], [9]. We show in Section III-C that the requirement of our decentralized algorithm is that the agents need only communicate coefficients representing their actions in order to make independent decisions that reduce the ergodic metric.

The outline of the paper is as follows: Section II defines the problem of area coverage for multi-agent networks. Section III introduces ergodicity and the ergodic metric as well as formulates the ergodic control problem for decentralized multi-agent systems. A game theoretic analysis on ergodic control policies is provided in Section IV. Section V demonstrates the algorithm on an area-coverage problem for multi-agents. We then present the problem for multi-agent target localization in Section VI and the conclusion is in Section VII.

This material is based upon work supported by the National Science Foundation under awards IIS-1426961 and IIS-1717951. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Authors are with the Neuroscience and Robotics Laboratory (NxR) at the Department of Mechanical Engineering, Northwestern University, 2145 Sheridan Road Evanston, IL 60208 USA.

Email: i-abr@u.northwestern.edu,
t-murphey@northwestern.edu

Manuscript received April 19, 2005; revised August 26, 2015.

¹In the sense that the robot is always in motion.

II. MULTI-AGENT AREA COVERAGE

In this section we present the problem statement that our method solves. Let us consider a set of N heterogeneous robotic agents where the evolution of the i^{th} agent's state $x_i(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^n$ at time $t \in \mathbb{R}^+$ is governed by the deterministic nonlinear equation

$$\dot{x}_i = f_i(x_i(t), u_i(t)) \quad (1)$$

where $u_i(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^m$ is an applied control and $f_i(x, u) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a nonlinear function. Furthermore, let us define a bounded domain $\mathcal{X}_v \subset \mathbb{R}^v$ whose limits are $[0, L_1] \times [0, L_2] \times \dots [0, L_v]$ with $v \leq n$. We can consider this bounded domain a “search space” where we can define any arbitrary spatial statistic $\phi(s) : \mathcal{X}_v \rightarrow \mathbb{R}^{+2}$ where $s \in \mathbb{R}^v \subset \mathbb{R}^n$. Typically $\phi(s)$ is generated from the expected information density [9], [7] based on the measurement model and sensor constraints. The goal of multi-agent area coverage is to position the agents in a system in such a manner that the states of the system $x(t) = [x_1(t)^\top, \dots, x_N(t)^\top]^\top$ are proportional to the spatial statistics $\phi(s)$. That is, we want the statistics of the trajectory of the robots, which we will define as $c(s, x(t))$ to be equal to the spatial statistics $\phi(s)$ through some metric (in our case ergodicity).

We note that our approach treats the problems of target tracking, estimation, and area coverage as the same problem of persistent area coverage, that is, the spatial statistics contains the information for all these problems (which we specify in Sections VI and V). Moreover, we emphasize persistent area coverage because the basis of the ergodic metric (see Section III-A) revolves around the time-averaged statistics of the multi-agent system trajectories in the search space. This results in persistent movement and monitoring, rather than placement, of an agent.

The following section formulates the decentralized ergodic controller for a multi-agent system.

III. DECENTRALIZED ERGODIC CONTROL

In this section, ergodicity and the ergodic metric are introduced and we formulate an ergodic control policy for multi-agent systems. We make note of the terminology *distributed* and *decentralized* used in this paper as two distinct terms:

Definition 1: A *distributed* algorithm is one where the initialization of the optimization occurs in a centralized computer hub and then the calculation for the optimization are offloaded onto a set of individual computation units.

Definition 2: A *decentralized* algorithm is one where each individual computational unit solves their own optimization problem that, through communication with a network, solves a larger global optimization problem (typically using some form of consensus) [21], [22].

A. Ergodicity and the Ergodic Metric

Assume the state at time t is given by $x(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^n$. Controls to the robot at time t are $u(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^m$.³

²Under the assumption that $\int_{\mathcal{X}_v} \phi(s) ds = 1$.

³We drop the i^{th} indexing notation for readability and to illustrate that the multi-agent system can be treated as a larger, unified system in later sections.

The dynamics of the robot are assumed to be governed by a control-affine dynamical system of the form

$$\dot{x}(t) = f(x(t), u(t)) = g(x(t)) + h(x(t))u(t) \quad (2)$$

where $g(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the free, unactuated dynamics of the robot, and $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ is the dynamic control response subject to input $u(t)$. Let us consider the robot's time-averaged statistics $c(s, x(t))$ for a trajectory $x(t)$ (i.e., the statistics describing where the robot spends most of its time) for some time interval $t \in [t_i, t_i + T]$ as

$$c(s, x(t)) = \frac{1}{T} \int_{t_i}^{t_i+T} \delta(s - x_v(t)) dt, \quad (3)$$

where δ is a Dirac delta function, $T \in \mathbb{R}^+$ is the time horizon, $t_i \in \mathbb{R}^+$ is the i^{th} sampling time, and $x_v(t) \in \mathbb{R}^v$ is the state that intersects with the search space. An ergodic metric [10] which relates the two distributions $c(s, x(t))$ and $\phi(s)$ is:

$$\begin{aligned} \mathcal{E}(x(t)) &= q \sum_{k \in \mathbb{N}^v} \Lambda_k (c_k - \phi_k)^2 \\ &= q \sum_{k \in \mathbb{N}^v} \left(\frac{1}{T} \int_{t_i}^{t_i+T} F_k(x(t)) dt - \phi_k \right)^2 \end{aligned} \quad (4)$$

where

$$\phi_k = \int_{\mathcal{X}_v} \phi(s) F_k(s) ds,$$

$q \in \mathbb{R}^+$ is a scalar weight on the metric, and c_k are the Fourier decompositions⁴ of $c(s, x(t))$ and $\phi(s)$ with

$$F_k(x) = \frac{1}{h_k} \prod_{i=1}^v \cos\left(\frac{k_i \pi x_i}{L_i}\right)$$

being the cosine basis function for a given coefficient $k \in \mathbb{N}^v$, h_k is a normalization factor defined in [10], and $\Lambda_k = (1 + \|k\|^2)^{-\frac{v+1}{2}}$ are weights on the frequency coefficients. A robot whose control inputs result in a trajectory $x(t)$ that minimizes (4) as $t \rightarrow \infty$ is then said to be optimally ergodic with respect to the target distribution.

Because we are computing the ergodic control in receding horizon, and the target distribution $\phi(s)$ can be time-varying, a history of where a robot has been is maintained in memory in order to compute the ergodic metric. The ergodic metric is then computed by adding a time parameter $\Delta t_{\mathcal{E}}$ which governs how far into the past the robot must remember where it has been. Equation (4) then becomes

$$\mathcal{E}(x(t)) = q \sum_{k \in \mathbb{N}^v} \left(\frac{1}{T_{\mathcal{E}}} \int_{t_i - \Delta t_{\mathcal{E}}}^{t_i + T} F_k(x(t)) dt - \phi_k \right)^2. \quad (5)$$

Note that choosing $\Delta t_{\mathcal{E}} = t_i$ would result in storing all past states. This can be avoided by recursively defining c_k as shown in [7]. In addition, choosing a $\Delta t_{\mathcal{E}} < T$ would result in very myopic behavior (i.e., only spending time in regions of high spatial statistics). This is often desired if a time-varying spatial distribution $\phi(s, t)$ is specified where past information is rendered uninformative as the underlying spatial statistics change

⁴The cosine basis function is used, however, any choice of basis function F_k can be used.

rapidly. In practice, a choice of $\Delta t_{\mathcal{E}} = 2T$ is empirically a reasonable start which can be tuned to performance needs after further evaluation.

B. Ergodic Control

In [8] the ergodic controller is formulated using a trajectory optimization scheme. While this approach does give optimal solutions, it is difficult for the controller to run in real time. As a result, [7] developed a hybrid systems approach using [23] to obtain control policies that sufficiently reduce the ergodic metric. We formulate our controller using a similar approach, but provide a variation to the controller that allows the policy to be fully distributable.

Rather than directly minimizing (4) with respect to $x(t)$ and $u(t)$, we consider the sensitivity of (4) with respect to an infinitesimal time of application $\lambda \in \mathbb{R}^+ \rightarrow 0$ of the best possible control $u_*(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^m$ that sufficiently reduces (4) at time $\tau \in \mathbb{R}^+$ from some default control $u_{\text{def}}(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^m$. Following [7], we take the derivative of (4) with respect to the duration time λ of control $u_*(t)$ which gives the sensitivity (known as the mode insertion gradient [24], [25], [26], [27]).

Proposition 1: The first order sensitivity of (4) with respect to the control duration λ of the applied control $u_*(\tau)$ is

$$\left. \frac{\partial \mathcal{E}}{\partial \lambda} \right|_{\tau} = \rho(\tau)^{\top} (f_2(\tau, \tau) - f_1(\tau)) \quad (6)$$

where $f_2(t, \tau) = f(x(t), u_*(\tau))$, $f_1(t) = f(x(t), u_{\text{def}}(t))$, and $\rho(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^n$ is given by the differential equation

$$\dot{\rho} = -2 \frac{q}{T} \sum_{k \in \mathbb{N}^v} \Lambda_k (c_k - \phi_k) \frac{\partial F_k}{\partial x} - \frac{\partial f}{\partial x}^{\top} \rho(t)$$

with $\rho(t_i + T) = \mathbf{0} \in \mathbb{R}^n$.

Proof: See [7] for more details. ■

The mode insertion gradient now represents the sensitivity of the ergodic metric with respect to an application of a control $u_*(t)$.

Given the mode insertion gradient, we seek to find the control $u_*(t)$ that most significantly decreases in the objective (4). We can write this as an unconstrained optimization problem of the form

$$J_2 = \int_{t_i}^{t_i+T} \left. \frac{\partial \mathcal{E}}{\partial \lambda} \right|_t + \frac{1}{2} \|u_*(t) - u_{\text{def}}(t)\|_R^2 \quad (7)$$

where $R \in \mathbb{R}^{m \times m}$ is a positive definite matrix that weighs $u_*(t)$. Note that (7) is quadratic in $u_*(t)$ which encodes a regularization term with respect the default control u_{def} and includes a cost on sufficient decrease in the mode insertion gradient. The minimizer of (7) with respect to $u_*(t)$ is the control that provides the most negative mode insertion gradient and reduces the objective (4).

Proposition 2: The solution to $u_*(t)$ that minimizes (7) is

$$u_*(t) = -R^{-1} h(x)^{\top} \rho(t) + u_{\text{def}}(t). \quad (8)$$

Proof: Taking the derivative of (7) with respect to control $u_*(t)$ and setting the solution to zero gives

$$\begin{aligned} \frac{\partial J_2}{\partial u_*} &= \int_{t_i}^{t_i+T} \frac{\partial}{\partial u_*} \left(\frac{\partial \mathcal{E}}{\partial \lambda} \right) + R(u_* - u_{\text{def}}) dt \\ &= \int_{t_i}^{t_i+T} h(x)^{\top} \rho + R(u_* - u_{\text{def}}) dt = 0 \end{aligned} \quad (9)$$

where the dependency on time is dropped for simplicity. Solving for u_* in (9) gives

$$u_*(t) = -R^{-1} h(x(t))^{\top} \rho(t) + u_{\text{def}}(t). \quad \blacksquare$$

Lemma 1: Assuming that $h(x)^{\top} \rho \neq 0$, the mode insertion gradient in (6) is always negative for $u_*(t)$ defined in (8), that is $\frac{\partial \mathcal{E}}{\partial \lambda} < 0 \forall u_* \in \mathcal{U}$ where $\mathcal{U}$ is the control space.

Proof: Inserting (8) into (6) gives

$$\begin{aligned} \frac{\partial \mathcal{E}}{\partial \lambda} &= h(x)^{\top} \rho (-R^{-1} h(x)^{\top} \rho) \\ &= -\rho^{\top} h(x) R^{-1} h^{\top} \rho = -\|h(x)^{\top} \rho\|_{R^{-1}}^2 < 0. \end{aligned} \quad (10)$$

Thus (10) shows us that $\forall u_* \in \mathcal{U}$ defined in (8), $\frac{\partial \mathcal{E}}{\partial \lambda} < 0$. ■ Because (8) always provides a negative $\frac{\partial \mathcal{E}}{\partial \lambda}$, this implies that each control that is chosen will result in a decrease in (4); thus eventually minimizing the ergodic metric. Additionally, as in [7], a contractive constraint on the reduction of the ergodic metric is enforced that further provides a reduction in the ergodic metric from the previous control calculation time.

In many robotics applications, it is required that the control is saturated due to actuation limits in the robot while maintaining some form of sufficient decrease in the objective cost. In this work, we select a time of application τ that results in the most negative mode insertion gradient, or more formally written by

$$\tau_* = \underset{\tau}{\operatorname{argmin}} \frac{\partial \mathcal{E}}{\partial \lambda}$$

where the subscript $\star$ indicates the time of application that results in the most negative mode insertion gradient. A line search [28] is then used to find the duration λ that significantly reduces (4) subject to the saturated control $u_*(\tau)$. The resulting control is then added to the default control $u_{\text{def}}(t) = u_*(\tau) \forall t \in [\tau, \tau + \lambda] \cap [t_i, t_i + t_s]$ where t_s is the sampling time and $u_*(\tau)$ is saturated.

The following subsection derives the ergodic control policy for a decentralized multi-agent systems.

C. Decentralized Ergodic Control using Consensus

Consider a set of N agents with state $x(t) = [x_1(t)^{\top}, x_2(t)^{\top}, \dots, x_N(t)^{\top}]^{\top} : \mathbb{R}^+ \rightarrow \mathbb{R}^{nN}$.⁵

Proposition 3: Given the default trajectory of each agent $x(t) \forall t \in [t_i - \Delta t_{\mathcal{E}}, t_i + T]$ subject to $u_{\text{def}}(t)$, the control policy (8) is *distributable* amongst each individual agent and independent of the other agent's control policy.

⁵For readability we consider a homogeneous set of agents with the same state dimension $x_i(t) \in \mathbb{R}^n$. However, this analysis can be done for a heterogeneous set of agents with arbitrary dynamics and state dimensions.

Proof: Let us first define the dynamics of the collective multi-agent system as

$$\begin{aligned} \dot{x} &= f(x, u) = g(x) + h(x)u \\ &= \begin{bmatrix} g_1(x_1) \\ g_2(x_2) \\ \vdots \\ g_N(x_N) \end{bmatrix} + \begin{bmatrix} h_1(x_1) & \dots & 0 \\ \vdots & \ddots & \\ 0 & & h_N(x_N) \end{bmatrix} u \end{aligned} \quad (11)$$

where $h(x)$ is block diagonal. The multi-agent system's contribution to the time-averaged statistics c_k can be rewritten as

$$\begin{aligned} c_k &= \frac{1}{N} \sum_{j=1}^N \frac{1}{T_{\mathcal{E}}} \int_{t_i - \Delta t_{\mathcal{E}}}^{t_i + T} F_k(x_j(t)) dt \\ &= \frac{1}{T_{\mathcal{E}}} \int_{t_i - \Delta t_{\mathcal{E}}}^{t_i + T} \tilde{F}_k(x(t)) dt \end{aligned} \quad (12)$$

where $\tilde{F}_k(x(t)) = \frac{1}{N} \sum_j F_k(x_j(t))$. The mode insertion gradient (6) under a multi-agent dynamical system now has $f_1(t)$ and $f_2(t, \tau)$ defined by (11) and the convolution equation for the adjoint variable $\rho(t)$ becomes

$$\dot{\rho} = -2 \frac{q}{T_{\mathcal{E}}} \sum_{k \in \mathbb{N}^v} \Lambda(c_k - \phi_k) \frac{\partial \tilde{F}_k}{\partial x} - \frac{\partial f}{\partial x}^{\top} \rho \quad (13)$$

where

$$\frac{\partial \tilde{F}_k}{\partial x} = \frac{1}{N} \begin{bmatrix} \frac{\partial F_k(x_1)}{\partial x_1} \\ \vdots \\ \frac{\partial F_k(x_N)}{\partial x_N} \end{bmatrix} \text{ and } \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & 0 & \dots & 0 \\ 0 & \frac{\partial f_2}{\partial x_2} & & \\ \vdots & & \ddots & \\ 0 & & & \frac{\partial f_N}{\partial x_N} \end{bmatrix}$$

is block diagonal. Because each agent's dynamics are independent of each other, (13) can be written independently for each agent as

$$\dot{\rho}_j = -2 \frac{q}{T_{\mathcal{E}} N} \sum_{k \in \mathbb{N}^v} \Lambda_k(c_k - \phi_k) \frac{\partial F_k(x_j)}{\partial x_j} - \frac{\partial f_j}{\partial x_j}^{\top} \rho_j.$$

Similarly, the ergodic control policy derived from (9) becomes

$$\begin{aligned} \begin{bmatrix} u_{*,1}(t) \\ \vdots \\ u_{*,N}(t) \end{bmatrix} &= R^{-1} \begin{bmatrix} h_1(x_1) & \dots & 0 \\ \vdots & \ddots & \\ 0 & & h_N(x_N) \end{bmatrix}^{\top} \begin{bmatrix} \rho_1(t) \\ \vdots \\ \rho_N(t) \end{bmatrix} \\ &\quad + \begin{bmatrix} u_{\text{def},1}(t) \\ \vdots \\ u_{\text{def},N}(t) \end{bmatrix} \end{aligned} \quad (14)$$

where $R \in \mathbb{R}^{mN \times mN}$ and mN is the size of the collective multi-agent system control input. Since $h(x)$ is block diagonal, (14) becomes

$$u_{*,j}(t) = -R_j^{-1} h_j(x_j)^{\top} \rho_j(t) + u_{\text{def},j}(t) \quad (15)$$

for each agent $j \in [1, \dots, N]$ and $R_j \in \mathbb{R}^{m \times m}$. The control policy in (15) for the j^{th} agent does not depend on the i^{th} agent and therefore is distributable. ■

While the j^{th} control policy is independent of the i^{th} control policy, it is assumed starting from (5) that each agent's past and anticipated trajectory is known to all agents before calculating

the control policy. We can consider this a distributed ergodic control policy where the control computation is still done on individual CPUs on-board the agents, but the initial conditions are required to be sent from a central communication hub. Instead of a distributed controller, we seek to completely remove the need for a centralized communication hub and have fully independent agents solve smaller ergodic control problems that solve the same larger multi-agent ergodic control problem. We address this problem using consensus-based methods where a network of agents communicates with one another the local c_k for the individual agent.

Rather than communicating the past and anticipated trajectories of each agent (which may have large dimensionality) in the network, we communicate the c_k values instead. ⁶

Proposition 4: A connected multi-agent network under consensus over the c_k coefficients approximates the time-average statistics c_k of the centralized ergodic metric (12), that is $\tilde{c}_k \rightarrow c_k$ as $t \rightarrow \infty$ where $\tilde{c}_k$ is the consensus-based time-average statistics.

Proof: Consider the collective time-averaged statistics c_k for the system in (12):

$$c_k = \frac{1}{N} \sum_{j=1}^N \frac{1}{T} \int_{t_i}^{t_i + T} F_k(x_j(t)) dt.$$

Equation (12) is simply an averaging of the individual agent's spatial statistics. Let us then define a row and column stochastic consensus matrix P (e.g., $\sum_j P_{ij} = 1$) that defines the network connectivity amongst the agents [21], [22]. The operation $\sum_j P_{ij} c_{k,j}$ is equivalent to taking an average of the local $c_{k,j}$ values for each neighboring agent. ⁷ Therefore, we can write a consensus on the collective c_k (12) using P as [21], [22]

$$\lim_{t_k \rightarrow \infty} \sum_j P_{ij}^{t_k} c_{k,j} = \frac{1}{N} \sum_j \frac{1}{T} \int_{t_0}^{t_0 + T} F_k(x_j(t)) dt$$

where N is the number of agents, t_k is the number of times that $P_{ij} c_{k,j}$ values have been communicated through the network and averaged. Thus consensus amongst all the agents approximates the collective multi-agent system time-averaged statistics c_k in (12). ■

Algorithm 1 is provided to illustrate the decentralized ergodic control policy for multi-agent systems.

D. Communication Complexity and Scalability

Since the ergodic metric is defined in terms of the Fourier coefficients of the agent's trajectory and the spatial statistics, each agent is only required to transmit their own $c_{k,j}$ trajectory coefficients. The benefit of this is two-fold: First, each agent in the decentralized network need only store their own past trajectory information for computing $c_{k,j}$. Thus, the required storage for a 64 bit memory is $64 * \Delta t_{\mathcal{E}} / t_s * n$ bits where t_s is the sampling rate. We can further reduce the memory

⁶It is assumed that each agent has the same ϕ_k target, however, the same analysis can be done to form a consensus on the target ϕ_k values.

⁷For simplicity in notation, we assume that P_{ij} refers to a block matrix such that $P \in \mathbb{R}^{|k|N \times |k|N}$ and $P_{ij} \in \mathbb{R}^{|k| \times |k|}$ where $|k|$ is the total number of c_k coefficients.

Algorithm 1 Decentralized Ergodic Control

```

1: initialize: agents  $N$  with initial condition  $x_j(0)$ , initial
   target distribution  $\phi_{k,0}$ ,  $t_0, t_f, t_s$ , time horizon  $T$ , ergodic
   memory  $\Delta\mathcal{E}$  and network  $P$ .
2: while  $t_i < t_f$  do
3:   for each agent do ▷ Control step
4:     simulate  $x(t), \rho(t)$  for  $t \in [t_i, t_i + T]$  from  $x(t_i)$ 
5:     compute  $u_*(\tau)$  from (8)
6:     calculate  $\tau$  and  $\lambda$  from [23], [7]
7:      $u_{\text{def}}(t) = u_*(\tau) \forall t \in [\tau, \tau + \lambda] \cap [t_i, t_i + t_s]$ 
8:   end for
9:   for each agent do ▷ Communication Loop
10:    Send  $c_{k,j}$  to  $i^{\text{th}}$  neighbors in the network  $P$ 
11:    Receive  $c_{k,i}$  from neighbors and average amongst
        $i^{\text{th}}$  neighbors
12:   end for
13:   apply control  $u_{\text{def}}(t_i)$ 
14:    $i \leftarrow i + 1$ 
15: end while

```

requirements by recursively defining the $c_{k,j}$ values as done in [7]. The second benefit is in the complexity and scale of the algorithm as the number of agents increases. Since each agent only needs to communicate their local $c_{k,j}$ values to their neighbors, the computational burden lies in computing the ergodic control for the individual agents themselves. Because we have shown that we can fully decentralize the ergodic control calculations, the computation remains constant to each robot. Thus, the computational complexity of the ergodic controller only scales with the dimensions of the single agent's state (which for practical purposes will remain constant as the agents' state dimensions are not time-varying) and the decentralized algorithm does not scale by increasing the number of agents in the network.

In the following section, we provide an analysis of the ergodic control policy in a game-theoretic point of view.

IV. ERGODIC CONTROL POLICIES AS NASH EQUILIBRIUM STRATEGIES

In this section, we analyze the ergodic control policy from a game theoretic point of view in adversarial multi-agent games.

Definition 3: A game is defined by a tuple $(\mathcal{P}, \mathcal{A}, \mathcal{O}, \mu, \mathcal{U})$ where $\mathcal{P}$ is the set of players in a game, $\mathcal{A}$ is the set of control actions $u(t)$ where $u_i(t) \forall t \in [t_i, t_i + T], \forall i \in \mathcal{P}$ is considered an action or strategy profile, $\mathcal{O}$ is the set of outcomes (or state trajectories in our case), $\mu : \mathcal{A} \rightarrow \mathcal{O}$ is the function that maps actions to outcomes (in our case this is the robot dynamics), and last $\mathcal{U} : \mathcal{O} \rightarrow \mathbb{R}$ is a utility function that we index for each i^{th} player using the subscript i .

Each agent is defined by $\mathcal{P}$. The action profile or strategy $\mathcal{A}$ is defined by the ergodic control policy subject to a target distribution. The resultant trajectory $x(t)$ for each agent is the outcome $\mathcal{O}$ subject to the actions passing through the dynamics $f(x, u)$ of the system (μ). Here, we treat the utility function $\mathcal{U}$ as the ergodic metric. In game theory, the notion of Nash

equilibrium [29], [30] is often used to describe a strategy in a game.

Definition 4: A strategy is a Nash equilibrium if for each agent i , $\mathcal{U}_i(u) \leq \mathcal{U}_i(u_{-i})$ where u_{-i} is the updated strategy profile for all agents not including agent i 's strategy.

Nash equilibrium tells us whether a strategy results in the best possible expected utility of each agent subject to the other agents' actions. We consider Nash equilibrium in the problem of target localization and evasion. Specifically, we look at what strategy an evader can use to acquire a Nash equilibrium with the pursuer (localizer) (i.e., a game between the pursuer and evader while the pursuer expends energy not localizing the evader).

Theorem 1: A Nash equilibrium strategy against a pursuer with an ergodic policy is for the evader to adopt an ergodic policy.

Proof: Consider two agents, a and b on opposing sides of a game. Agent a is ergodic with respect to a target distribution $\phi_a(s)$. Agent b is ergodic with respect to $\phi_b(s)$. We assume that the target distribution of agent a and b is a function of the state of the agents, that is, $\phi_a(s) = \phi_a(s, x_a(t), x_b(t))$ and $\phi_b(s) = \phi_b(s, x_a(t), x_b(t))$. From Lemma 1, we have shown that $\frac{\partial \mathcal{E}}{\partial \lambda} < 0$, $\forall u(t)$ defined by an ergodic policy. As a result, as $t \rightarrow \infty$, both agents are asymptotically optimally ergodic with respect to their own target distributions so long as each action reduces the ergodic objective. Therefore, we can write the change in the utility function—which we define as the ergodic metric—as

$$\begin{aligned} \mathcal{U}_i(x(t) | u) - \mathcal{U}_i(x(t) | u_{-i}) &= \Delta U_i \approx \frac{\partial \mathcal{E}_i}{\partial \lambda} \lambda < 0 \\ \mathcal{U}_i(x(t) | u) - \mathcal{U}_i(x(t) | u_{-i}) &< 0 \\ \mathcal{U}_i(x(t) | u) &< \mathcal{U}_i(x(t) | u_{-i}) \end{aligned}$$

Thus, an ergodic control strategy is a Nash equilibrium strategy. ■

This kind of analysis lends some insight towards formally viewing ergodic policies with respect to game theory and with application in general multi-agent games.

In the following section, we provide examples for typical uses of our proposed method for multi-agent area coverage problems and a comparison with an area coverage in corridors and tracking a time-varying distribution.

V. ERGODIC AREA COVERAGE FOR MULTI-AGENT ELEVATION MAPPING

In this section we illustrate the capabilities of a decentralized ergodic controller for multi-agent area coverage for elevation mapping. We use this example to show improved area coverage of a decentralized ergodic controller while comparing with a centralized controller for the same task.

A. Problem Setup

A 12 dimensional quadrotor [31] is used for the robotic agent dynamics with 4 inputs directly controlling thrust, yaw, pitch, and roll angular accelerations. Each agent measures ground elevation relative to the agent's altitude which it uses to construct a model of the terrain. Three agents are

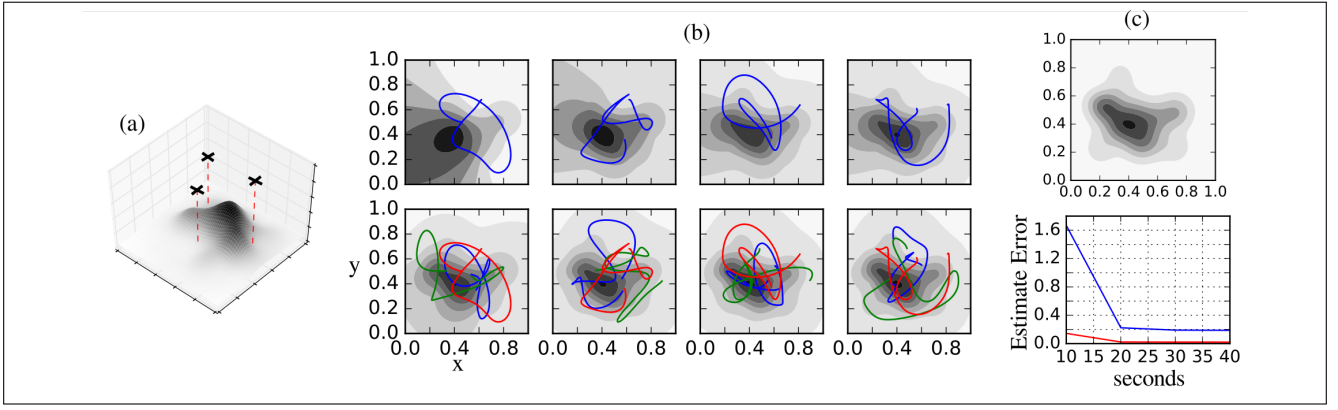


Fig. 1. (a) Three quadcopter agents are depicted in the map with the terrain. The red dashed line indicates the location. (b) Trajectories of a single agent (blue) and a multi-agent system (blue, red, green) are shown estimating terrain. The terrain map is obtained from height measurements (dark regions represent high elevation). (c) Top-down orthographic view of the terrain map for comparison with the results in (b). Error of the estimate is shown at 10 second intervals of collected data. Our algorithm is able to coordinate the agents such that more area is covered, enabling the collection of more data and a resulting better terrain map.

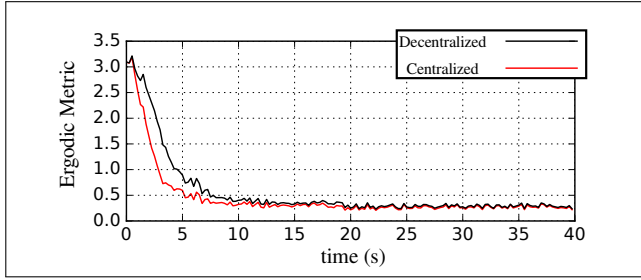


Fig. 2. Comparison of ergodic metric for a decentralized ergodic scheme versus a centralized ergodic scheme. Due to communication and consensus amongst the agent in a decentralized scheme, the ergodic metric does not reduce as quickly as a centralized scheme would. However, the decentralized scheme is quick to reach consensus and performs comparably to the centralized scheme.

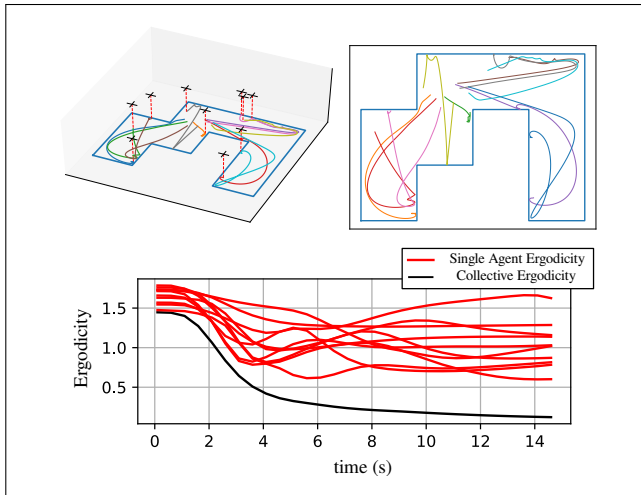


Fig. 3. Comparison of area coverage in a corridor defined in [17] with small (point-wise) visibility for each agent. Ergodic control allows the agents to compensate for the limited sensing through motion while coordinating the decentralized agents to minimize the area coverage objective.

used that are fully connected to one another. The agents are randomly initialized and a Gaussian Process [32], [33] is used to construct the terrain elevation from data collected after 10 second intervals.

B. Results

Figure 1 illustrates the algorithm for area coverage using a network of three decentralized robotic agents. For comparison purposes, the area coverage of a single agent under the ergodic control policy is shown. Due to sharing where each agent intends to go and where they have been, the outcome is a more efficient search as each agent chooses the best possible action that reduces the ergodic metric. The ergodic control automatically takes into account dynamic constraints and the histories of the other agents in order to allocate where each agent should go in a decentralized fashion. We see this in Fig. 1(c) where the multi-agent system immediately acquires a good terrain model within the first ten seconds according to the error norm on the estimate compared to what the single agent could be capable of accomplishing.

A comparison is presented in Fig. 2 with respect to the centralized formulation of the algorithm. Not much performance is lost within the first 5 seconds of the algorithm when the robotic agents are still trying to achieve a consensus. After each agent has fulfilled consensus, the decentralized ergodic policy functions minimize the ergodic metric comparably to the centralized version of the algorithm.

We further compare our algorithm with the work done in [17]. In [17], the algorithm uses a visibility constraint which determined the location of the robot with linear dynamics based on the corridor. We compare to this method using a very small visibility (only the point below the quadcopter) using the decentralized ergodic control scheme. We present the area coverage problem in Fig. 3 where we show the corridor used in [17] for area coverage using agents with nonlinear dynamics (quadcopter dynamics defined previously). The initial positions of the agents were placed as closely as possible to [17]. Since the visibility constraint is significantly small, this would require the robot to move in order to

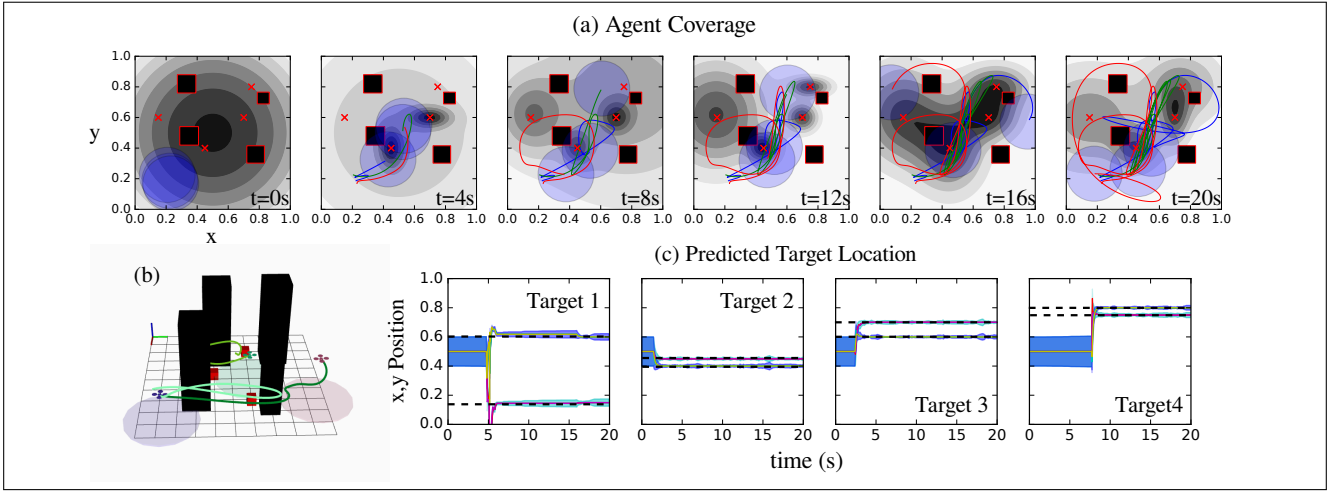


Fig. 4. Target localization is illustrated using 3 agents with deterministic quadcopter dynamics and 4 unknown stationary targets depicted as the red crosses. (a) Area coverage of the quadcopter is shown as the lines associated with each quadcopter at distinct times. The density function underneath shows the likely locations of the targets with darker regions indicating higher likelihood values. Obstacles are shown as black squares with red outlines. A 3D rendering of the environment is shown in (b). Each agent only has a field of range of 0.36 meters as shown as the transparent blue circles. Within 10 seconds the agents under decentralized ergodic policies are able to provide a consensus on the location of the targets. (c) Extended Kalman filter values on the location of the target is shown. The black dashed line is the target location ground truth. We refer the reader to the multimedia <https://youtu.be/Jibt4GLj5sw> for more examples of target localization with a moving target.

sufficiently cover the area. As a result, the work in [17] would not be appropriate in a situation where the dynamics of the robot are needed to compensate for the sensor inefficiencies. In contrast, our method compensates for the small visibility with motion as shown with the trajectories in Fig. 3.

We note in Fig. 3 that the individual agents' respective ergodicity measures do poorly, whereas the ergodicity measure of the whole system does well. This illustrates the efficacy of our method to coordinate the decentralized network to successfully minimize the ergodic objective.

VI. DECENTRALIZED ERGODIC CONTROL FOR MULTI-AGENT TARGET LOCALIZATION

In this section, decentralized ergodic control is used for multi-agent target localization. We use the example of multi-agent target localization because this platform provides us with novel demonstration of the decentralized ergodic control algorithm through a well known robotics problem.

A. Problem Setup

The goal of target localization is to have the agents locate the target (or targets) in the environment. Bearing only sensors [7], [34] are used for sensing the target with the same three agents as mentioned in Section V with quadcopter dynamics. The obstacles are incorporated into the objective with an obstacle avoidance cost which we define by the function $\Theta(x) : \mathbb{R}^n \rightarrow \mathbb{R}^+$ which is a direct penalty if the agent goes near an obstacle. In addition, we constrain the radius of the target sensor to 0.38 meter diameter, thus limiting the total area coverage from the sensor. The targets are uniformly dispersed throughout the terrain of size $[0, 1] \times [0, 1]$ ⁸ such that they do not intersect with the obstacles. Targets

⁸This can be easily adjusted in experimentation if the terrain is much larger.

are localized using an extended Kalman filter (EKF) [35], [36] with sensor noise assumed to be zero mean Gaussian with variance $\sigma^2 = 0.01$. The ergodic controller is initialized with a uniform target distribution. The prior on the targets is initialized as uniform over the search terrain and a distributed EKF updates the prior for the network system [37]. The target distribution is given by the expected information density [9], [7]

$$\phi(s) = \eta \det \left[\int_{\theta} \frac{\partial (\Upsilon(\theta, s))}{\partial \theta}^T \Sigma^{-1} \frac{\partial (\Upsilon(\theta, s))}{\partial \theta} p(\theta) d\theta \right]$$

where η is a normalization factor, $\Upsilon(\theta, s)$ is the bearing only measurement model parametrized by the position of the targets θ .

B. Results

Figure 4 illustrates trajectories of the 3 agents localizing 4 targets in the environment. In Fig. 4 (a), each agent chooses a different path that reduces the ergodic measure as well as increases the area coverage. The agents each localize the targets within the first 10 seconds (as shown in Fig. 4(c)) while successfully avoiding obstacles (illustrated as the black colored squares). Here, each agent is solving their own local control problem and only communicating the respective agent's c_k values to the neighboring agents. The resulting estimate error is within 0.001 as specified by the Kalman filter and the measurement noise (zero mean Gaussian noise with variance $\sigma^2 = 0.01$).

We provide an additional example with a moving target in the attached multimedia <https://youtu.be/Jibt4GLj5sw>. In this example, we use a particle filter to track the position of the target. Note that each agent does not share the particle filter information with one another. Instead, the ϕ_k values are also communicated which results in the agents converging over the

single target. As a result, the agents are able to hone in on the target even in the presence of obstacles.

VII. CONCLUSIONS

We present a fully decentralized formulation of ergodic control for multi-agent systems with nonlinear dynamics. A game theoretic analysis of the algorithm is provided showing the capabilities that the algorithm has on multi-agent games. Examples of area coverage and target localization illustrate the flexibility of the algorithm for various multi-agent coordination tasks with nonlinear dynamics. This opens up the possibilities of adapting other multi-agent objectives—such as pursuit-evasion games—into a decentralized network.

REFERENCES

- [1] D. Carmel and S. Markovitch, "Exploration strategies for model-based learning in multi-agent systems: Exploration strategies," *Autonomous Agents and Multi-agent systems*, vol. 2, no. 2, pp. 141–172, 1999.
- [2] G. Dudek, M. R. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.
- [3] C. Manss, D. Shutin, T. Wiedemann, A. Viseras, and J. Mueller, "Decentralized multi-agent entropy-driven exploration under sparsity constraints," in *Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*, 2016, pp. 143–147.
- [4] A. Viseras Ruiz, M. Angermann, I. Wieser, M. Frassl, and J. Mueller, "Efficient multi-agent exploration with Gaussian processes," 2014.
- [5] M. A. Khamis and W. Goma, "Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 134–151, 2014.
- [6] F. Pei, M. Wu, and S. Zhang, "Distributed SLAM using improved particle filter for mobile robot localization," *The Scientific World Journal*, 2014.
- [7] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, and T. D. Murphey, "Real-time area coverage and target localization using receding-horizon ergodic exploration," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 62–80, 2018.
- [8] L. M. Miller and T. D. Murphey, "Trajectory optimization for continuous ergodic exploration," in *American Control Conference*, 2013, pp. 4196–4201.
- [9] L. M. Miller, Y. Silverman, M. A. MacIver, and T. D. Murphey, "Ergodic exploration of distributed information," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 36–52, 2016.
- [10] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4, pp. 432–442, 2011.
- [11] D. A. Shell and M. J. Mataric, "Ergodic dynamics for large-scale distributed robot systems," in *International Conference on Unconventional Computation*. Springer, 2006, pp. 254–266.
- [12] I. Abraham, A. Prabhakar, M. J. Hartmann, and T. D. Murphey, "Ergodic exploration using binary sensing for nonparametric shape estimation," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 827–834, 2017.
- [13] L. M. Miller and T. D. Murphey, "Optimal planning for target localization and coverage using range sensing," in *IEEE International Conference on Automation Science and Engineering*, 2015, pp. 501–508.
- [14] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [15] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot control using time-varying density functions," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 489–493, 2015.
- [16] S. Miah, A. Y. Panah, M. M. H. Fallah, and D. Spinello, "Generalized non-autonomous metric optimization for area coverage problems with mobile autonomous agents," *Automatica*, vol. 80, pp. 295–299, 2017.
- [17] Y. Kantaros, M. Thanou, and A. Tzes, "Distributed coverage control for concave areas by a heterogeneous robot-swarm with visibility sensing constraints," *Automatica*, vol. 53, pp. 195–207, 2015.
- [18] J. Vander Hook, P. Tokekar, and V. Isler, "Algorithms for cooperative active localization of static targets with mobile bearing sensors under communication constraints," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 864–876, 2015.
- [19] C. Freundlich, S. Lee, and M. M. Zavlanos, "Distributed active state estimation with user-specified accuracy," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 418–433, 2018.
- [20] A. Dirafzoon, M. Bagher Menhaj, and A. Afshar, "Decentralized coverage control for multi-agent systems with nonlinear dynamics," vol. 94-D, pp. 3–10, 01 2011.
- [21] N. Deo, *Graph theory with applications to engineering and computer science*. Courier Dover Publications, 2016.
- [22] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.
- [23] A. R. Ansari and T. D. Murphey, "Sequential action control: Closed-form optimal control for nonlinear and nonsmooth systems," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1196–1214, 2016.
- [24] R. Vasudevan, H. Gonzalez, R. Bajcsy, and S. S. Sastry, "Consistent approximations for the optimal control of constrained switched systems—part 1: A conceptual algorithm," *SIAM Journal on Control and Optimization*, vol. 51, no. 6, pp. 4463–4483, 2013.
- [25] H. Axelsson, Y. Wardi, M. Egerstedt, and E. Verriest, "Gradient descent approach to optimal mode scheduling in hybrid dynamical systems," *Journal of Optimization Theory and Applications*, vol. 136, no. 2, pp. 167–186, 2008.
- [26] M. Egerstedt, Y. Wardi, and H. Axelsson, "Transition-time optimization for switched-mode dynamical systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 110–115, 2006.
- [27] T. Caldwell and T. Murphey, "Projection-based iterative mode scheduling for switched systems," *Nonlinear Analysis: Hybrid Systems*, vol. 21, pp. 59–83, 2016.
- [28] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.
- [29] S. Bhattacharya and S. Hutchinson, "On the existence of Nash equilibrium for a two player pursuit-evasion game with visibility constraints," in *Algorithmic Foundation of Robotics VIII*, 2009, pp. 251–265.
- [30] R. B. Myerson, "Refinements of the Nash equilibrium concept," *International Journal of Game Theory*, vol. 7, no. 2, pp. 73–80, 1978.
- [31] P. Martin and E. Salaün, "The true role of accelerometer feedback in quadrotor control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 1623–1629.
- [32] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009.
- [33] C. Plagemann, S. Mischke, S. Prentice, K. Kersting, N. Roy, and W. Burgard, "Learning predictive terrain models for legged robot locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 3545–3552.
- [34] M. Deans and M. Hebert, "Experimental comparison of techniques for localization and mapping using a bearing-only sensor," in *Experimental Robotics VII*, 2001, pp. 395–404.
- [35] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [36] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97*. International Society for Optics and Photonics, 1997, pp. 182–193.
- [37] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed kalman filtering based on consensus strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, 2008.