Preprints of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, July 3-6, 2012

Animal-Inspired Optimal Foraging via a Distributed Actor-Critic Algorithm *

Ioannis Ch. Paschalidis† and Yingwei Lin‡

Abstract— **We consider a group of mobile agents operating in a mission space that collaborate to solve a general dynamic optimization problem. The agents seek to maximize the total reward collected and minimize their energy cost. We construct a control policy specifying how the agents move subject to constraints due to the geometry of the mission space, the presence of obstacles, their sensing range, their available energy, and the need to avoid collisions with other agents. The mission space is discretized and modeled as a graph. We apply the distributed actor-critic method from [1] properly modified to benefit from least squares temporal difference learning. We present one concrete application: air vehicles flying below a forest's canopy. We demonstrate that the incorporation of bio-inspired features such as eavesdropping, spatial memory, directional sensing, and grouping into the control policy significantly improves its performance compared to a policy from [2] that included no such features.**

I. Introduction

The work in this paper is motivated by coverage control problems in complex mission spaces using multiple agents. Such problems have received quite a bit of attention, see [3], [4] and references therein. This earlier work formulated coverage control as a static optimization problem and developed distributed ("on-line") methods for solving it. Here we adopt a different and more ambitious perspective. We formulate the problem as a dynamic optimization problem and seek a flexible *control policy* that specifies how the agents move in the mission space to optimize a long-term average performance metric.

A particularly motivating application concerns air vehicles flying under a forest's canopy. They need to avoid tree trunks and each other while visiting certain (potentially moving) targets to collect rewards. From time to time, the air vehicles need to return to a base station for unloading rewards and refueling.

This scenario is not unlike what animals do while hunting for food. Flying animals, like bats, have extraordinary skills in collaborative emergence and foraging. [5]–[9] captured these biological behaviors using modern sensory technologies. Motivated by these observations, in this work we incorporate a number of bio-inspired features into our control policy, including eavesdropping, spatial memory, directional sensing, and grouping. We demonstrate that such features significantly improve performance; by as much as 40% in some examples. It is important to note the flexibility of our control policy that allows us to incorporate these bio-inspired features.

Technically, our problem is cast into a *Markov Decision Process (MDP)* framework. In our setting, the MDP is to be cooperatively solved by multiple agents that can simultaneously explore the state-control space. Each agent communicates and exchanges information with agents in its vicinity and uses this information to affect its own policy.

A general strategy for solving large-scale MDPs using so-called *Actor-Critic (AC) algorithms* was presented in [10]. Such algorithms optimize over a randomized class of policies which are parameterized by a (low-dimensional) parameter vector. An AC algorithm optimizes policy performance with respect to the parameter vector by using a simulation (or a realization) of the MDP. According to its name, the algorithm interleaves two steps: (i) the actor, which amounts to a policy improvement step that descends along the performance gradient with respect to the parameter vector, and (ii) the critic, which is a policy evaluation step at which the algorithm learns an approximate value function from a sample path that uses the current policy. [11] adopts the least squares temporal difference learning in actor-critic algorithms, and improves the convergence rate.

[1] introduced a *Distributed Multi-agent Actor-Critic (D-AC)* algorithm which allowed multiple agents to simultaneously explore the state-control space. Each agent maintained its own policy parameter and updated it based on local information and information received from a subset of other agents. This updating followed a consensus-like algorithm and under suitable conditions, all agents reached consensus and converged to the same parameter vector. In [2] we considered a similar coverage problem as the one we study in this paper and modeled the mission space as a general graph whose nodes correspond to all possible positions of the agents.

[2] provided a parametric class of policies and demonstrated how to tune the policy parameters to enable efficient reward collection. It did not, however, considered bio-inspired features as we do in this paper. Furthermore, in this work we also provide an analysis of the "obstacle density" of the mission space by estimating the number of accessible target locations. This is useful and can be used as an indicator of whether reward collection can be effective.

The rest of the paper is organized as follows. Sec. II introduces some preliminaries. Sec. III formulates the reward

* Research partially supported by the NSF under grant EFRI-0735974, by the DOE under grant DE-FG52-06NA27490, by the ARO under grant W911NF-11-1-0227 and by the ODDR&E MURI10 program under grant N00014-10-1-0952.

† Corresponding author. Dept. of Electrical & Computer Eng., and Division of Systems Eng., Boston University, 8 St. Mary's St., Boston, MA 02215, e-mail: yannisp@bu.edu, url: http://ionia.bu.edu/.

‡ Center for Information & Systems Eng., Boston University, e-mail: yingwei@bu.edu.

978-1-4673-2529-5/12/$31.00 ©2012 IEEE

collection problem. Sec. IV proposes a parametric class of bio-inspired policies. Sec. V discusses the air vehicle application using the bio-inspired control policy to collect rewards. Sec. VI contains our "obstacle density" analysis and assesses the effectiveness of the bio-inspired features. Conclusions are in Sec. VII.

II. Preliminaries

Consider a Markov decision process with finite state and action spaces $\mathscr{X}$ and $\mathscr{U}$, respectively. Let $c : \mathscr{X} \times \mathscr{U} \mapsto \mathbb{R}$ be a reward function. Let $\{\mu_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \mathbb{R}^n\}$ be a set of *randomized stationary policies (RSPs)*, parametrized by $\boldsymbol{\theta}$. In particular, $\mu_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})$ denotes the probability of taking the action $\mathbf{u}$ given the state $\mathbf{x}$, under the RSP $\boldsymbol{\theta}$. For every $\boldsymbol{\theta}$, the Markov chains $\{\mathbf{x}_k\}$ and $\{(\mathbf{x}_k, \mathbf{u}_k)\}$ are irreducible and aperiodic, with stationary probabilities $\pi_{\boldsymbol{\theta}}(\mathbf{x})$ and $\eta_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}) = \pi_{\boldsymbol{\theta}}(\mathbf{x})\mu_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x})$, respectively.

We are interested in finding a $\boldsymbol{\theta}$ that maximizes the average reward function:

$$\overline{\alpha}(\boldsymbol{\theta}) = \sum_{\mathbf{x}\in\mathscr{X},\mathbf{u}\in\mathscr{U}} c(\mathbf{x}, \mathbf{u})\eta_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}). \quad (1)$$

For each $\boldsymbol{\theta}$ define a differential reward function $V_{\boldsymbol{\theta}} : \mathscr{X} \mapsto \mathbb{R}$, as solution of the following Poisson equation:

$$\overline{\alpha}(\boldsymbol{\theta}) + V_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{\mathbf{u}\in\mathscr{U}} \mu_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}) \Big[c(\mathbf{x}, \mathbf{u}) + \sum_{\mathbf{y}\in\mathscr{X}} p(\mathbf{y}|\mathbf{x}, \mathbf{u})V_{\boldsymbol{\theta}}(\mathbf{y}) \Big], \quad (2)$$

where $p(\mathbf{y}|\mathbf{x}, \mathbf{u})$ is the probability that the next state is $\mathbf{y}$ given that the current state is $\mathbf{x}$ and action $\mathbf{u}$ is taken. $V_{\boldsymbol{\theta}}(\mathbf{x})$ can be interpreted as the relative reward of starting at state $\mathbf{x}$, that is, the excess reward we collect on top of the average reward if we start at $\mathbf{x}$. Define the Q-value function:

$$Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}) = c(\mathbf{x}, \mathbf{u}) - \overline{\alpha}(\boldsymbol{\theta}) + \textstyle\sum_{\mathbf{y}\in\mathscr{X}} p(\mathbf{y}|\mathbf{x}, \mathbf{u})V_{\boldsymbol{\theta}}(\mathbf{y}). \quad (3)$$

The following result is from [12] where for the components of $\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$ we write $(\psi_{\boldsymbol{\theta},1}(\mathbf{x}, \mathbf{u}), \ldots, \psi_{\boldsymbol{\theta},n}(\mathbf{x}, \mathbf{u}))$.

Theorem II.1 (Average Reward Gradient) *We have*

$$\nabla\overline{\alpha}(\boldsymbol{\theta}) = \sum_{\mathbf{x}\in\mathscr{X},\mathbf{u}\in\mathscr{U}} \eta_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}), \quad (4)$$

where

$$\boldsymbol{\psi}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}) = \nabla_{\boldsymbol{\theta}} \ln \mu_{\boldsymbol{\theta}}(\mathbf{u}|\mathbf{x}). \quad (5)$$

The actor-critic algorithm works with a parametrization of the Q-function in terms of a vector $\mathbf{r} = (r_1, \ldots, r_m) \in \mathbb{R}^m$:

$$Q^{\mathbf{r}}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}) = \textstyle\sum_{l=1}^{m} r_l\phi_{\boldsymbol{\theta},l}(\mathbf{x}, \mathbf{u}).$$

A typical choice for the features $\phi_{\boldsymbol{\theta},l}(\mathbf{x}, \mathbf{u})$ is to set $m = n + 1$, $\phi_{\boldsymbol{\theta},l}(\mathbf{x}, \mathbf{u}) = \psi_{\boldsymbol{\theta},l}(\mathbf{x}, \mathbf{u})$ for $l = 1, \ldots, n$, and fix $\phi_{\boldsymbol{\theta},n+1}(\mathbf{x}, \mathbf{u})$ to the constant function that is everywhere equal to one except at some special point, say $\mathbf{x} = \mathbf{0}$, where $\phi_{\boldsymbol{\theta},n+1}(\mathbf{0}, \mathbf{u}) = 0$ for all $\mathbf{u}$. The critic estimates the parameter $\mathbf{r}$ on the basis of observations from a sample path of the Markov process while the actor uses $\mathbf{r}$ to compute the performance gradient and to update $\boldsymbol{\theta}$. A distributed actor-critic algorithm using Least Squares Differences Learning (LSTD) learning has been introduced in [2]. It has been proved that the algorithm converges to a policy with a parameter θ that is locally optimal.

III. A reward collection mission

Consider now a mission space and partition it into a set of regions. We do not know agent positions exactly but we have enough information to know the region each agent can be found. This, for instance, can be achieved with localization systems such as the ones in [13], [14].

[1] formulated the mission space as a 2D grid, which is suitable for agents moving in an obstacle-free flat environment. However, when it comes to real world scenarios in which the terrains are usually not flat and contain obstacles such as trees and mountains, we need to have a more general formulation of the mission space.

[2] modeled the mission space as a undirected graph $\mathscr{G}(\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ is the set of regions. Let v^* correspond to a special region which we call the *base station*. $\mathscr{E}$ is the set of links that specify connectivity between the regions. The existence of a link (i, j) indicates that an agent can directly move from region i to region j and vice versa, i.e., these regions are "adjacent."

Let now $\mathscr{T} = \{t_1, \ldots, t_D\}$ be a set of D targets. The position (region) of target t_i at time k is indicated by $v_k^{t_i} \in \mathscr{V}$. We assume the targets are always moving in the mission space and they are at some region at any time. Each target is associated with a certain reward. The reward of target t_i at time k is indicated by $\Phi_k^{t_i} \geq 0$.

The mission space is explored by N mobile agents, $a_1, \ldots, a_N$. The position (region) of agent a_j at time k is indicated by $v_k^{a_j} \in V$. We assume the initial position of agent a_j to be $v_0^{a_j} = v^*$, for all $j = 1, \ldots, N$, which means they all start from the base station. To each agent a_j we associate a capacity whose value at time k is denoted $w_k^{a_j} \geq 0$. The maximum speed of agent a_j is denoted by ρ^{a_j}. For a discretized 3-dimensional (3D) mission space, we define the flying direction of agent a_j at time k as $d_k^{a_j} = (v_{k+1}^{a_j} - v_k^{a_j})/\|v_{k+1}^{a_j} - v_k^{a_j}\|$. The direction from agent a_j to target t_i at time k is $d_k^{ij} = (v_k^{a_j} - v_k^{t_i})/\|v_k^{a_j} - v_k^{t_i}\|$. We model the one step energy cost of agent a_j as $E_k^{a_j} = \|v_{k+1}^{a_j} - v_k^{a_j}\| + \|d_k^{a_j} - d_{k-1}^{a_j}\|$; it accounts for both travel distance and directional change.

When an agent visits a target point, it collects a reward which depends on the available reward at the target point and the capacity of the agent. Every visit has also the effect of depleting a part of the agent's capacity. The capacity will also decrease over time, because flying consumes energy. The agents can return to the base station to replenish their capacities to its initial value. The dynamics of the sequences $\{\Phi_k^{t_i}\}$ and $\{w_k^{a_j}\}$ are described as follows.

For all targets $i = 1, \ldots, D$,

- $\Phi_{k+1}^{t_i} = \max(\Phi_k^{t_i} - w_k^{a_j}, 0)$, if $\exists$ agent j such that $v_k^{t_i} = v_k^{a_j}$.

- $\Phi_{k+1}^{t_i} = \Phi_k^{t_i} + \omega_k$, otherwise, where $\{\omega_k\}$ is a sequence of i.i.d. random variables.

For all agents $j = 1, \ldots, N$,

- $w_{k+1}^{a_j} = \max(w_k^{a_j} - \Phi_k^{t_i}, 0)$, if $\exists$ target i such that $v_k^{t_i} = v_k^{a_j}$.
- $w_{k+1}^{a_j} = w_0^{a_j}$, if $v_k^{a_j} = v^*$.
- $w_{k+1}^{a_j} = w_k^{a_j} - C + g_k$, otherwise, where C is a constant, and $\{g_k\}$ is also a sequence of i.i.d. random variables.

The sequence of rewards, $\Phi_k^{a_j}$, collected by each agent j over time is characterized as $\Phi_k^{a_j} = \min(\Phi_k^{t_i}, w_k^{a_j})$, if $\exists$ target i such that $v_k^{t_i} = v_k^{a_j}$, and 0 otherwise. That is, agents when in the same region with a target collect as much reward available at the target and allowed by their capacity.

As each agent roams in the mission space, it is able to sense the presence of targets with rewards in neighboring regions within some fixed sensing range. Since our mission space is modeled as a connected graph, we model the sensing range as the maximum hop count from the region of the agent to the region of the target. We denote the sensing range for agent a_j as $\delta^{a_j} \in \mathbb{Z}_+$. We formally assume that agent a_j can detect target t_i at time k if and only if $\text{dist}(v_k^{t_i}, v_k^{a_j}) \leq \delta^{a_j}$, where the function $\text{dist}(x, y)$ is the shortest path function that returns the number of edges on the shortest path from node x to node y.

The agent a_j who detects a target t_i can also compute a metric which increases with the reward available at the target and decreases with the distance from the target. We refer to this metric as "signal strength" from target t_i detected by agent a_j positioned at $v_k^{a_j}$ and denote it by $s_k^{t_i}(v_k^{a_j})$ at time k. We assume:

$$s_k^{t_i}(v_k^{a_j}) = \begin{cases} \frac{\Phi_k^{t_i}}{Z} e^{-\text{dist}(v_k^{a_j}, v_k^{t_i})}, & \text{if } \text{dist}(v_k^{a_j}, v_k^{t_i}) \leq \delta^{a_j}, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

where Z is a positive normalizing constant. We will use this signal strength later in constructing a policy that decides where each agent should be headed.

IV. A Parametric Class of Policies

Let us now model the energy gain for agent a_j at time k as: $\Omega_k^{a_j} = \Phi_k^{a_j} - E_k^{a_j}$. Given this setup we are interested in a policy that guides the agents in the mission space to maximize the long-term average total energy gain given by

$$\lim_{k \to \infty} \frac{1}{k} \sum_{\tau=1}^{k} \sum_{j=1}^{N} \Omega_\tau^{a_j}. \tag{7}$$

We note that if the agents get close to each other, the chance of collision is higher. As a result, we are also interested in a policy that can keep the agents apart so that they avoid collision. Such a policy should also "distribute" them evenly within the mission space, which leads to better exploration and reward collection. To that end, we introduce the following pair-wise agent "dispersion" metric:

$$\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \text{dist}(v_k^{a_i}, v_k^{a_j}). \tag{8}$$

The policy for each agent is based on its current position, the signals it measures from all target points, and, potentially, any information it receives from other agents. At time k the jth agent can choose a control action u in the set $\mathscr{U}_k^{a_j} = \{v_k^{a_j}\} \cup \{v | \text{dist}(v, v_k^{a_j}) \leq \rho^{a_j}\}$, The agent moves so that $v_{k+1}^{a_j} = u$. Note that staying at the current position is always one of the available options.

We consider the following class of RSPs where each agent a_j at time k and position $v_k^{a_j}$ selects control $u \in \mathscr{U}_k^{a_j}$ with probability

$$\mu(u | v_k^{a_j}) = \frac{e^{(\xi_{\boldsymbol{\theta}^j}(u))}}{\sum_{v \in \mathscr{U}_k^{a_j}} e^{(\xi_{\boldsymbol{\theta}^j}(v))}}, \tag{9}$$

where

$$\xi_{\boldsymbol{\theta}^j}(u) = \sum_{i=1}^{D} \theta_i^j w_k^{a_j} s_k^{t_i}(u) + \theta_0^j e^{-\text{dist}(u, v^*)}, \tag{10}$$

and where $\boldsymbol{\theta}^j = (\theta_0^j, \ldots, \theta_D^j)$. The vector $\boldsymbol{\theta}^j$ parameterizes agent j's policy. This policy favors control actions that lead to targets emitting stronger signals. When the agent's capacity or the available rewards are low then the policy favors control actions that tend to bring the agent closer to the base station where it can replenish its capacity.

V. An Application: Bio-inspired Reward Collection in a 3D mission space

Fig. 1. Dynamically controlled air vehicles (blue bats) collect rewards (green apples) in a 3D mission space. The size of the targets is proportional to the reward of the targets. The trees are obstacles that bats can not fly into, the base station (cave) is for replenishing bats' capacities.

In [2] we have introduced an application of the general reward collection problem where air vehicles (the agents) fly in an occluded terrain (e.g., under a forest's canopy). Next we significantly expand the setting of [2] and investigate a new policy structure that incorporates bio-inspired features.

The mission space graph is defined by discretizing the 3-dimensional (3D) environment using a regular grid, and then linking adjacent locations. Without loss of generality, the maximum speed is set to 1, the

vehicles can move on the grid, and at time k the jth vehicle can choose a control action $\mathbf{u}_k^{a_j} \in \mathscr{U}_k^{a_j} \triangleq \{(1,0,0),(-1,0,0),(0,1,0),(0,-1,0),(0,0,1),(0,0,-1),(0,0,0)\}$ to move from grid position $\mathbf{x}_k^{a_j}$ to the position $\mathbf{x}_{k+1}^{a_j} = \mathbf{x}_k^{a_j} + \mathbf{u}_k^{a_j}$; see Fig. 1. The following subsections consider various bio-inspired features.

1) Avoid Other agents: The air vehicles are likely to collide with each other when they are too close. In order to reduce the chance of collision, we want to evenly distribute them in the mission space. In addition, spreading out the agents in a mission space with sparse targets can result into better target detection. The policy needs to be modified to incorporate this requirement. First, we enrich the policy structure and consider terms in (10) corresponding to each target and each agent. Let $\boldsymbol{\theta}^j = (\theta_0^j, \ldots, \theta_D^j, \theta_{D+1}^j, \ldots, \theta_{D+N}^j)$. The policy for agent a_j given its position $\mathbf{x}_k^{a_j}$ has similar structure as in (9):

$$\mu(\mathbf{u}|\mathbf{x}_k^{a_j}) = \frac{e^{(\xi_{\boldsymbol{\theta}^j}(\mathbf{u},\mathbf{x}_k^{a_j}))}}{\sum_{\mathbf{v}\in\mathscr{U}_k^{a_j}} e^{(\xi_{\boldsymbol{\theta}^j}(\mathbf{v},\mathbf{x}_k^{a_j}))}}, \tag{11}$$

where

$$\xi_{\boldsymbol{\theta}^j}(\mathbf{u},\mathbf{x}) = \sum_{i=1}^{D} \theta_i^j w_k^{a_j} s_k^{t_i}(\mathbf{x}+\mathbf{u}) + \theta_0^j e^{-\text{dist}(\mathbf{x}+\mathbf{u},v^*)} - \sum_{\{i|\text{dist}(\mathbf{x},v_k^{a_i})\le\delta^{a_j}\}} \theta_{D+i}^j e^{-\text{dist}(\mathbf{x}+\mathbf{u},v_k^{a_i})}. \tag{12}$$

Note that the last term of the above penalizes controls that bring a vehicle too close to the others. Although it is not possible to guarantee that the agents will not move to the same location (they do not communicate the movement decision they are going to take in the next iteration), simulation results can show that the average distance increases as we increase the sensing range.

2) Eavesdropping targets from other agents: While agents are trying to keep away from each other to avoid collision, [8] discovered that bats are attracted by playback of echolocation calls produced during prey capture, and bats of the same social unit forage together to benefit from passive information transfer via the change in group members' echolocation calls upon finding prey. It is therefore desirable to move close to the agents who sense the targets. To that end, we use the policy structure in (11) where

$$\xi_{\boldsymbol{\theta}^j}(\mathbf{u},\mathbf{x}) = \sum_{i=1}^{D} \theta_i^j w_k^{a_j} s_k^{t_i}(\mathbf{x}+\mathbf{u}) + \theta_0^j e^{-\text{dist}(\mathbf{x}+\mathbf{u},v^*)} - \sum_{\{i|\text{dist}(\mathbf{x},v_k^{a_i})\le\delta^{a_j},\text{dist}(v_k^{a_i},v_k^{t_l})>\delta^{a_i},\forall l\in\{1,\ldots,D\}\}} \theta_{D+i}^j e^{-\text{dist}(\mathbf{x}+\mathbf{u},v_k^{a_i})} + \sum_{\{i|\text{dist}(\mathbf{x},v_k^{a_i})\le\delta^{a_j},\text{dist}(v_k^{a_i},v_k^{t_l})\le\delta^{a_i},\exists l\in\{1,\ldots,D\}\}} \theta_{D+i}^j e^{-\text{dist}(\mathbf{x}+\mathbf{u},v_k^{a_i})}. \tag{13}$$

While the second term penalizes controls that bring an agent too close to others which do not sense any target, the last term favors controls towards agents which sense other targets. Note that agents who detect targets only need to emit "feeding buzzes" and other agents within the sensing range can pick up this passive information and use it for selecting their control.

3) Spatial Memory: Bats have very good spatial memory during foraging. Even when they can not sense the target directly, they intent to fly to the area where they hunted food last time. To incorporate this feature, we assign to agents some short-term memory. For agent a_j, let $m^{a_j} \in \mathbb{Z}_+$ denote its memory period. If agent a_j has ever visited target t_i, the most recent visiting time is $\tau_{ij} \in \mathbb{Z}_+$. If agent a_j has never visited target t_i, $\tau_{ij} = -m^{a_j}$. Similar to the definition of "Signal Strength", we define the "Memory Strength" at agent a_j for target t_i as:

$$M_k^{t_i}(v_k^{a_j}) = \begin{cases} \frac{\Phi_{\tau_{ij}}^{t_i}}{C} e^{-(\text{dist}(v_k^{a_j},v_{\tau_{ij}}^{t_i})+k-\tau_{ij})}, & \text{if } k-\tau_{ij} \\ & < m^{a_j}, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

The "memory strength" is a variable that indicates how well agent a_j remembers target t_i. It is increasing with the target reward, and decreasing with the time lapsed from the most recent visit to that target. At some point the agent will forget the target and the "memory strength" is reduced to 0. Now the agent can use both "signal strength" and "memory strength" to make a decision. We update (13) as:

$$\xi_{\boldsymbol{\theta}^j}(\mathbf{u},\mathbf{x}) = \sum_{i=1}^{D} \theta_i^j w_k^{a_j} (s_k^{t_i}(\mathbf{x}+\mathbf{u}) + M_k^{t_i}(\mathbf{x}+\mathbf{u})) + \theta_0^j e^{-\text{dist}(\mathbf{x}+\mathbf{u},v^*)} - \sum_{\{i|\text{dist}(\mathbf{x},v_k^{a_i})\le\delta^{a_j},\text{dist}(v_k^{a_i},v_k^{t_l})>\delta^{a_i},\forall l\in\{1,\ldots,D\}\}} \theta_{D+i}^j e^{-\text{dist}(\mathbf{x}+\mathbf{u},v_k^{a_i})} + \sum_{\{i|\text{dist}(\mathbf{x},v_k^{a_i})\le\delta^{a_j},\text{dist}(v_k^{a_i},v_k^{t_l})\le\delta^{a_i},\exists l\in\{1,\ldots,D\}\}} \theta_{D+i}^j e^{-\text{dist}(\mathbf{x}+\mathbf{u},v_k^{a_i})}. \tag{15}$$

4) Directional Sensing: Directional sensing is widely adopted by animals like bats. The "signal strength" is related to the direction the agent is flying and the direction to the target. It is larger if the agent is flying along the direction to a target. To incorporate this feature we update the "signal strength" definition as:

$$s_k^{t_i}(v_k^{a_j}) = \begin{cases} \frac{\Phi_k^{t_i}}{C} e^{-(\text{dist}(v_k^{a_j},v_k^{t_i})+\|d_k^{a_j}-d_k^{ij}\|)}, & \text{dist}(v_k^{a_j},v_k^{t_i}) \\ & \le \delta^{a_j}, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

By using directional sensing, agents will favor targets that are on their current flying path, thus, avoiding turning too often and wasting energy associated with changing direction.

5) Long Term Memory / Grouping: Bats as they emerge from the cave, travel a long distance towards a food resource. It is assumed that they have long-term memory and know the approximate location of a good feeding area. It has also been discovered that bats from the same roosts maintain social cohesion by flying within hearing distance of each other [8]. To incorporate such a feature, we can group the

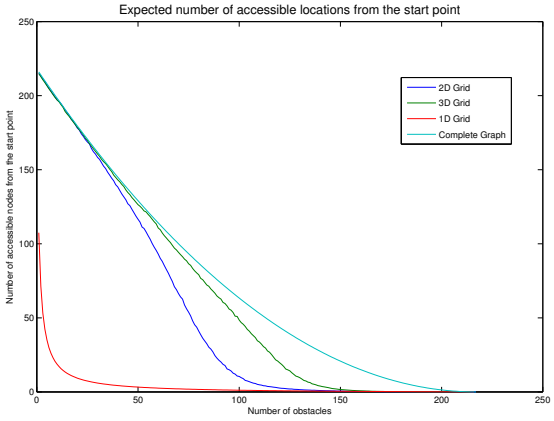

Fig. 2. Expected number of accessible targets in a 2D or 3D mission space estimated by Monte Carlo simulation.

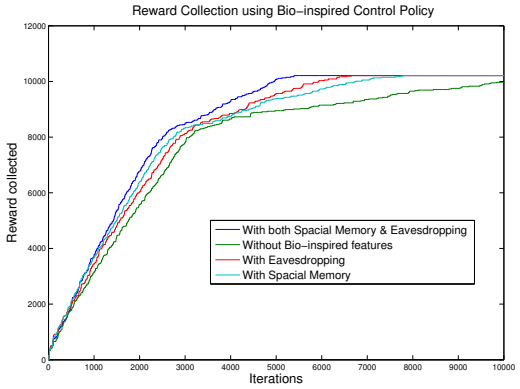

Fig. 3. Reward collection in $10,000$ iterations, showing the advantage of bio-inspired policies.

agents into teams and assign them a specific part of the mission space that is likely to have targets. [1] and [2] have shown faster and more effective reward collection using a number of virtual beacons. The control policy for a team of agents has an extra term that directs them towards a particular beacon.

VI. Performance Analysis and Simulation Results

All agents start exploring the mission space from the base station. However, due to the existence of obstacles, some targets might be blocked and inaccessible from the agents. As a way to assess whether the mission space is overly dense and prevents effective reward collection, we estimate the number of accessible targets from the base station.

Let S denote the number of discrete locations in the mission space, O the number of obstacles which we distribute uniformly within the mission space, T the number of target which are also uniformly distributed, and A the (random) number of accessible target from the start location. We establish the following theorem; we omit the proof due to space limitations.

Theorem VI.1 *For a 2D or a 3D mission space it holds:*

$$\frac{\sum_{i=1}^{S-O+1} i\frac{\binom{O-1}{S-i}}{\binom{O}{S}}}{S}T \leq E[A] \leq \frac{(S-O)^2}{S^2}T. \qquad (17)$$

We used Monte Carlo simulation to estimate the expected number of accessible targets and the results are depicted in Fig. 2. The expected number of accessible targets in a 3D grid mission space is closer to the upper bound than in the 2D grid, because the connectivity is better in 3D so the probability of blocking a location from the start point is lower than in the 2D space. In the 3D grid mission space, when the density of obstacles is less than 25%, most of the obstacle-free locations are reachable from the start point. When, however, the obstacle density is larger than 75%, almost no target is reachable.

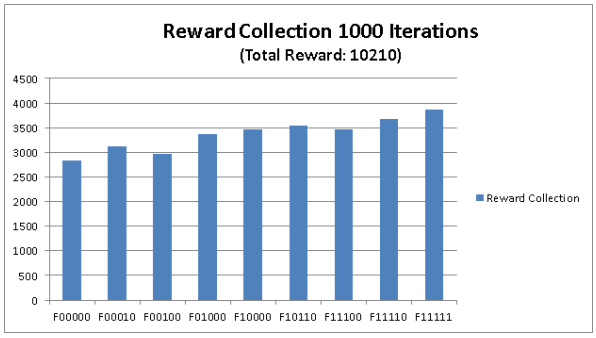

Fig. 4. Reward collected in the first $1,000$ iterations. The policy features are denoted by F followed by 5 bits corresponding to [eavesdropping, avoiding other agents, spatial memory, directional sensing, grouping]. The value is averaged over 100 i.i.d. experiments.

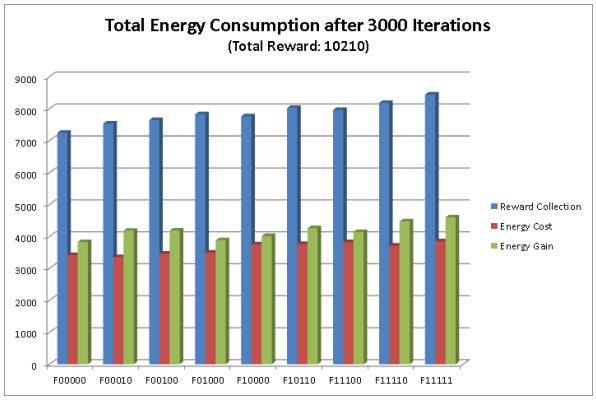

Fig. 5. Energy gained in the first $3,000$ iterations. The value is averaged over 100 i.i.d. experiments.

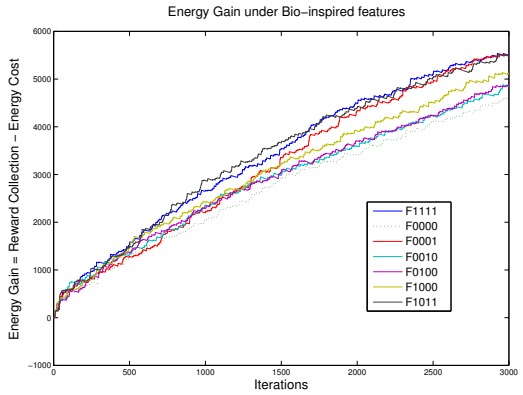

Fig. 6. Energy gained in the first $3,000$ iterations from a typical simulation run.

We simulated our bio-inspired control policy in a $100\times100\times100$ 3D grid mission space. 16 agents have capacities ranging from 20 to 40, speed ranging from 1 to 4, and sensing ranging from 35 to 200. Targets with total reward equal to $10,210$ were randomly distributed in the mission space. 16 vertical trees with height ranging from 10 to 46 were also uniformly distributed. A typical simulation instance for the reward collected over $10,000$ iterations is shown in Fig. 3. By using the bio-inspired features like spatial memory and eavesdropping, the reward collection speeds up as all reward has been collected by the $5,500$th iteration, while the non-bio-inspired policy can not even collect all the reward by the $10,000$th iteration. By incorporating the remaining bio-features such as directional sensing, avoiding other agents, and grouping, we collected about 40% more reward by the $1,000$th iteration (see Fig. 4).

Our results also show that the energy gain is higher using the bio-inspired control policies. Although the energy cost is slightly higher when the bio-inspired features are in effect, this is because there is a trade-off between reward collection and energy cost. Overall, our goal to increase the energy gain is achieved as the result in Figs. 5 and 6 demonstrate. The energy gain is about 20% higher using the bio-inspired control policy.

VII. Conclusions

We developed a class of bio-inspired control policies for solving a general reward collection problem and used a distributed actor-critic algorithm to optimize the policy parameters. The bio-inspired policies have been numerically shown to lead to faster reward collection and increased energy gain rate.

The features we considered, including eavesdropping, spatial memory, directional sensing, avoiding other agents, and grouping are inspired by the foraging behavior of bats. We used our machinery to a setting where a group of air vehicles collect rewards by visiting targets in an occluded terrain, for instance when they fly under a forest's canopy. The eavesdropping feature only requires the agents to monitor a passive signal, while the spatial memory and grouping need some constant size local memory. All these features can be implemented in a distributed fashion and do not substantially increase communications among agents. Directional sensing is a more "native" feature for some agents since some sensors like a camera are indeed directional.

We also established analytical bounds on the expected number of agents reachable from the start location. These bounds help us assess the "obstacle density" of any given mission space.

References

[1] P. Pennesi and I. C. Paschalidis, "A distributed actor-critic algorithm and applications to mobile sensor network coordination problems," *IEEE Trans. Automat. Contr.*, vol. 55, no. 2, pp. 492–497, 2010.

[2] I. C. Paschalidis and Y. Lin, "Mobile agent coordination via a distributed actor-critic algorithm," in *Proceedings of the 19th Mediterranean Conference on Control and Automation (MED 11)*, Corfu, Greece, June 20–23, 2011 2011.

[3] W. Li and C. Cassandras, "Distributed cooperative coverage control," *IEEE Conf. on Decision and Control*, pp. 2542–2547, 2005.

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

[5] N. I. Hristov, M. Betke, and T. H. Kunz, "Applications of thermal infrared imaging for research in aeroecology," *Integrative and Comparative Biology*, vol. 48, no. 1, pp. 50–59, 2008. [Online]. Available: http://icb.oxfordjournals.org/content/48/1/50.abstract

[6] J. W. Horn and T. H. Kunz, "Analyzing nexrad doppler radar images to assess nightly dispersal patterns and population trends in brazilian free-tailed bats (tadarida brasiliensis)," *Integrative and Comparative Biology*, vol. 48, no. 1, pp. 24–39, 2008. [Online]. Available: http://icb.oxfordjournals.org/content/48/1/24.abstract

[7] T. G. Hallam, A. Raghavan, H. Kolli, D. T. Dimitrov, P. Federico, H. Qi, G. F. McCracken, M. Betke, J. K. Westbrook, K. Kennard, and T. H. Kunz, "Dense and sparse aggregations in complex motion: Video coupled with simulation modeling," *Ecological Complexity*, vol. 7, no. 1, pp. 69 – 75, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1476945X09000609

[8] D. K. Dechmann, S. L. Heucke, L. Giuggioli, K. Safi, C. C. Voigt, and M. Wikelski, "Experimental evidence for group hunting via eavesdropping in echolocating bats," *Proceedings of the Royal Society B: Biological Sciences*, 2009.

[9] E. H. Gillam, N. I. Hristov, T. H. Kunz, and G. F. McCracken, "Echolocation behavior of brazilian free-tailed bats during dense emergence flights," *Journal of Mammalogy*, 2010. [Online]. Available: http://www.asmjournals.org/doi/abs/10.1644/09-MAMM-A-302.1

[10] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," *SIAM Journal on Control and Optimization*, vol. 42, no. 4, pp. 1143–1166, 2003.

[11] R. Moazzez-Estanjini, K. Li, and I. C. Paschalidis, "A least squares temporal difference actor-critic algorithm with applications to warehouse management," *Naval Research Logistics*, 2012, in print.

[12] P. Marbach and J. Tsitsiklis, "Simulation-based optimization of Markov reward processes," *IEEE Trans. Automat. Contr.*, vol. 46, no. 2, pp. 191–209, 2001.

[13] I. C. Paschalidis and D. Guo, "Robust and distributed stochastic localization in sensor networks: Theory and experimental results," *ACM Trans. Sensor Networks*, vol. 5, no. 4, pp. 34:1–34:22, 2009.

[14] S. Ray, W. Lai, and I. C. Paschalidis, "Statistical location detection with sensor networks," *Joint special issue IEEE/ACM Trans. Networking and IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2670–2683, 2006.