Design of Control Policies for Spatially Inhomogeneous Robot Swarms with Application to Commercial Pollination

(Article begins on next page)

Design of Control Policies for Spatially Inhomogeneous Robot Swarms with Application to Commercial Pollination

Spring Berman, Vijay Kumar, and Radhika Nagpal

Abstract—We present an approach to designing scalable, decentralized control policies that produce a desired collective behavior in a spatially inhomogeneous robotic swarm that emulates a system of chemically reacting molecules. Our approach is based on abstracting the swarm to an advection-diffusion-reaction partial differential equation model, which we solve numerically using smoothed particle hydrodynamics (SPH), a meshfree technique that is suitable for advection-dominated systems. The parameters of the macroscopic model are mapped onto the deterministic and random components of individual robot motion and the probabilities that determine stochastic robot task transitions. For very large swarms that are prohibitively expensive to simulate, the macroscopic model, which is independent of the population size, is a useful tool for synthesizing robot control policies with guarantees on performance in a top-down fashion. We illustrate our methodology by formulating a model of rabbiteye blueberry pollination by a swarm of robotic bees and using the macroscopic model to select control policies for efficient pollination.

I. INTRODUCTION

Tasks that require parallelism, redundancy, and adaptation to dynamic environments can potentially be performed very robustly and efficiently by a swarm robotic system, which would consist of hundreds or thousands of autonomous robots with limited sensing, communication, and computation capabilities. Key challenges in the development of such systems include the accurate prediction of swarm behavior and the design of robot controllers that can be proven to produce a target macroscopic outcome. The controllers should be scalable, meaning that they maintain the successful operation of the system regardless of the number of robots.

To ensure scalability, we propose a control approach that is decentralized, in which robots make decisions using only local information obtained via sensing and/or communication without knowledge of the global system state. The robots that we consider are unidentified and each programmed with the same set of control algorithms. We assume a broadcast architecture [26] in which a supervisory agent computes the parameters that govern the robots' motion and task transitions and transmits them to the swarm, without requiring knowledge of the population size or individual

S. Berman and R. Nagpal are with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 (e-mail: {sberman,rad}@eecs.harvard.edu)

V. Kumar is with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104 (e-mail: kumar@seas.upenn.edu)

robot actions. This paradigm facilitates the communication of information to all the robots while avoiding the lossiness and inefficiencies of a multi-hop ad-hoc network.

In previous work [5], [24], we described a methodology for producing target swarm behaviors that can be implemented using this control paradigm. Our approach consists of developing an abstraction of a swarm and using this model to analyze system performance and synthesize robot control policies that cause the populations of different swarm elements, defined as robots at various tasks and objects of different types with which they interact, to evolve in a desired way. The methodology was developed for swarms in which robots execute task transitions *stochastically*, which allows us to produce any of a range of population distributions by designing an appropriate set of transition probability rates. For systems in which there are task transitions that are triggered by encounters between elements, we specified that swarm elements are always uniformly randomly distributed throughout the environment, which can be realized when the robots execute random walks. This allowed us to model the swarm as a *well-mixed* chemical reaction system, which could be abstracted to a set of ordinary differential equations.

In this paper, we extend our methodology to *spatially inhomogeneous* swarms, in which elements are arbitrarily distributed throughout space. We consider systems in which robots follow a deterministic velocity field in addition to random motion. We generate the continuous dynamics of individual elements as they move and interact in a physical environment using a *micro-continuous model*. In this model, we define the robots' motion using the technique of random walk particle tracking [10], [33], a method from statistical physics, and implement the robot task-switching using the formulation given in [4]. In the limit of infinite swarm populations, the system abstracts to a set of *advection-diffusion-reaction* (ADR) partial differential equations (PDE's) [8], the *macro-continuous model*, which governs the time evolution of concentrations of different elements over a spatial domain. Similar to this approach is the work [15] on developing rigorous abstractions of a swarm to a PDE model based on the Fokker-Planck equation, which describes the time evolution of the probability density function of the position of a robot. By designing the parameters of the macro-continuous model and using them to define the robot motion controllers and stochastic policies for task transitions, a supervisory agent can produce a target collective behavior. This constitutes a novel application of a "top-down" controller synthesis approach to spatially inhomogeneous swarms.

The general ADR equations that define the macro-

continuous model cannot be solved analytically, and instead must be solved using numerical methods [17]. A widely applied technique is a grid or mesh-based method such as finite difference, finite volume, and finite element methods [6]. Grid-based methods are unsuitable for systems that consist of discrete physical entities (such as robots) rather than a continuum [21], and they can suffer from numerical dispersion and artificial oscillations when simulating advection-dominated systems [16], [33], which we may want to design. A more appropriate technique to use for our type of system is a meshfree method [20], which solves integral equations and PDE's with all kinds of boundary conditions using a set of arbitrarily distributed nodes or particles without imposing connectivity among these elements with a mesh. We numerically solve the ADR equations using smoothed particle hydrodynamics (SPH), a meshfree particle method that was originally developed to simulate astrophysical phenomena and has been extended to a variety of problems in computational fluid dynamics and solid mechanics [21], [22].

We apply our methodology to synthesize control policies for a swarm of insect-inspired micro air vehicles [37] to pollinate an orchard of rabbiteye blueberries (*Vaccinium ashei*), a scenario of interest to the Robobees project [1], whose objective is to develop a colony of robotic bees. Insect pollination is necessary for adequate commercial yields of this crop; almost all rabbiteye cultivars are self-sterile and require cross-pollination with a compatible cultivar, of which at least two should be interplanted [9], [25]. Our investigation of effective swarm strategies and their required robot capabilities will inform the design of the robotic bees. We describe micro-continuous and macro-continuous models of the scenario, demonstrate a close correspondence between the models, and illustrate how the macro-continuous model can be used to select motion controllers and stochastic policies for the robots that produce efficient pollination.

II. MICRO-CONTINUOUS MODEL

A. Robot Motion and Concentration Distribution

We consider systems in which the position $\mathbf{q}_i \in Y \subset \mathbf{R}^d$, $d \in \{1, 2, 3\}$, of each robot i (and any other swarm element) evolves according to a deterministic velocity field $\mathbf{v}(\mathbf{q}_i)$ and a Brownian motion that drives diffusion, where D is the associated diffusion coefficient. The velocity field can be designed, and it may have a component that is fixed by the bulk motion of the medium, such as wind or water, in which the robots operate. Here diffusion models randomness in robot movement that either arises from inherent noise due to sensor and actuator errors or is actively added by the robot's motion controllers, or both. D is the sum of D_{inh}, a constant determined by the first source of randomness, and D_{act}, a tunable parameter that produces the second source. A robot i, which can be represented as a point kinematic agent, updates its displacement at each (very small) timestep Δt according to the Itō-Taylor integration scheme [12]

$$\mathbf{q}_i(t + \Delta t) = \mathbf{q}_i(t) + \mathbf{v}(\mathbf{q}_i(t))\Delta t + \sqrt{2D\Delta t}\,\mathbf{I}\,\mathbf{Z}(t)\,, \quad (1)$$

where $\mathbf{I} \in \mathbb{R}^{d \times d}$ is the identity matrix and $\mathbf{Z} \in \mathbb{R}^d$ is a vector of independent, normally distributed random variables with zero mean and unit variance.

A swarm of N robots that move according to equation (1) can be viewed as a random walk particle tracking (RWPT) scheme, a Lagrangian approach to solving advection-diffusion problems. In the limit $N \to \infty$, the Fokker-Planck equation becomes equivalent to the advection-diffusion PDE (defined in Section III), which governs the time evolution of the concentration of robots over the domain, $x(\mathbf{q}, t)$ [10], [33]. Using the particle tracking analogy, we consider the robots to be a discrete distribution of the mass of the swarm and associate each robot i with a mass $m_i = m$. The robot masses can be mapped to concentration values as described in [2], [36]. The concentration $x(\mathbf{q}, t)$ at a point $\mathbf{q} \in Y$ can be approximated by the smoothed integral interpolation

$$x(\mathbf{q}, t) = \int_Y \tilde{x}(\mathbf{q}', t)\, \zeta(\mathbf{q} - \mathbf{q}')\, d\mathbf{q}', \quad (2)$$

$$\tilde{x}(\mathbf{q}, t) = \sum_{i=1}^{N} m\delta(\mathbf{q} - \mathbf{q}_i(t)), \quad (3)$$

where $\delta(\mathbf{q})$ is the Dirac delta function and $\zeta(\mathbf{q})$ is a projection function with finite support that satisfies $\int_Y \zeta(\mathbf{q})d\mathbf{q} = 1$ and is ideally invariant with respect to coordinate transformations. A particle approximation for $x(\mathbf{q}, t)$ is given by

$$x(\mathbf{q}, t) = \sum_{i=1}^{N} m\zeta(\mathbf{q} - \mathbf{q}_i(t)). \quad (4)$$

Most particle tracking implementations use $\zeta(\mathbf{q}) = 1/V$ for points within a small cube of volume V around $\mathbf{q}$ and $\zeta(\mathbf{q}) = 0$ otherwise [16]. This corresponds to dividing the domain Y into a grid of G cubic cells with volume V_c, finding the mass of all particles inside each cell c, M_c, and defining the concentration at all points $\mathbf{q}$ inside cell c as $x(\mathbf{q}, t) = M_c/V_c$.

B. Robot Task Switching

The robots' transitions between tasks can be modeled as chemical reactions: the robots switch stochastically, either spontaneously or upon encountering certain objects or other robots, at rates that are determined by constants that are analogous to *reaction rate constants*. Borrowing chemical reaction terminology, a *species* X_i symbolizes a robot that is performing task i or an object of type i with which the robots interact, and a *complex* is a combination of species that appears before or after a reaction arrow. The rate constant k_{ij} associated with a reaction that converts complex i into complex j is a function of a parameter c_{ij}, where $c_{ij}dt$ is the average probability that a particular combination of reactant elements will transition to product elements in the next infinitesimal time interval dt.

A robot's decision to switch tasks spontaneously is modeled as a unimolecular reaction, for which $k_{ij} = c_{ij}$ [13]. The robot control policy that implements this transition has no dependence on spatial properties of the system: a robot

that is performing task i computes a uniformly distributed random number in $[0,1]$, $u \sim U(0,1)$, at each timestep Δt and executes the transition if $u < c_{ij}\Delta t$.

A robot's decision to switch tasks upon encountering another element is modeled as a bimolecular reaction. Let $\mathbf{r} = \mathbf{q}_o(t) - \mathbf{q}_p(t)$ be the initial relative displacement of two reactants o and p and let $\Delta \mathbf{q} = \mathbf{q}_p(t+\Delta t) - \mathbf{q}_o(t+\Delta t) + \mathbf{r}$. The probability of the reactants occupying the same position at time $t+\Delta t$ is $P(\Delta \mathbf{q} = \mathbf{r})$. Defining $p(\mathbf{r})$ as the probability density of $\Delta \mathbf{q}$ and specifying that all reactants are associated with the same arbitrary mass m, c_{ij} is given by [4]

$$c_{ij} = k_{ij} m p(\mathbf{r}). \tag{5}$$

We consider reactions in which one reactant is a robot that moves according to equation (1) and the other is stationary. The function $p(\mathbf{r})$ is then the probability density of the moving reactant's displacement $\Delta \mathbf{q}_i(t) = \mathbf{v}(\mathbf{q}_i(t))\Delta t + \sqrt{2D\Delta t}\mathbf{IZ}(t)$, which is a bivariate normal distribution with mean $\mathbf{v}(\mathbf{q}_i(t))\Delta t$ and covariance matrix $2D\Delta t\mathbf{I}$. The standard deviation of the robot's displacement is $\sigma = (2D\Delta t)^{1/2}$.

As in the case of a well-mixed system [14], the robot control policy that implements the transition can be defined by decomposing the probability $c_{ij}\Delta t$ into the product of c_{ij}^e, the probability that the robot will encounter a particular reactant within its sensing range in the next Δt, and $c_{ij}^d \Delta t$, the probability that the robot will decide to follow through with the transition given an encounter. We assume that a robot's sensing range is a sphere in $\mathbb{R}^d$ whose radius r is on the order of the robot's typical random displacement per timestep, reflecting the robot's very limited sensing capacity. The probability that a robot will encounter a reactant at a position $\mathbf{r}$ relative to the robot in the next timestep is $c_{ij}^e = \int_\Omega p(\mathbf{r}')d\mathbf{r}'$, where $\Omega \subset \mathbb{R}^d$ is a sphere of radius r centered at the relative position $\mathbf{r}$. For small r, $\int_\Omega p(\mathbf{r}')d\mathbf{r}' \approx p(\mathbf{r})V_\Omega$, where V_Ω is the volume of Ω. Using equation (5), we can decompose c_{ij} as

$$c_{ij} = c_{ij}^d \cdot c_{ij}^e, \quad c_{ij}^d = \frac{k_{ij}m}{V_\Omega}, \quad c_{ij}^e = p(\mathbf{r})V_\Omega. \tag{6}$$

At each timestep, a robot moves according to equation (1) and, if it senses a reactant, computes $u \sim U(0,1)$ and executes the transition if $u < c_{ij}^d \Delta t$. The robot then becomes a species in the product of the reaction and may follow a different displacement equation of the form (1).

In simulation, we may implement an encounter-based transition in an alternate way. At each timestep Δt, robot i is moved a deterministic displacement $\mathbf{v}(\mathbf{q}_i)\Delta t$ from equation (1). We define a sphere of radius r_{sim}, centered at the robot's new position, which contains the vast majority of reactants that the robot is likely to reach in Δt due to its diffusive motion. For instance, we may set $r_{sim} = 2\sigma$. The value of c_{ij} is computed from equation (5) for each potential reactant in this region, along with $u \sim U(0,1)$, and the reaction is executed with the reactant for which $u < c_{ij}\Delta t$, if any. The robot is moved to the position of the chosen reactant. If no

reaction occurs, the robot's diffusive motion is simulated as $\sqrt{2D\Delta t}\mathbf{IZ}(t)$ from equation (1).

III. MACRO-CONTINUOUS MODEL

The macro-continuous model of the time evolution of the species concentrations, $x_s(\mathbf{q},t)$, $s = 1,...,S$, is given by a set of *advection-diffusion-reaction* (ADR) equations, which describe the conservation of chemical species in a fluid. Define $\mathbf{v}_s(\mathbf{q})$ as a velocity field that specifies the deterministic motion of species s, D_s as the diffusion coefficient of species s, K_s as the set of rate constants k_{ij} that are associated with the reactions involving species s, and $R_s(K_s, \mathbf{x})$ as the sum of the corresponding reaction rates evaluated at local concentrations $\mathbf{x}(\mathbf{q})$. The ADR equations are:

$$\frac{\partial x_s}{\partial t} + \nabla \cdot (\mathbf{v}_s(\mathbf{q})x_s) = D_s\nabla^2 x_s + R_s(K_s, \mathbf{x}), \quad s = 1,...,S \tag{7}$$

The ADR equation for x_s is known as the *advection-diffusion equation* when $R_s(K_s, \mathbf{x}) = 0$ and the *reaction-diffusion* equation when $\mathbf{v}_s = \mathbf{0}$. To synthesize a target macroscopic behavior in a swarm modeled by equations (7), a supervisory agent computes the k_{ij} and the tunable components of $\mathbf{v}_s(\mathbf{q})$ and D_s, $s = 1,...,S$.

The technique of smoothed particle hydrodynamics (SPH) can be used to numerically solve the ADR equations (7). In this method, a function $f(\mathbf{q},t)$ over a domain is represented in terms of its values at a set of arbitrarily distributed particles. This converts the governing PDE equations of a system into a set of ODE's, each of which describes the time evolution of a variable associated with a particle. The value of a variable at a particle is influenced by values at particles within a local neighborhood only.

The SPH formulation is derived by first approximating $f(\mathbf{q})$ with a smoothed integral interpolation, similarly to equation (2):

$$f(\mathbf{q}) = \int_Y f(\mathbf{q}') w(\mathbf{q} - \mathbf{q}', h) d\mathbf{q}'. \tag{8}$$

Here $w(\mathbf{q} - \mathbf{q}', h)$ is a differentiable kernel function with smoothing length h and compact support. The kernel w satisfies the properties

$$\lim_{h \to 0} w(\mathbf{q} - \mathbf{q}', h) = \delta(\mathbf{q} - \mathbf{q}'), \tag{9}$$

$$\int_Y w(\mathbf{q} - \mathbf{q}', h) d\mathbf{q}' = 1, \tag{10}$$

$$w(\mathbf{q} - \mathbf{q}', h) = 0 \quad \text{for} \quad ||\mathbf{q} - \mathbf{q}'|| > \kappa h, \tag{11}$$

where κ is a constant that defines the support domain of the kernel of point $\mathbf{q}$. The kernel function is frequently defined as a Gaussian or a spline that approximates a Gaussian [21].

The integral (8) can be approximated using a Monte Carlo integration scheme as follows. The spatial distribution of the mass in the system is represented by a set of Q particles. Particle i has position $\mathbf{r}_i$, mass m_i, and density ρ_i, where m_i/ρ_i is the volume associated with the particle. The number

of particles per unit volume, $n_i = \rho_i/m_i$, is calculated as [35]

$$n_i = \sum_{j=1}^{Q} w(\mathbf{r}_{ij}, h), \quad \mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i. \quad (12)$$

Replacing $d\mathbf{q}'$ in equation (8) by the volume $m_i/\rho_i = 1/n_i$ at the particle locations, the particle approximation for $f(\mathbf{q})$ is given by

$$f(\mathbf{q}) = \sum_{i=1}^{Q} \frac{1}{n_i} f(\mathbf{r}_i) w(\mathbf{q} - \mathbf{r}_i, h) . \quad (13)$$

The error in approximating equation (8) by equation (13) is $O(h^2)$ [27]. Computing the gradient of $f(\mathbf{q})$ using the particle approximation entails differentiating the kernel w rather than the function itself.

In recent years, the SPH technique has been used to define decentralized motion control laws for robot swarms at the *micro-continuous* level to achieve pattern generation in obstacle-filled environments [30], [31] as well as deployment, sensor coverage, patrolling, flocking, and formation control behaviors [28], [29]. These works model a swarm as a fluid, which may be subjected to an external force, and represent each robot as an SPH particle that has an associated position, velocity, mass, density, energy, and pressure. The robots use the SPH formulations of the governing equations for the conservation of mass, momentum, and energy, along with an equation of state for pressure, to update their associated quantities and compute their control law. The approach is scalable because these updates only require information from robots within a local neighborhood that corresponds to the support domain of w.

In contrast, our application of the SPH method at the *macro-continuous* level does not associate particles with robots, but rather employs the particles as computational elements that track concentration values at points in space. We use an SPH scheme similar to those that have been recently applied to model solute transport in porous media [16], [35]. We define a set of particles for each of the $S' \leq S$ distinct velocity fields $\mathbf{v}_s$ in equations (7). The set associated with $\mathbf{v}_{s'}$, $s' \in \{1, ..., S'\}$, tracks the concentrations of all species s for which $\mathbf{v}_s = \mathbf{v}_{s'}$. The velocity of each particle $i \in \{1, ..., Q_{s'}\}$ in this set is

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_{s'}(\mathbf{r}_i). \quad (14)$$

The diffusion term in equations (7), $D_s\nabla^2 x_s$, could be evaluated by differentiating the interpolated concentration twice. However, the resulting expression contains the second derivative of w, which can be noisy and sensitive to particle disorder, so instead we use an SPH discretization of the Laplace operator that involves only first order derivatives of w [18]:

$$\nabla^2 x_s|_{\mathbf{r}_i} = -2 \sum_{j=1}^{Q_{s'}} \frac{1}{n_j} (x_s(\mathbf{r}_j) - x_s(\mathbf{r}_i)) F(\mathbf{r}_{ij}), \quad (15)$$

$$F(\mathbf{r}_{ij}) = \frac{\mathbf{r}_{ij} \cdot \nabla_{\mathbf{r}_{ij}} w(\mathbf{r}_{ij}, h)}{||\mathbf{r}_{ij}||^2}, \quad (16)$$

where $\nabla_{\mathbf{r}_{ij}} w(\mathbf{r}_{ij}, h)$ is the directional derivative of w along $\mathbf{r}_{ij}$. For a set of irregularly spaced particles, $n_i \neq n_j$ in general, which produces an asymmetry in the magnitude of the contribution of concentration from particle i to particle j and vice versa. To rectify this, n_j is replaced by either the arithmetic or harmonic average of n_i and n_j [16]. Choosing the harmonic average and multiplying equation (15) by D_s, the particle approximation of the diffusion term becomes

$$D_s\nabla^2 x_s|_{\mathbf{r}_i} = D_s \sum_{j=1}^{Q_{s'}} \frac{n_i + n_j}{n_i n_j} (x_s(\mathbf{r}_i) - x_s(\mathbf{r}_j)) F(\mathbf{r}_{ij}). \quad (17)$$

In the reaction rate terms of equations (7), the concentration at $\mathbf{r}_i$ of a species that is not tracked by particle i is evaluated using the particle approximation (13) with the set of particles that do track the species.

The SPH formulation of the ADR equation for each species s that is tracked by particle $i \in \{1, ..., Q_{s'}\}$ is

$$\frac{dx_s}{dt}\bigg|_{\mathbf{r}_i} = D_s\nabla^2 x_s|_{\mathbf{r}_i} + R_s(K_s, x_1(\mathbf{r}_i), ..., x_S(\mathbf{r}_i)). \quad (18)$$

The SPH method is implemented by initializing the particle positions and concentrations and then numerically integrating equations (14) and (18) using standard techniques such as Runge-Kutta methods and the Velocity Verlet scheme [21].

IV. APPLICATION: COMMERCIAL POLLINATION
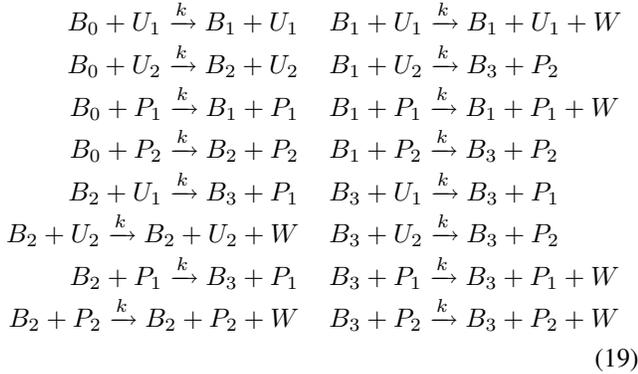
A. Micro-Continuous Model

We simulate a 50 ft $\times$ 18 ft section of a rabbiteye blueberry orchard that consists of alternating rows of two cultivars, as is commonly recommended in the literature [23]. The section contains four rows of three plants each, which are spaced 6 ft apart within a row and 12 ft apart between rows, the industry standard [32]. Each plant has 10^4 flowers, the quantity for a mature rabbiteye plant [34], which are uniformly randomly distributed throughout the plant canopy. Initially, a colony of robotic bees occupies a 2 ft $\times$ 3 ft hive to the left of the rows; the robots are uniformly randomly distributed throughout the hive. Fig. 1 illustrates the orchard layout and the initial placement of the robots.

The supervisory agent in our architecture can be a computer at the hive [3] with the computational resources to calculate the robot motion and task transition parameters for a specific macroscopic objective, which may involve solving the ADR equations to predict the system performance. The robots would have sufficient power to undertake brief flights away from the hive [19]. They would return to the hive to recharge, upload data, and receive parameters from the hive computer. Here we simulate a portion of one flight.

We assume that each robotic bee is equipped with a compass and thus can fly with a constant heading. The robots search for flowers by flying with a deterministic component to the right superimposed with a random walk according to equation (1), with $\mathbf{v}(\mathbf{q}_i) = \mathbf{v} = [v\ 0]^T$. We specify the robot flight speed as $v + 2\sigma/\Delta t = 16.4$ ft/s, which is believed to be a realistic speed for robotic bees as it falls within the upper range of speeds attainable by wild orchid bees [7].

The robotic bees in this scenario are assumed to be capable of recognizing a flower that is very close by, potentially using an ultraviolet light sensor; flying to the flower; and hovering briefly while vibrating the flower to release its pollen as is done by efficient blueberry pollinators [9]. The flower visits are modeled as being instantaneous. We assume that a rabbiteye flower, each of which produces 8000-17000 pollen grains [11], transfers pollen to each robot that visits it and that sufficient pollen remains on the robot to pollinate an arbitrary number of flowers on subsequent visits.

We define a set of reactions that represents the actions of pollen retrieval, pollination, and unproductive flower visits. The robot species are defined as follows: B_0 is a robot without pollen, B_i is a robot carrying pollen from cultivar $i \in \{1, 2\}$, and B_3 is a robot carrying pollen from both cultivars. U_i and P_i symbolize an unpollinated flower and a pollinated flower, respectively, of cultivar $i \in \{1, 2\}$. W is a "waste product" indicating that a flower visit has not resulted in pollination or retrieval of a pollen type that is not already on the robot. Each reaction is associated with the same rate constant k, since it is unrealistic for the robots to be able to distinguish between flowers of different cultivars or flowers that have been visited by other robots. The reactions are:

$$
\begin{array}{ll}
B_0 + U_1 \xrightarrow{k} B_1 + U_1 & B_1 + U_1 \xrightarrow{k} B_1 + U_1 + W \\
B_0 + U_2 \xrightarrow{k} B_2 + U_2 & B_1 + U_2 \xrightarrow{k} B_3 + P_2 \\
B_0 + P_1 \xrightarrow{k} B_1 + P_1 & B_1 + P_1 \xrightarrow{k} B_1 + P_1 + W \\
B_0 + P_2 \xrightarrow{k} B_2 + P_2 & B_1 + P_2 \xrightarrow{k} B_3 + P_2 \\
B_2 + U_1 \xrightarrow{k} B_3 + P_1 & B_3 + U_1 \xrightarrow{k} B_3 + P_1 \\
B_2 + U_2 \xrightarrow{k} B_2 + U_2 + W & B_3 + U_2 \xrightarrow{k} B_3 + P_2 \\
B_2 + P_1 \xrightarrow{k} B_3 + P_1 & B_3 + P_1 \xrightarrow{k} B_3 + P_1 + W \\
B_2 + P_2 \xrightarrow{k} B_2 + P_2 + W & B_3 + P_2 \xrightarrow{k} B_3 + P_2 + W
\end{array}
$$
$$(19)$$

The robot behavior of encountering a flower in very close proximity with its limited sensing range and deciding whether to visit it is simulated according to the procedure described in the last paragraph of Section II-B.

We compute the concentration fields of the species as described in Section II-A. The species initially present in the simulation are B_0, U_1, and U_2. The initial nonzero concentrations of B_0 were set to 1 at the center $\mathbf{q}^c$ of each grid cell c within the hive region. The mass associated with each of the N_{B_0} robots was computed as $m = \frac{1}{N_{B_0}} \sum_{c=1}^{G} x_{B_0}(\mathbf{q}^c, 0) V_c$. The U_1, U_2 flowers were assigned the same mass m as the robots by setting their initial nonzero concentrations to $x_{U_k}^0 = (m N_{U_k})/(V_c G_{U_k})$ at the center of each of the G_{U_k} grid cells in the rows of cultivar $k \in \{1, 2\}$. To initialize the robot and flower positions, we computed their populations per cell as $N_s^c = x_s(\mathbf{q}^c, 0) V_c/m$, $s \in \{B_0, U_1, U_2\}$, rounded this number to the nearest integer, and distributed the N_s^c robots uniformly randomly inside the cell.
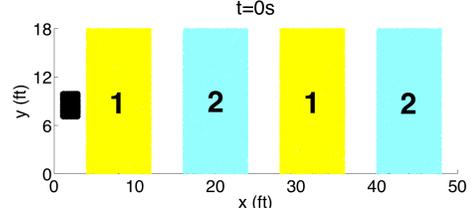

Fig. 1. An initial distribution of robotic bees (black) and unpollinated flowers of cultivars 1 (yellow) and 2 (cyan) in the micro-continuous model.

B. Macro-Continuous Model

The time evolution of the concentrations of the species in the set of reactions (19) is governed by the following ADR equations,

$$
\begin{aligned}
\frac{\partial x_{B_i}}{\partial t} + \nabla \cdot (\mathbf{v} x_{B_i}) &= D\nabla^2 x_{B_i} + R_{B_i}(k, \mathbf{x}), \\
& \quad i = 0, 1, 2, 3, \\
\frac{\partial x_{U_1}}{\partial t} = -\frac{\partial x_{P_1}}{\partial t} &= -k(x_{B_2} x_{U_1} + x_{B_3} x_{U_1}), \\
\frac{\partial x_{U_2}}{\partial t} = -\frac{\partial x_{P_2}}{\partial t} &= -k(x_{B_1} x_{U_2} + x_{B_3} x_{U_2}), \\
\frac{\partial x_W}{\partial t} &= k(x_{B_1} x_{U_1} + x_{B_1} x_{P_1} + x_{B_2} x_{U_2} + \\
& \quad x_{B_2} x_{P_2} + x_{B_3} x_{P_1} + x_{B_3} x_{P_2}),
\end{aligned}
$$
$$(20)$$

where

$$
\begin{aligned}
R_{B_0}(k, \mathbf{x}) &= -k(x_{B_0} x_{U_1} + x_{B_0} x_{U_2} + x_{B_0} x_{P_1} + x_{B_0} x_{P_2}), \\
R_{B_1}(k, \mathbf{x}) &= k(x_{B_0} x_{U_1} + x_{B_0} x_{P_1} - x_{B_1} x_{U_2} - x_{B_1} x_{P_2}), \\
R_{B_2}(k, \mathbf{x}) &= k(x_{B_0} x_{U_2} + x_{B_0} x_{P_2} - x_{B_2} x_{U_1} - x_{B_2} x_{P_1}), \\
R_{B_3}(k, \mathbf{x}) &= k(x_{B_1} x_{U_2} + x_{B_1} x_{P_2} + x_{B_2} x_{U_1} + x_{B_2} x_{P_1}).
\end{aligned}
$$
$$(21)$$

Our SPH formulation of the system uses a set B of Q_B particles with positions $\mathbf{r}_i$ that move with velocity $\mathbf{v}$ and track the concentrations of robot species, as well as a set F of Q_F stationary particles with positions $\mathbf{s}_j$ that track the concentrations of flower species and unproductive flower visits. Each set is arranged on a lattice with particle spacing $\Delta r = h/h_0$, where h_0 is a constant. The SPH formulation of model (20), (21) consists of equation (14) with $\mathbf{v}_{s'}(\mathbf{r}_i) = \mathbf{v}$ and $\mathbf{v}_{s'}(\mathbf{s}_j) = \mathbf{0}$ and the set of equations for $dx_{B_k}/dt|_{\mathbf{r}_i}$, $k = 0, 1, 2, 3$; $dx_{U_l}/dt|_{\mathbf{s}_j}$, $dx_{P_l}/dt|_{\mathbf{s}_j}$, $l = 1, 2$; and $dx_W/dt|_{\mathbf{s}_j}$ that are defined by equation (18). In the reaction rate terms of equation (18), the concentration of a species s that is tracked by one set of particles at a particle in the other set is approximated according to equation (13):

$$
\begin{aligned}
x_s(\mathbf{r}_i) &= \sum_{k=1}^{Q_F} \frac{x_s(\mathbf{s}_k)}{n_k^F} w(\mathbf{r}_i - \mathbf{s}_k, h), \ s \in \{U_l, P_l, W\} \\
x_s(\mathbf{s}_j) &= \sum_{k=1}^{Q_B} \frac{x_s(\mathbf{r}_k)}{n_k^B} w(\mathbf{s}_j - \mathbf{r}_k, h), \ s \in \{B_m\},
\end{aligned}
$$

where $l = 1, 2$, $m = 0, 1, 2, 3$, and n_k^P is the number of particles in set P per unit volume evaluated at particle k. We define the kernel w as a cubic spline with compact support and a shape similar to a Gaussian:

$$w(\mathbf{r}_{ij}, h) = \gamma \cdot \begin{cases} \frac{2}{3} - R^2 + \frac{1}{2}R^3 & \text{if} \quad 0 \le R < \kappa/2 \\ \frac{1}{6}(2 - R)^3 & \text{if} \quad \kappa/2 \le R < \kappa \\ 0 & \text{if} \quad R \ge \kappa \end{cases} \tag{22}$$

where $R = ||\mathbf{r}_{ij}||/h$ and $\gamma = \frac{15}{7\pi h^2}$ in two dimensions. This function has been widely used in the SPH literature [21].

Because the lattice spacings do not change over time, we can precompute the identities of a particle's neighbors in its lattice (*i.e.*, those within the support domain of the kernel w), as well as all quantities derived from its distance to its neighbors. In general, however, these variables must be updated at every iteration. The initial nonzero species concentrations are $x_{B_0}(\mathbf{r}_i) = 1$ at B particles within the hive region and $x_{U_k}(\mathbf{s}_j) = x_{U_k}^0$ at F particles in the rows of cultivar $k \in \{1, 2\}$. At each timestep, the particle positions and concentrations are computed by numerically integrating the SPH equations using the Euler method.

C. Results

For the analysis in this section, we ran the micro-continuous model and the SPH formulation of the macro-continuous model with $\Delta t = 0.002$ s. The micro-continuous model used a grid of 50×18 cells over the domain $[0\ 50]\ ft \times [0\ 18]\ ft$. In the SPH method, set B was comprised of a lattice of 53×35 particles that initially covered the domain $[-11\ 15]\ ft \times [0\ 17]\ ft$, and set F was a lattice of 101×37 particles that covered the domain $[0\ 50]\ ft \times [0\ 18]\ ft$. Using the suggested values of $h_0 = 1.2$ and $\kappa = 2$ [21] as a guideline, we defined $h_0 = 2$ and $\kappa = 2$.

1) Accuracy of the Macro-Continuous Model: To verify that the macro-continuous model is an accurate abstraction of the system, we compared concentrations computed by the micro- and macro-continuous models for the four parameter sets in Table I. The quantity $\sigma/(v\Delta t)$ is a ratio of diffusive to deterministic robot motion per timestep, where v and σ are subject to the flight speed constraint in Section IV-A. The micro-continuous model simulated 10^4 robotic bees. Fig. 2 and 3 show that there is a close match between the concentrations of B_3, P_1, P_2, and W computed by the two models at specified times for parameter sets 1 and 2. Note that raising $\sigma/(v\Delta t)$ results in a broader pollinated region and higher maximum concentrations of both pollinated flowers and unproductive flower visits. The first row of cultivar 1 is not pollinated because the robots visiting that row have not yet acquired pollen from cultivar 2.

We quantify the degree of closeness between the models with the error metric $\mu_s = \frac{1}{G}||\mathbf{x}_s^{micro} - \mathbf{x}_s^{macro}||_1$, where $\mathbf{x}_s^{micro}$ is the vector of concentrations of species s computed at each of the G cell centers in the micro-continuous model and $\mathbf{x}_s^{macro}$ is the corresponding vector computed at SPH particles in set F that overlap the cell centers. Fig. 4 shows that the concentration fields of P_1 and P_2 computed with

TABLE I
PARAMETERS USED IN THE MICRO- AND MACRO-CONTINUOUS MODELS

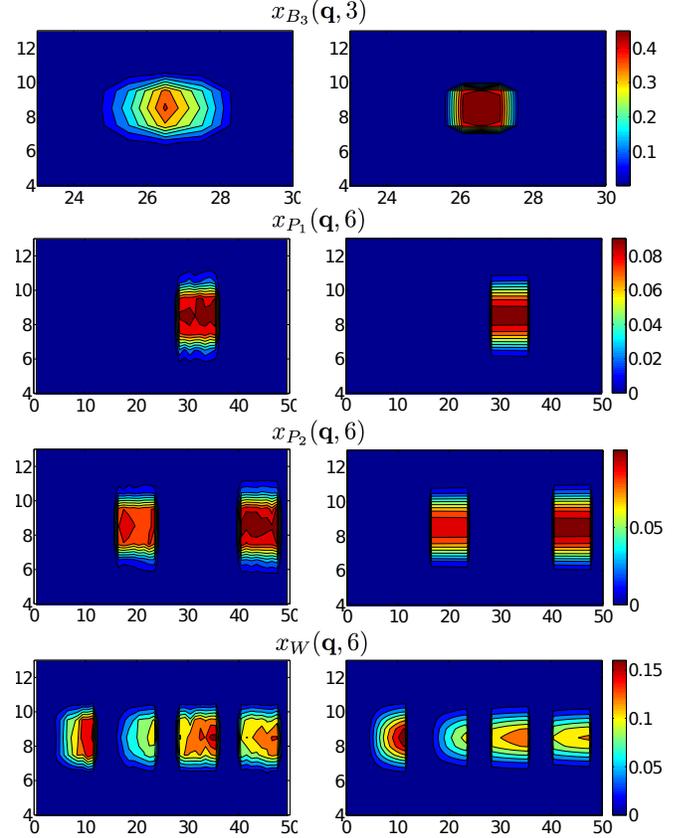| Set	$v\ (ft/s)$	$D\ (ft^2/s)$	k	$\sigma/(v\Delta t)$
Par 1	8.20	0.01681	10	0.5
Par 2	3.28	0.04303	10	2
Par 3	8.20	0.01681	5	0.5
Par 4	3.28	0.04303	5	2

Fig. 2. Snapshots of the concentration distributions of B_3 at time $t = 3$ s and P_1, P_2, and W at $t = 6$ s over subsets of the domain $[0\ 50]\ ft \times [0\ 18]\ ft$. Concentrations were computed for parameter set 1 using the micro-continuous model (left column) with 10^4 robots and the SPH method (right column).

parameter sets 3 and 4 have similar degrees of closeness to those that are visually compared in Fig. 2 and 3, respectively.

2) Macro-Continuous Model as a Tool for Top-Down Design: The correspondence between the micro- and macro-continuous models demonstrates that the macro-continuous model can be used to design the parameters v, D, and k that will produce desired collective behaviors in the physical system. The advantage of using the macro-continuous model for this purpose is that, unlike the micro-continuous model, its computational time is invariant to the sizes of the species populations. Fig. 5 demonstrates this advantage with the ratio of τ_{micro}, the runtime on a standard 2.53 GHz laptop per iteration of the micro-continuous model averaged over 201 iterations in the interval $t = 1.6 - 2.0$ s, to $\tau_{macro} = 1.42$ s, the same quantity in the macro-continuous model. As the number of robots N increases above ~ 3000, it is faster to run the macro-continuous model.
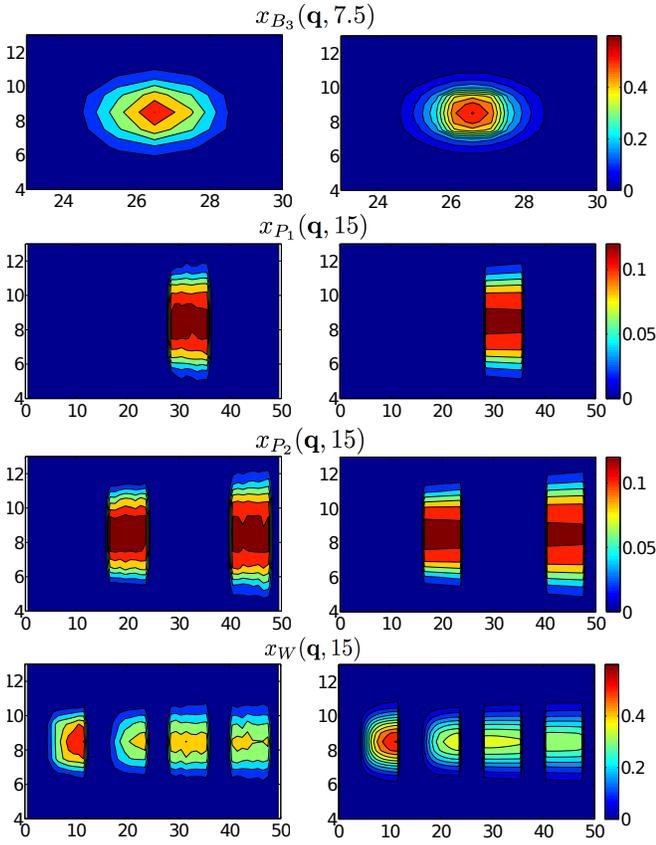
$x_{B_3}(\mathbf{q}, 7.5)$

$x_{P_1}(\mathbf{q}, 15)$

$x_{P_2}(\mathbf{q}, 15)$

$x_W(\mathbf{q}, 15)$

Fig. 3. Snapshots of the concentration distributions of B_3 at time $t = 7.5\ s$ and P_1, P_2, and W at $t = 15\ s$ over subsets of the domain $[0\ 50]\ ft \times [0\ 18]\ ft$. Concentrations were computed for parameter set 2 using the micro-continuous model (left column) with 10^4 robots and the SPH method (right column).

Fig. 4. Normalized errors μ_{P_1}, μ_{P_2} over time for all four parameter sets and 10^4 robots. Each plot ends at a time t when the robotic bees have covered an average x distance slightly larger than $vt = 50\ ft$, the width of the field.

Fig. 5. Ratio of averaged runtimes per iteration in the micro- and macro-continuous models versus number of robots N for parameter set 1.

Hence, for very large populations, it is more suitable to use the macro-continuous model in an optimization technique such as a Monte Carlo method to compute the parameters that maximize some metric of success. We illustrate this concept by using the macro-continuous model to find the value of k that maximizes a metric of pollination efficiency for fixed v and D. Computing the mass of species $s \in \{U_1, U_2, P_1, P_2, W\}$ as $M_s = \sum_{j=1}^{Q_F} x_s(\mathbf{s}_j)/n_j^F$, the total mass of flowers is $M_0 \equiv M_{U_1} + M_{U_2} + M_{P_1} + M_{P_2}$. We define η_{poll} as the mass fraction of pollinated flowers, $(M_{P_1} + M_{P_2})/M_0$, and η_{waste} as the ratio of the "mass" of unproductive flower visits to the total mass of flowers, M_W/M_0. Thus, η_{poll}/η_{waste} is a measure of the number of pollination events to the number of unproductive visits. As Fig. 6 shows, this metric is a maximum at $k = 3$ and then decreases below 1 for $k > 9$, since η_{waste} increases faster than η_{poll}.

V. CONCLUSIONS AND FUTURE WORK

We have described an approach to synthesizing robot control policies that produce a target collective behavior in a spatially inhomogeneous swarm that emulates an advection-diffusion-reaction chemical system. Our approach relies on a rigorous correspondence between two models of a swarm.
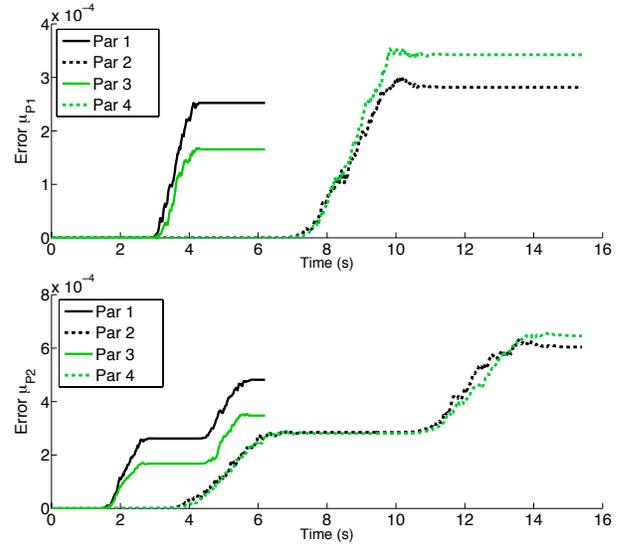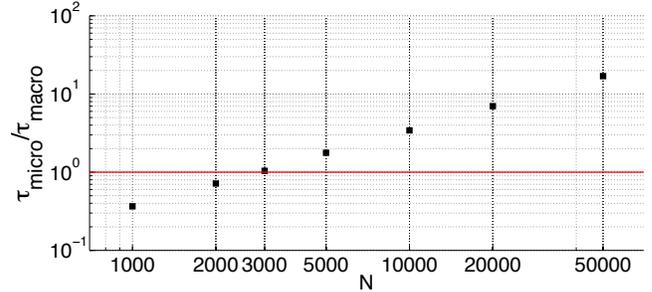
The micro-continuous model offers a realistic representation of individual robot activities and captures variability in macroscopic outcomes that an actual system would produce over multiple trials. However, it becomes more computationally intensive to run as the swarm population is increased, and studying the effect of parameter variation requires simulations under many different conditions. These factors motivate us to abstract the system to a macro-continuous model, which is independent of the population size and amenable to techniques for analysis, control, and optimization that yield theoretical guarantees on performance. This model becomes a more accurate description as the swarm population increases, and it does not capture variability in performance. Using the application of commercial pollination with robotic insects, we demonstrate the utility of this model in designing the parameters that govern the deterministic and random components of the robot motion and the robot probabilistic task transitions.

Future work includes the investigation of analysis and controller synthesis methodologies for the PDE macro-continuous model. These may include nonlinear stability
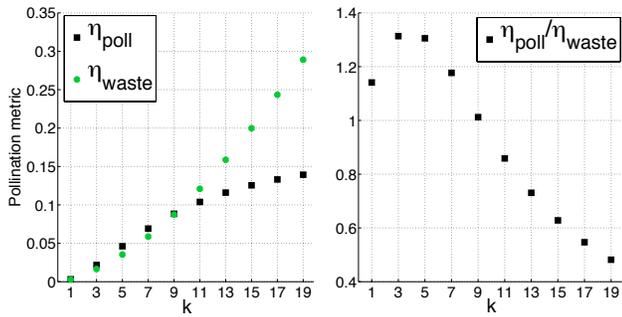
Fig. 6. Values of pollination metrics η_{poll}, η_{waste}, and η_{poll}/η_{waste} computed at $t = 6.2$ s using the macro-continuous model over a range of k with $v = 8.20$ ft/s, $D = 0.01681$ ft^2/s.

analysis, nonlinear and robust control methods for parabolic PDE's, and stochastic optimization techniques. Parameter optimization in higher dimensions can be facilitated by speeding the simulation execution time with parallelization of the SPH code [21] and the optimization technique.

We are also interested in applying our controller synthesis methodology to expanded models of the pollination scenario. We would like to model different hive placements, more complex velocity fields, time delays associated with pollination, and inter-robot communication. We can add unimolecular reactions that represent a robotic bee failing in the field or returning to the hive at a time-dependent rate that is related to its energy consumption. The flower visit rate constants can be modified to vary with time, depend on the presence of pollen on a robot, and incorporate the probability of fertilization given pollination [34]. Finally, we would like to include feedback from the robots about their flower visits upon their return to the hive. The hive computer would use this data to recalculate the parameters for the next round of flights to produce robot behavior that compensates for pollination deficiencies due to environmental disturbances and robot failures.

REFERENCES

[1] *Robobees Project*. http://robobees.seas.harvard.edu/, 2011.
[2] A. Bagtzoglou, A. Tompson, and D. Dougherty. Projection functions for particle-grid methods. *Numer. Meth. Part. Diff. Eq.*, 8(4):325–340, 1992.
[3] P. Bailis, K. Dantu, B. Kate, J. Waterman, and M. Welsh. KARMA: A distributed operating system for micro-UAV swarms (poster abstract). In *Proc. Operating Systems Design and Implementation (OSDI)*, 2010.
[4] D. A. Benson and M. M. Meerschaert. Simulation of chemical reaction via particle tracking: Diffusion-limited versus thermodynamic rate-limited regimes. *Water Resour. Res.*, 44:W12201, 2008.
[5] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar. Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, 25(4):927–937, Aug. 2009.
[6] T. J. Chung. *Computational Fluid Dynamics*. Cambridge Univ. Press, Cambridge, 2002.
[7] S. A. Combes and R. Dudley. Turbulence-driven instabilities limit insect flight performance. *PNAS*, 106(22):9105–9108, 2009.
[8] W. M. Deen. *Analysis of Transport Phenomena*. Oxford University Press, New York, 1998.
[9] K. S. Delaplane and D. F. Mayer. *Crop Pollination by Bees*. CABI Publishing, New York, NY, 2000.
[10] F. Delay, P. Ackerer, and C. Danquigny. Simulating solute transport in porous or fractured formations using random walk particle tracking: A review. *Vadose Zone Journal*, 4:360–379, 2005.
[11] J. B. Free. *Insect Pollination Of Crops*. Academic Press, London, UK, 2nd edition, 1993.
[12] C. Gardiner. *Stochastic Methods: A Handbook for the Natural and Social Sciences*. Springer, 4th edition, 2009.
[13] D. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22:403–434, 1976.
[14] D Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188(1-3):404–425, 1992.
[15] H. Hamann and H. Wörn. A framework of space-time continuous models for algorithm design in swarm robotics. *Swarm Intelligence*, 2(2-4):209–239, 2008.
[16] P. A. Herrera, M. Massabo, and R. D. Beckie. A meshless method to simulate solute transport in heterogeneous porous media. *Advances in Water Resources*, 32:413–429, 2009.
[17] W. Hundsdorfer and J. G. Verwer. *Numerical Solutions of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer-Verlag, New York, 2003.
[18] M. Jubelgas, V. Springel, and K. Dolag. Thermal conduction in cosmological SPH simulations. *Mon. Not. Roy. Astron. Soc.*, 351:423–435, 2004.
[19] M. Karpelson, J. P. Whitney, G.-Y. Wei, and R. J. Wood. Energetics of flapping-wing robotic insects: towards autonomous hovering flight. In *Int'l. Conf. Intell. Robots and Syst. (IROS)*, pages 1630–1637, 2010.
[20] G. R. Liu. *Meshfree Methods: Moving Beyond the Finite Element Method*. CRC Press, Boca Raton, FL, 2nd edition, 2010.
[21] G. R. Liu and M. B. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific Publishing Comp., 2003.
[22] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics: An overview and recent developments. *Arch. Comp. Methods Eng.*, 17:25–76, 2010.
[23] P. M. Lyrene and T. E. Crocker. Poor fruit set on rabbiteye blueberries after mild winters: Possible causes and remedies. In *Proc. Fla. State Hort. Soc.*, volume 96, pages 195–197, 1983.
[24] L. Matthey, S. Berman, and V. Kumar. Stochastic strategies for a swarm robotic assembly system. In *Proc. Int'l. Conf. on Robotics and Automation (ICRA)*, pages 1953–1958, 2009.
[25] S. E. McGregor. *Insect Pollination Of Cultivated Crop Plants*. USDA AGRIC, 1976. Handbook 496.
[26] N. Michael, J. Fink, S. Loizou, and V. Kumar. Architecture, abstractions, and algorithms for controlling large teams of robots: Experimental testbed and results. *Proc. Int'l. Symposium of Robotics Research (ISRR)*, 2007.
[27] J. J. Monaghan. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.*, 30:543–574, 1992.
[28] M. R. Pac, A. M. Erkmen, and I. Erkmen. Control of robotic swarm behaviors based on smoothed particle hydrodynamics. In *Int'l. Conf. Intell. Robots and Syst. (IROS)*, pages 4194–4200, 2007.
[29] J. R. Perkinson and B. Shafai. A decentralized control algorithm for scalable robotic swarms based on mesh-free particle hydrodynamics. In *Proc. IASTED Int'l. Conf. on Robotics and Appl.*, pages 1–6, 2005.
[30] L. Pimenta, M. Mendes, R. Mesquita, and G. Pereira. Fluids in electrostatic fields: An analogy for multirobot control. *IEEE Trans. on Magnetics*, 43(4):1765–1768, 2007.
[31] L. Pimenta, N. Michael, R. Mesquita, G. Pereira, and V. Kumar. Control of swarms based on hydrodynamic models. In *Proc. Int'l. Conf. on Robotics and Automation (ICRA)*, pages 1948–1953, 2008.
[32] A. Powell, W. A. Dozier Jr., and D. G. Himelrick. Commercial blueberry production guide for Alabama. 2002. Alabama Cooperative Extension System, ANR-904.
[33] P. Salamon, D. Fernandez-Garcia, and J. J. Gomez-Hernandez. A review and numerical assessment of the random walk particle tracking method. *J. of Contaminant Hydrology*, 87:277–305, 2006.
[34] B. J. Sampson and J. H. Cane. Pollination efficiencies of three bee (Hymenoptera: Apoidea) species visiting rabbiteye blueberry. *J. Econ. Entomol.*, 93(6):1726–1731, 2000.
[35] A. M. Tartakovsky, T. D. Scheibe, G. Redden, P. Meakin, and Y. Fang. Smoothed particle hydrodynamics model for reactive transport and mineral precipitation. In *Int'l Conference on Computational Methods in Water Resources (CMWR XVI)*, Copenhagen, Denmark, 2006.
[36] A. F. B. Tompson. Particle-grid methods for reacting flows in porous media with application to Fisher's equation. *Appl. Math. Modelling*, 16:374–383, 1992.
[37] R. J. Wood. The first takeoff of a biologically inspired at-scale robotic insect. *IEEE Trans. Robotics*, 24(2):341–347, 2008.