Multi-Robot Control for Surveillance of Dynamic Population Distributions

Andrew Q. Pham*, Christopher Lowe†, and Christopher Clark*
*Dept. of Engineering, Harvey Mudd College, Claremont, CA 91711
Email: apham@, clark@g.hmc.edu
†Dept. of Biological Sciences, California State University Long Beach, Long Beach, CA 90840
Email: clowe@csulb.edu

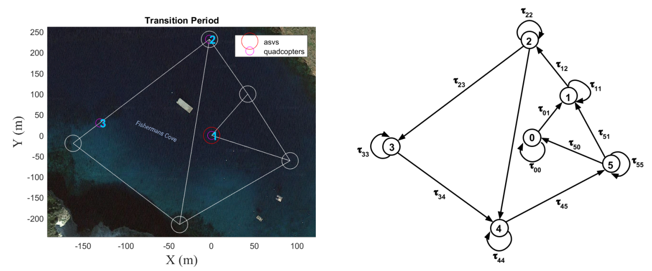
***Abstract*—This paper presents a multi-robot control system that enables the robots to monitor dynamic populations of individuals. In this work, the population's state is modeled as a geo-referenced graph, where individuals occupy nodes on the graph and use edges to travel between nodes. At any given time some fraction of the population may occupy each node. The control system presented uses calculated transition probabilities to dictate robot motion about the same graph. The system has the following properties: 1) robot navigation is collision-free, 2) robots only require local information when calculating their transition probabilities, and 3) the entire system dynamics are provably stable in that over time the distribution of robots at nodes will equal the distribution of individuals occupying the nodes. Simulation results indicate that for a 100 node graph occupied by 60 robots, the average difference between the steady state robot node distribution and desired node distribution was less than 0.01. Field test results with real quadcopters demonstrate that the controller can be implemented on a real system.**

***Index Terms*—Multi-robot system, Controller, Distribution Tracking, UAV Surveillance, Shark Tracking**

I. Introduction

Persistent environmental monitoring is a common need across a variety of applications. Examples include patrolling an area to detect intruders, surveying a location for air quality data, or monitoring animals in their natural habitat. Such tasks can oftentimes be dangerous, mundane, and/or costly for human beings, and are thus well suited for automation with robots [1]. Accordingly, a recent field of research has emerged around the use of multi-robot systems for environmental monitoring [2].

This paper presents an algorithmic solution to the control of a multi-robot system for monitoring a spatial population distribution that is abstracted as a graph, where each node represents a geo-referenced location where individuals of the population being monitored may be located. The individuals can move between nodes across edges of the graph, and the distribution of individuals across nodes is dynamic. Proposed is a decentralized collision-free stochastic controller that coordinates multiple robots to provably match the expected robot's node visit likelihood with the population's distribution of individuals across the nodes.



(a) Simulation of multi-robot system monitoring Fisherman's Cove, CA

(b) Equivalent diagrammed graph showing possible transitions

Fig. 1. Two depictions of graph abstraction for the multi-robot controller

To motivate this work, it is being applied to a fish monitoring application where nodes of a graph represent potential locations of fish. This model motivates the goal of developing a stochastic controller for multi-robot distributions. An example is illustrated in Fig. 1(a), where a graph is anchored to a particular area off the California coast, in which fish are known to aggregate. This graph can also be used as a navigation map for quadcopters equipped with downward facing cameras. The robots can fly between nodes and land on an Autonomous Surface Vehicle (ASV), which is stationed at the central node 0, for recharging. A similar stochastic framework is leveraged here, and the contributions of this work include:

- A decentralized, collision-free, stochastic control system that matches a node's robot visitation likelihood with the proportion of the tracked population residing at the node.
- Hardware experiments performed with real quadcopters demonstrating system feasibility.

This work differs from prior work in that the controller that can track dynamic distributions. It also allows higher priority nodes, (those with higher fractions of the distribution being tracked,) to be visited at a frequency proportional to their priority level. The distributions can be updated dynamically and the controller can respond in real-time. Combined with the benefits of being decentralized and producing collision-free navigation, the proposed controller provides a novel approach to a general multi-robot coordination problem.

II. Control System

A. Problem Definition

The problem can be described as coordinating n quadcopters in the set $Q = \{q_0, \ldots, q_{n-1}\}$ that are restricted to navigate along edges, and between nodes of a graph $G = (V, E)$. V is the set of nodes and E is the set of edges. The state of a quadcopter $x_t(q_i)$ at discrete time t is associated with some node v_j, such that $x_t(q_i) = v_j$. More generally, the state of all quadcopters is defined as $X_t(Q) = [x_t(q_0) \ldots x_t(q_{n-1})]$ where $X_t(Q) \in V$. The set of nodes that are occupied by quadcopters is denoted as V_{occ}

To further define the problem, the likelihood that a robot occupies any particular node $v_j \in V$ at time t is $p_{j,t}$, where $\sum_j p_{j,t} = 1$ for all t. Each node v_j also has a time varying fraction $p_{j,des,t}$ of the population of individuals being tracked by the quadcopters such that $\sum_j p_{j,des,t} = 1$ for all t. The goal of the controller is determine robot motion such that $p_{j,des,t} = p_{j,t}$ for all t.

Also navigating the nodes of G are m ASVs in the set $A = \{a_0, \ldots, a_{m-1}\}$. At periodic intervals, quadcopters must visit a node occupied by an ASV to land and recharge. That is, there must exist some time step t_r within the recharge period T_r where each quadcopter q_i is collocated with an ASV a_l at some vertex v_k. Specifically $x_{t_r}(q_i) = x_{t_r}(a_l) = v_k$. For work done in the remainder of this paper, each ASV will be stationary and reside at a single node.

If at every time step t, any quadcopter q_i chooses an action $u_{i,t} : x_{t-1}(q_i) \to x_t(q_i)$, and the action set for all quadcopters at time t is $U_t = \{u_{0,t}..u_{n,t}\}$, then the problem statement can be formulated as follows:

$$\begin{aligned} \underset{U_t}{\text{minimize}} \quad & \sum_{t=0}^{t_{max}} \sum_{j=0}^{|G|} p_{j,t,des} - p_{j,t}(X_t(Q), U_t) \\ \text{subject to} \quad & X_t(Q) \in V \;\; \forall t \\ & \exists t_r \in T_r \;\mid\; x_{t_r}(q_i) = x_{t_r}(a_l) \;\; \forall i \\ & (x_{t-1}(q_i) \to x_t(q_i)) \in E \;\; \forall t, \forall i \end{aligned}$$

The constraints state that 1) all quadcopters must reside on nodes during the discrete time steps, 2) there exists some time step within the recharge period at which each quadcopter is on an ASV, and 3) quadcopters must move only along edges.

B. Graph Construction

For this study, the nodes in V are generated by clustering geo-referenced positions from a historical data of fish population activity [3]. Methods such as k-means clustering were used to extract ideal locations for nodes. Other graph extraction algorithms can also be used.

Once the nodes are generated, directed edges are added to connect all nodes in V sequentially to create a uni-directional cyclic tour. Various TSP methods can be used to generate this tour. Since the tour is uni-directional, collision negotiations only occur between adjacent quadcopters, making the collision checking local and decentralized.

C. Motion Controller

To determine how quadcopters navigate along the tour, a stochastic motion controller that requires minimal run time and communication requirements is proposed. This controller specifies the transition probability $\tau_{j+1,j,t}$ at every time step t such that in real time operations, any quadcopter q_i located at v_j will randomly select to either stay at v_j or navigate to the next edge connected node v_{j+1} with likelihood $\tau_{j+1,j,t}$. Within this context, the control action $u_{i,t}$ uses a uniform sampling function $rand() \in [0, 1]$ to determine the next node to visit:

$$u_{i,t} = \begin{cases} x_{t-1}(q_i) \to x_t(q_i) = v_{j+1} & if \ rand() < \tau_{j+1,j,t} \\ x_{t-1}(q_i) \to x_t(q_i) = v_j & else \end{cases} \tag{1}$$

These transition probabilities can be encapsulated in a transition matrix $\mathbf{R}_t$, where the jth row and kth column element in $\mathbf{R}_t$ is denoted as $\tau_{j,k,t}$. Then, given the current distribution of quadcopters $\mathbf{p}_t = [p_{0,t} \ldots p_{|V|-1,t}]^T$, the quadcopter distribution update can then be described by:

$$\mathbf{p}_{t+1} = \mathbf{R}_t \mathbf{p}_t \tag{2}$$

Thus, the goal is to set the elements $\tau_{j,k,t}$ of $\mathbf{R}_t$ such that $\mathbf{p}_t$ converges to the desired distribution $\mathbf{p}_{des,t} = [p_{0,des,t} \cdots p_{|V|-1,des,t}]^T$ over time. The following values for the elements of $\mathbf{R}_t$ are proposed:

$$\begin{aligned} \tau_{j,j,t} &= \begin{cases} 1 & if \ v_{j+1} \in V_{occ} \\ \tau_{stay-} & else \ if \ v_{j-1} \in V_{occ} \\ \tau_{stay+} & else \end{cases} \\ \tau_{j+1,j,t} &= \begin{cases} 0 & if \ v_{j+1} \in V_{occ} \\ \tau_{move-} & else \ if \ v_{j-1} \in V_{occ} \\ \tau_{move+} & else \end{cases} \end{aligned} \tag{3}$$

In Eq. (3), $\tau_{j,j,t}$ determines the likelihood of staying at node v_j, while $\tau_{j+1,j,t}$ determines the likelihood of moving from v_j to the next node of a tour v_{j+1}. Since a quadcopter must either stay or move on, then $\tau_{j,j,t}$ + $\tau_{j+1,j,t}$=1.

To eliminate the possibility of collisions, both the likelihood of staying at current node v_j is set to 1, and the likelihood of transitioning to the next node v_{j+1} is set to 0, if v_{j+1} is occupied, (i.e. $v_{j+1} \in V_{occ}$).

Equations (4)-(7) provide the transition probabilities for cases where the next node on a tour is not occupied and can be transitioned to. These transition probabilities are functions of individual node errors $e_{j,t} = p_{des,j,t} - p_{j,t}$ and constant proportional control gains K_a and K_b.

$$\tau_{stay-} = \frac{\left[1 - p_{j+1,t} + \frac{K_a e_{j,t}}{p_{j,t}} + \frac{K_b (e_{j-1,t} - e_{j+1,t})}{p_{j-1,t}(p_{j,t})}\right]}{1 - p_{j+1,t}} \tag{4}$$

$$\tau_{stay+} = \frac{\left[1 - p_{(j+1),t} + \frac{K_a e_{j,t}}{p_{j,t}}\right]}{1 - p_{(j+1),t}} \tag{5}$$

$$\tau_{move-} = \frac{-\left(\frac{K_a e_{j,t}}{p_{j,t}} + \frac{K_b(e_{j-1,t} - e_{j+1,t})}{p_{j-1,t}(p_{j,t})}\right)}{1 - p_{j+1,t}} \tag{6}$$

$$\tau_{move+} = \frac{-K_a e_{j,t}}{p_{j,t}(1 - p_{j+1,t})} \tag{7}$$

As shown in in Eq. (3), if the next node is *not* occupied, the controller next considers whether or not the previous node v_{j-1} is occupied, (i.e. $v_{j-1} \in V_{occ}$). If it is, the controller uses proportional control on node density errors for nodes v_{j-1}, v_j, v_{j+1} to determine the likelihood of staying versus moving on, (Eq. (4) and (6)). This ensures that quadcopters at the previous node v_{j-1} don't get stuck at that node when there is less there to observe, (e.g. $p_{j-1,des}$ is low).

If the previous node is not occupied, a proportional controller using error associated with the current node v_j is used to set the transition probability, (Eq. (5) and (7)).

Ideally, the individual node errors will all decay to zero over time, such that for all nodes the likelihood of robots visiting a node matches the fraction of individuals being monitored visiting that node. It can be proved, (outside the scope of this abstract), that setting the control gains K_a and K_c appropriately will lead to stable error dynamics that decay errors to zero over time.

A few notes on the transition probability calculations. First, quadcopters only need to know information about previous, current, and next nodes: $e_{j-1,t}$, $e_{j,t}$, $e_{j+1,t}$, allowing the system to be decentralized. Second, they ensure collision-free navigation by setting likelihoods of transitioning to occupied nodes to 0. Third, the desired distribution $\mathbf{p}_{des,t}$ can be estimated and initialized with prior knowledge (i.e. known population migratory behavior) or possibly estimated in real-time as it changes, thereby enabling the tracking of dynamic distributions. Fourth, the calculations in equations (3)-(7) require relatively insignificant computational power and can be accomplished in real-time. Finally, the following theorem can be proven, although outside the scope of this abstract:

Theorem II.1. *The error dynamics for $\mathbf{e}_t$ will be stable for gain values of $0 < K_a < 1$ and $0 < K_b < \frac{1}{2}$.*

III. Simulation Tests

All simulation tests were written in MATLAB. To test the controller, simulations were run on graphs with random desired tracking distributions. The number of nodes in the graph were varied between values of 5, 10, 20, 40, 60, 80, and 100. The number of quadcopters was set to values of 1, 10, 20, 30, 40, 50, and 60 percent of the nodes. For each node/quadcopter number combination, 100 trials were conducted and performance was evaluated as the average error from Eq. (3) across all nodes, across all 100 trials.

As shown in Fig. 2 the error values predominantly falls within < 0.05. The error plot indicates that the controller converges to the desired distribution with some steady state error. Error increases as the percentage of nodes occupied increase and simultaneously the number of nodes decrease.

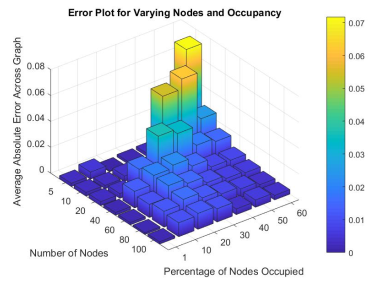
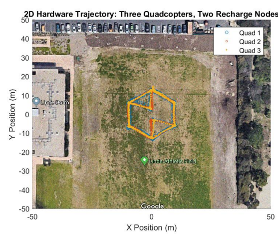
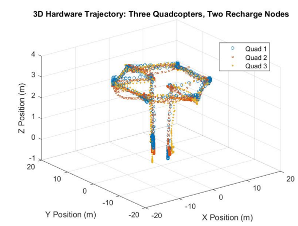


Fig. 2. Average absolute error as a function of number of nodes and robots.



(a) 2D trajectory of two recharge node trial



(b) 3D trajectory of two recharge node trial

Fig. 3. Trajectories of hardware trials conducted at Linde Field at Harvey Mudd College.

IV. Field Tests

Physical experiments were conducted with three modified 3DR Solo quadcopters running OpenSolo. Flight times range from 10 to 20 minutes. The flight controller of the 3DR Solos were upgraded from a Pixhawk 2.1 Black Cube to a Pixhawk 2.1 Green Cube flashed with Arducopter v3.5. Autonomous functionality was programmed with Dronekit. The script was run on a remote computer that connected to each quadcopter through a designated WiFi link. For these preliminary tests, the system controller was centralized for ease of implementation.

Fig. 3 depicts the trajectories logged during some of the hardware trials. As shown in the figures, theses trials validated the ability for the controller to coordinate the collision-free motion of multiple quadcopters around the graph.

References

[1] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics Automation Magazine*, vol. 19, no. 1, pp. 24–39, March 2012.

[2] D. Portugal and R. Rocha, "A survey on multi-robot patrolling algorithms," in *Technological Innovation for Sustainability*, L. M. Camarinha-Matos, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 139–146.

[3] K. D. Smith, S. Hsiung, C. White, C. G. Lowe, and C. M. Clark, "Stochastic modeling and control for tracking the periodic movement of marine animals via auvs," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 3101–3107.