Leader-Enabled Deployment Onto Planar Curves: A PDE-Based Approach

Paul Frihauf, *Student Member, IEEE*, and Miroslav Krstic, *Fellow, IEEE*

***Abstract*—We introduce an approach for stable deployment of agents onto families of planar curves, namely, 1-D formations in 2-D space. The agents' collective dynamics are modeled by the reaction–advection–diffusion class of partial differential equations (PDEs), which is a broader class than the standard heat equation and generates a rich geometric family of deployment curves. The PDE models, whose state is the position of the agents, incorporate the agents' feedback laws, which are designed based on a *spatial* internal model principle. Namely, the agents' feedback laws allow the agents to deploy to a family of geometric curves that correspond to the model's equilibrium curves, parameterized by the continuous agent identity $\alpha \in [0,1]$. However, many of these curves are open-loop unstable. Stable deployment is ensured by *leader* feedback, designed in a manner similar to the boundary control of PDEs. By discretizing the PDE model with respect to α, we impose a fixed communication topology, specifically a chain graph, on the agents and obtain control laws that require communication with only an agent's nearest neighbors on the graph. A PDE-based approach is also used to design observers to estimate the positions of all the agents, which are needed in the leader's feedback, by measuring only the position of the leader's nearest neighbor. Hence, the leader uses only local information when employing output feedback.**

***Index Terms*—Boundary control, cooperative control, multiagent systems.**

I. Introduction

MUCH research has been conducted in multiagent formation control, leading to many approaches for stable deployment onto curves. However, the agents in many of these works implement controllers that depend on the desired deployment, hence the parameters of each agent's controller must be updated to move the agents from one deployment to another, which may be cumbersome for systems with large numbers of agents.

We propose a framework that enables agents to achieve deployment *families* while employing a *single* controller with no knowledge of the desired deployment. These families correspond to the potentially unstable, nonzero equilibria of either two decoupled, linear reaction–advection–diffusion partial differential equations (PDEs) or one complex-valued, linear Ginzburg–Landau PDE and are stabilized by a leader agent. The agents' positions, needed for the leader's feedback, are estimated by observers that require position information from only the leader's nearest neighbor. Thus, when the leader employs output feedback, all the agents use only local information. We also propose a nonlinear approach for deployment to a family of circular arcs, including circles, that does not require leader feedback. This design, however, is limited to formations of fixed radius, whereas the leader-enabled design allows the agents to stabilize formations of any size with an arbitrary convergence rate. For large multiagent systems, our framework allows a user to deploy *many* agents to multiple configurations while communicating with only *two* agents.

1) Literature Review: This paper draws from multiagent systems research in formation control, estimation, and PDE-related designs. In formation control, feasible geometric patterns are characterized for agents with global information in [1], and stabilization of any geometric pattern using Laplacian controls are studied in [2]–[4]. Formation control for unicycles under cyclic pursuit is considered in [5], [6], a sequence of maneuvers between formation patterns is achieved with a behavior-based approach in [7], and planar formations are controlled in a Lie group setting in [8]. By selecting appropriate density functions that are known by all the agents, coverage control algorithms designed in [9] can be used to achieve deployments onto a desired planar curve and 2-D distributions within a desired planar curve. In [10], decentralized controllers that maintain connectivity are used to form geometric patterns specified by a smooth function. Deployment and rendezvous on a line are considered in [11] for agents connected by a data-rate-constrained network.

Other works use a leader agent to influence the collective behavior of the agents. Artificial potentials and virtual leaders are used to control the group's geometry and mission in [12]. In leader–follower systems, nonholonomic followers use nonlinear controllers to stabilize their relative distances and orientation in [13], leader-to-formation stability gains are used to quantify a formation's stability properties in [14], and bounds on a leader's velocity and the curvature of its path, which guarantee the existence of a follower's formation-maintaining controller, are determined in [15]. A leader agent is used to steer a formation in [4] and to optimally transfer agents to desired waypoints at specified times in [16]. In [17], leaders employ a hybrid Stop–Go policy to drive follower agents to a target location.

Multiagent estimation research has focused mainly on dynamic consensus filters. Vehicles use an information exchange methodology, whose stability is decoupled from the local control of the vehicles, to reach consensus on a formation's center

Manuscript received August 31, 2009; revised May 18, 2010; accepted October 06, 2010. Date of publication November 15, 2010; date of current version August 03, 2011. This work was supported by the Department of Defense (DoD), the Air Force Office of Scientific Research, the National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a, the National Science Foundation, and the Los Alamos National Laboratory. Recommended by Associate Editor M. Egerstedt.

The authors are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: pfrihauf@ucsd.edu; krstic@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2010.2092210

in [2]. Laplacian consensus dynamics are extended in [18] to handle time-varying signals, arbitrary time delays, and splitting and merging networks. Dynamic consensus filters are developed in [19] and implemented in [20] to estimate global information for use in an agent's local controller to achieve the system's desired global behavior. Dynamic consensus filters are also used in [21] to stabilize parallel or circular collective motion, recovering the results in [22].

Recent years have seen an increase in multiagent research utilizing PDEs for both design and analysis [23]–[28]. In particular, [24] uses a PDE from image processing to design boundary-tracking controllers, [26] models a swarm as an incompressible fluid for pattern generation, and [27] studies the stability of large vehicular platoons using a linear hyperbolic PDE. The Partial difference Equation (PdE) framework is used in [29] to show that Laplacian control, analyzed in [30] and [31], coincides with the heat equation. It is also used to develop control laws in [17] and [32], where in [32], agents use model reference adaptive control laws to track desired trajectories, using either the heat equation or the wave equation as reference models.

2) Results of This Paper: We introduce a framework for multiagent deployment into *families of geometric curves*. Our design employs linear reaction–advection–diffusion equations and boundary control techniques, which treat the agents as a continuum. These PDE models are an application of the internal model principle [33], but in a *spatial* sense, where the PDE models allow the agents to achieve a family of deployments that correspond to the models' nonzero equilibria. Specifically, the follower agents' feedback laws, incorporated by the PDE models, contain a model for a family of geometric curves, namely, signals in space rather than time. Consequently, the agents are able to achieve different formations while using the same controller, whereas other works require a controller's parameters to be changed to achieve a new deployment.

The follower agents' feedback laws, however, do not ensure stability, and many, if not most, nonelementary deployments are derived from unstable PDE models. Other PDE-based designs utilize only inherently stable models. Stabilization of the planar curves is guaranteed by two boundary agents—the *leader* agent and the *anchor* agent—which serve as boundary conditions for the PDE models. These special agents execute control laws designed using the backstepping approach [34]. Even for standard deployments, which correspond to stable PDEs, such as rendezvous or deployment to a line, our leader feedback can achieve any desired convergence speed, in contrast to the convergence speeds of the standard consensus-based algorithms that are limited by the first eigenvalue of the heat equation [2], [11], [29]. The desired deployment shape is encoded in these boundary controls in the form of a bias term, which allows the leader and anchor to select a specific curve from the deployment family for the agents to stabilize. By adjusting their respective bias terms, the leader and anchor (and by extension, the user) can induce the agents to unknowingly deploy to other curves within the deployment family.

Our framework also includes the design of observers, which are employed by the leader agent, using the backstepping approach for PDEs with boundary sensing [35]. These observers require knowledge of only the position of the leader's nearest neighbor to estimate the positions of all the agents. By spatially discretizing the PDE models, observers, and boundary controllers, we obtain control laws for the follower agents, the anchor, and the leader. This discretization imposes a fixed communication topology on the agents, shown in Fig. 1, where if the leader employs output feedback, all the agents utilize only local information (in the sense of the communication topology) in their control laws.

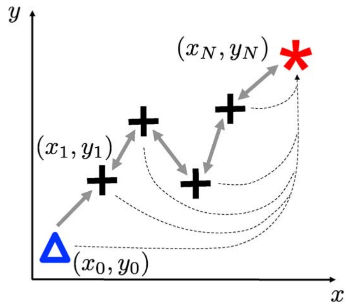

Fig. 1. Communication topology imposed by spatial discretization. The leader needs global information (dashed) unless it uses output feedback. Legend: $\triangle$ = anchor; $+$ = follower; $\star$ = leader.

The unstable PDEs place special emphasis on the leader for stability. For this reason, we explore alternative ways to achieve deployment families. We focus on circles since they correspond to unstable PDE models in the leader-based design and are important benchmarks that extend the deployment ideas beyond linear deployments [3], [11]. We present a nonlinear PDE model for deployment to a circle and prove its stability. We use anchor agents to move the agents from one deployment to another, similar to the use of leading reference agents in [32]. A limitation of this method is that the deployment's radius cannot be manipulated by the anchors (or user) without communicating the desired radius to all the agents. In contrast, the leader-based approach allows deployment to circles/ellipses of arbitrary size without the need to broadcast any parameters.

3) Organization: Section II introduces leader-enabled deployment for both decoupled 1-D deployments and complex-valued deployments with x–y cross-coupling. Section III details the design of stabilizing controllers with closed-loop stability proven in Section IV. Observer design, model discretization, and numerical simulations are presented in Sections V–VII. We introduce deployment to circular arc families in Section VIII and conclude with Section IX.

II. Leader-Enabled Deployment

For the planar deployment problem, we consider a large (continuum) group of fully actuated agents operating in a common reference frame, namely, we consider the dynamical model

$$x_t(\alpha, t) = u(\alpha, t) \quad y_t(\alpha, t) = v(\alpha, t) \tag{1}$$

where $(x(\alpha, t), y(\alpha, t))$ denotes the position of agent α at time t, $u(\alpha, t)$ and $v(\alpha, t)$ are the control inputs for agent α, $x_t(\alpha, t) = (\partial/\partial t)x(\alpha, t)$, $y_t(\alpha, t) = (\partial/\partial t)y(\alpha, t)$, and $\alpha \in [0, 1]$. We refer to the parameter α as the agent identity, which serves as an agent's identification number and as the spatial variable of a PDE model for the group's collective dynamics. Discretizing (1) with respect to α leads to the following dynamical model for agent i: $\dot{x}_i(t) = u_i(t)$, $\dot{y}_i(t) = v_i(t)$, $i \in \{0, \ldots, n\}$. For later use, we define the notation: $x_\alpha(\alpha, t) = (\partial/\partial\alpha)x(\alpha, t)$, $x_{\alpha\alpha}(\alpha, t) = (\partial^2/\partial\alpha^2)x(\alpha, t)$.

Our goal is to stably deploy a continuum of agents to families of planar curves by designing the controllers $u(\alpha, t)$, $v(\alpha, t)$. Then, a finite number of agents implement the discretized controllers $u_i(t)$, $v_i(t)$. This 2-D deployment problem can be approached either through: 1) two decoupled 1-D deployment problems, where the horizontal feedback is decoupled from vertical feedback, i.e., actuation in the x-direction does not depend on the position measurement in the y-direction, and vice-versa; or 2) as a single complex-valued deployment, where the real and imaginary components represent the horizontal and vertical coordinates, and actuation in each coordinate direction depends on the entire position vector. Restated, in 1), the horizontal velocity command is a function of only the x-position, and the vertical velocity command is a function of only the y-position. In 2), the complex-valued formulation allows for horizontal and vertical velocity commands that are functions of the planar position (x, y). Using two decoupled 1-D deployments is simpler than employing a single complex-valued deployment, so we consider it first for clarity.

A. Decoupled 1-D Deployments

It is common to approach the deployment problem through consensus-based control laws [2]–[4], [11], whose basic form is given by

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} (x_j(t) - x_i(t)) \tag{2}$$

where $\mathcal{N}_i$ denotes the set of agents/neighbors that communicate with agent i. In [29], (2) is formally shown to coincide with the heat equation

$$x_t(\alpha, t) = x_{\alpha\alpha}(\alpha, t) \tag{3}$$

where each agent employs the diffusion-based feedback $u(\alpha, t) = x_{\alpha\alpha}(\alpha, t)$, which depends only on local agent interactions, i.e., an agent's nearest neighbors. This simple agent strategy is stable, but it is limited in its convergence rate and is capable of achieving only linear formations [because the equilibrium equation is the simplest second-order ordinary differential equation (ODE), $\bar{x}''(\alpha) = 0$].

Remark 2.1: Throughout this paper, "nearest-neighbor" refers to agents that are nearest in terms of the fixed communication topology, not in terms of physical distance.

Drawing from the connection between consensus and the heat equation, we approach the PDE-based deployment with the more general linear reaction–advection–diffusion equation

$$x_t(\alpha, t) = x_{\alpha\alpha}(\alpha, t) + b x_\alpha(\alpha, t) + \lambda x(\alpha, t) \tag{4}$$

where the agents' velocity-actuated feedback laws $u(\alpha, t)$ are given by the right-hand side of (4), and $v(\alpha, t)$ follows analogously for the y-dimension. These feedback laws maintain the simplicity of the diffusion-based feedback as they are still based only on nearest-neighbor information with all the agents applying the same constant gains b and λ. In the sequel, we drop the arguments (α, t) whenever the context allows us to do so without harming clarity.

We designate a special role for the two boundary agents, i.e., agent $\alpha = 0$ and agent $\alpha = 1$, whose motions are governed by

$$x_t(0, t) = U_0(t) \quad x_t(1, t) = U_1(t) \tag{5}$$

where $U_0(t)$ and $U_1(t)$ are controls to be designed, and which act as the boundary conditions for the PDE (4). The *leader* agent $(\alpha = 1)$ and the *anchor* agent $(\alpha = 0)$ will control the follower agents $(0 < \alpha < 1)$. As indicated by their names, the leader stabilizes the deployment profile $\bar{x}(\alpha)$, while the anchor simply deploys to its designated position $\bar{x}(0)$. Either the leader agent or both the leader and anchor agents may be treated as virtual agents if desired (and as suggested by the use of virtual edge leaders in [3]), but it is not necessary.

The deployment families of interest correspond to the nonzero equilibrium curves of (4), which satisfy the two-point boundary value problem

$$\bar{x}''(\alpha) + b\bar{x}'(\alpha) + \lambda\bar{x}(\alpha) = 0 \tag{6}$$

with $\bar{x}(0)$ and $\bar{x}(1)$ given. This allows for a much more general family of deployments than the linear (in α) equilibrium curves of the heat (3). Equation (6), which is a second-order ODE with constant coefficients, characterizes all the achievable 1-D deployments with the follower agent feedbacks (4). While these feedbacks make these deployments feasible, they do not guarantee stability since the open-loop response of (4) is $x(\alpha, t) = \sum_{k=1}^{\infty} C_k e^{\chi t} \sin(k\pi\alpha)$ with eigenvalues $\chi = \lambda - b^2/4 - \pi^2 k^2$, $k \in \{1, 2, \ldots\}$, and constants C_k, whose values depend on the initial condition $x(\alpha, 0)$. In particular, deployment families where $\lambda > b^2/4 + \pi^2$ are unstable. Hence, the leader and the anchor agents play a crucial role in stabilizing the possibly nonlinear (in α) deployment curves.

For planar deployment, we utilize a 1-D PDE model for each coordinate axis, which yields two deployments, $\bar{x}(\alpha)$ and $\bar{y}(\alpha)$, that characterize a planar curve parameterized in α

$$\bar{x}(\alpha) = a_0\psi_0(\alpha) + a_1\psi_1(\alpha) \quad \bar{y}(\alpha) = a_2\psi_2(\alpha) + a_3\psi_3(\alpha) \tag{7}$$

where (ψ_0, ψ_1) and (ψ_2, ψ_3) are basis functions associated with the solutions of (6) for the respective horizontal and vertical PDE models. We term the coefficients, a_0, a_1, a_2, and a_3 the *deployment coefficients*, which are scalars the user is free to select to define a desired deployment. It is of interest to see how rich the family of possible geometric curves is. Table I categorizes the basis functions according to the values of b and λ. To the user, who has particular planar formations in mind, the basis functions are a starting point in selecting the strategies of the follower agents and also of the leader and anchor agents.

The ability to use two disparate PDE models (one for each dimension) provides the user with a wide variety of basis-function combinations that produce various planar deployments. Interestingly, the well-known Lissajous curves, given by $\bar{x}(\alpha) = A\sin(C\alpha + E)$, $\bar{y}(\alpha) = B\sin(D\alpha)$, where A, B, C, D, and E are scalars and C, $D > 0$, are achieved when the 1-D deployments are governed by the reaction–diffusion equations, $x_t =$

TABLE I
BASIS FUNCTIONS FOR 1-D DEPLOYMENT CURVES OF THE REACTION–ADVECTION–DIFFUSION EQUATION

b, λ **Values**	**Basis Functions** $(\psi_0(\alpha),\ \psi_1(\alpha))$	
$b = \lambda = 0$	$(1,\ \alpha)$	
$b \neq 0,\ \lambda = 0$	$(1,\ e^{-b\alpha})$	
$b = 0,\ \lambda > 0$	$(\cos(\theta\alpha),\ \sin(\theta\alpha))$,	$\theta = \sqrt{\lambda}$
$b = 0,\ \lambda < 0$	$(\cosh(\sigma\alpha),\ \sinh(\sigma\alpha))$,	$\sigma = \sqrt{-\lambda}$
$b^2 = 4\lambda$	$(e^{-\sigma\alpha},\ \alpha e^{-\sigma\alpha})$,	$\sigma = \sqrt{\lambda}$
$b^2 < 4\lambda$	$(e^{\sigma\alpha}\cos(\theta\alpha),\ e^{\sigma\alpha}\sin(\theta\alpha))$,	$\sigma = -\frac{b}{2}$, $\theta = \frac{1}{2}\sqrt{4\lambda - b^2}$
$b^2 > 4\lambda$	$(e^{\sigma_0\alpha},\ e^{\sigma_1\alpha})$,	$\sigma_0 = -\frac{b}{2} + \frac{1}{2}\sqrt{b^2 - 4\lambda}$, $\sigma_1 = -\frac{b}{2} - \frac{1}{2}\sqrt{b^2 - 4\lambda}$

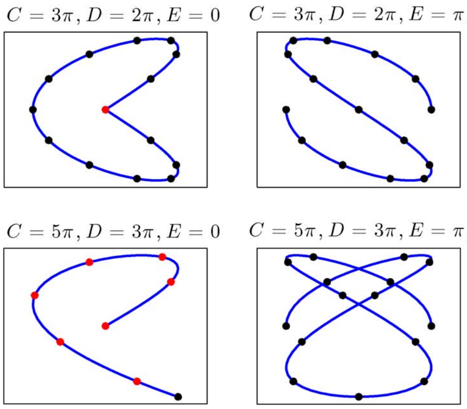

Fig. 2. Lissajous curves for various values of C, D, and E with $A = B = 1$, $\alpha \in [0, 1]$. The dots represent agents. Legend: black dots $=$ 1 agent; light dots $=$ 2 agents.

$x_{\alpha\alpha} + \sqrt{C}x$ and $y_t = y_{\alpha\alpha} + \sqrt{D}y$, and the following deployment coefficients are selected: $a_0 = A\sin E$, $a_1 = A\cos E$, $a_2 = 0$, and $a_3 = B$. Fig. 2 depicts four possible deployments of 15 agents based on Lissajous curves.

When the same PDE model is used in each dimension, the parameterized deployment can be written in vector form as

$$\begin{bmatrix} \bar{x}(\alpha) \\ \bar{y}(\alpha) \end{bmatrix} = \begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix} \begin{bmatrix} \psi_0(\alpha) \\ \psi_1(\alpha) \end{bmatrix} \tag{8}$$

and the coefficient matrix can be chosen to be a rotation, scaling, shear, or reflection matrix. For example, the coefficients can be selected to define the desired deployment

$$\begin{bmatrix} \bar{x}(\alpha) \\ \bar{y}(\alpha) \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} b_0\psi_0(\alpha) \\ b_1\psi_1(\alpha) \end{bmatrix} \tag{9}$$

which is a counterclockwise rotation of the scaled curve $\bar{x}(\alpha) = b_0\psi_0(\alpha)$, $\bar{y}(\alpha) = b_1\psi_1(\alpha)$ about the origin by the angle ϕ. When two identical reaction–diffusion equations with $\lambda > 0$ are used, $(\psi_0(\alpha), \psi_1(\alpha)) = (\cos(\sqrt{\lambda}\alpha), \sin(\sqrt{\lambda}\alpha))$, and (9) represents a rotated ellipse. If the same PDE models are used, but with $\lambda < 0$, $(\psi_0(\alpha), \psi_1(\alpha)) = (\cosh(\sqrt{-\lambda}\alpha), \sinh(\sqrt{-\lambda}\alpha))$, and (9) represents a rotated hyperbola.

B. Complex-Valued 2-D Deployment With x–y Cross-Coupling

Until now, we have considered the already rich family of planar deployments that are created by pairing two independent 1-D deployments. We now extend this family of achievable deployments by utilizing an agent's full position vector $z = (x, y)$ in a feedback law for each coordinate direction. To do so, we consider the complex-valued Ginzburg–Landau PDE as a continuum model of the collective dynamics of the agents in the plane.

Let $z(\alpha, t) = x(\alpha, t) + jy(\alpha, t)$ be the complex-valued position at time t of agent α, where j denotes the imaginary unit, $\sqrt{-1}$. Now, consider the complex-valued reaction–advection–diffusion equation (which is a linear Ginzburg–Landau PDE with constant coefficients)

$$\begin{aligned} z_t(\alpha, t) = &(\varepsilon_R + j\varepsilon_I)z_{\alpha\alpha}(\alpha, t) + (b_R + jb_I)z_\alpha(\alpha, t) \\ &+ (\lambda_R + j\lambda_I)z(\alpha, t) \end{aligned} \tag{10}$$

where $\varepsilon_R > 0$. In the sequel, we use $\varepsilon = \varepsilon_R + j\varepsilon_I$, $b = b_R + jb_I$, and $\lambda = \lambda_R + j\lambda_I$ for conciseness. Equation (10) represents the followers' velocity-actuated feedback laws. As before, the leader and anchor agents serve as the boundary conditions for the PDE (10)

$$z_t(0, t) = U_0(t) \quad z_t(1, t) = U_1(t) \tag{11}$$

where $U_0(t)$ and $U_1(t)$ are controls to be designed. The open-loop system is unstable when λ_R is positive and large.

The deployments associated with (10) are the equilibrium curves that satisfy the complex-valued two-point boundary value problem

$$\varepsilon\bar{z}''(\alpha) + b\bar{z}'(\alpha) + \lambda\bar{z}(\alpha) = 0 \tag{12}$$

where $\bar{z}(0)$ and $\bar{z}(1)$ are given. The second-order complex ODE (12) is in fact a fourth-order real ODE, whose solution is given in terms of four basis functions as $\bar{x}(\alpha) + j\bar{y}(\alpha) = (a_0 + ja_1)(\psi_0(\alpha) + j\psi_1(\alpha)) + (a_2 + ja_3)(\psi_2(\alpha) + j\psi_3(\alpha))$, alternatively written as

$$\begin{bmatrix} \bar{x}(\alpha) \\ \bar{y}(\alpha) \end{bmatrix} = \begin{bmatrix} a_0 & -a_1 \\ a_1 & a_0 \end{bmatrix} \begin{bmatrix} \psi_0(\alpha) \\ \psi_1(\alpha) \end{bmatrix} + \begin{bmatrix} a_2 & -a_3 \\ a_3 & a_2 \end{bmatrix} \begin{bmatrix} \psi_2(\alpha) \\ \psi_3(\alpha) \end{bmatrix}. \tag{13}$$

The presence of four basis functions affords the designer additional flexibility in shaping deployments, but the restrictive structure of the matrices in (13) prevents the user from being able to shear, reflect, or scale disproportionately the formations. The deployments can only be rotated or equally scaled.

For the second-order complex-valued ODE (12), the resulting basis functions are not easy to categorize in terms of the values of the real and imaginary parts of ε, b, λ, as was done in Table I for real second-order ODEs. We characterize the basis functions for specific subclasses of the complex-valued reaction–advection–diffusion equation. For $b = \lambda = 0$, the equilibrium profiles are linear in α, regardless of the value of ε. The more interesting cases are characterized next.

1) Advection–Diffusion Equation ($\lambda = 0$)*:*

$$\begin{bmatrix}\bar{x}(\alpha)\\ \bar{y}(\alpha)\end{bmatrix} = \begin{bmatrix}a_0 & -a_1\\ a_1 & a_0\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} + \begin{bmatrix}a_2 & -a_3\\ a_3 & a_2\end{bmatrix}\begin{bmatrix}\sin(\theta\alpha)\\ \cos(\theta\alpha)\end{bmatrix} e^{-\sigma\alpha} \tag{14}$$

where $\theta = (\varepsilon_R b_I - b_R \varepsilon_I)/|\varepsilon|^2$, $\sigma = (\varepsilon_R b_R + \varepsilon_I b_I)/|\varepsilon|^2$, and $|\varepsilon| = \sqrt{\varepsilon_R^2 + \varepsilon_I^2}$. If $\varepsilon_R b_I = b_R \varepsilon_I$, the deployment reduces to two independent 1-D profiles.

2) Reaction–Diffusion Equation ($b = 0$)*:*

$$\begin{bmatrix}\bar{x}(\alpha)\\ \bar{y}(\alpha)\end{bmatrix} = \begin{bmatrix}a_0 & -a_1\\ a_1 & a_0\end{bmatrix}\begin{bmatrix}\sin(\theta\alpha)\sinh(\sigma\alpha)\\ \cos(\theta\alpha)\cosh(\sigma\alpha)\end{bmatrix} + \begin{bmatrix}a_2 & -a_3\\ a_3 & a_2\end{bmatrix}\begin{bmatrix}\sin(\theta\alpha)\cosh(\sigma\alpha)\\ \cos(\theta\alpha)\sinh(\sigma\alpha)\end{bmatrix} \tag{15}$$

where $\theta = (1/\sqrt{2}|\varepsilon|)\sqrt{|\varepsilon||\lambda| + \varepsilon_R\lambda_R + \lambda_I\varepsilon_I}$, $\sigma = (1/\sqrt{2}|\varepsilon|)\mathrm{sgn}(\varepsilon_R\lambda_I - \varepsilon_I\lambda_R)\sqrt{|\varepsilon||\lambda| - \varepsilon_R\lambda_R - \lambda_I\varepsilon_I}$, and sgn denotes the signum function. The deployments become decoupled if $\varepsilon_R\lambda_I = \lambda_R\varepsilon_I$.

3) Reaction–Advection–Diffusion Equation:

$$\begin{bmatrix}\bar{x}(\alpha)\\ \bar{y}(\alpha)\end{bmatrix} = \begin{bmatrix}a_0 & -a_1\\ a_1 & a_0\end{bmatrix}\begin{bmatrix}\cos(\theta_0\alpha)\\ \sin(\theta_0\alpha)\end{bmatrix} e^{\sigma_0\alpha} + \begin{bmatrix}a_2 & -a_3\\ a_3 & a_2\end{bmatrix}\begin{bmatrix}\cos(\theta_1\alpha)\\ \sin(\theta_1\alpha)\end{bmatrix} e^{\sigma_1\alpha} \tag{16}$$

where $\theta_0 = -(1/2\sqrt{2}|\varepsilon|^2)(\sqrt{2}(\varepsilon_R b_I - \varepsilon_I b_R) - \varepsilon_R \mathrm{sgn}(a_I)\sqrt{|a| - a_R} + \varepsilon_I\sqrt{|a| + a_R})$, $\theta_1 = -(1/2\sqrt{2}|\varepsilon|^2) \times (\sqrt{2}(\varepsilon_R b_I - \varepsilon_I b_R) + \varepsilon_R \mathrm{sgn}(a_I)\sqrt{|a| - a_R} - \varepsilon_I\sqrt{|a| + a_R})$, $\sigma_0 = -(1/2\sqrt{2}|\varepsilon|^2)(\sqrt{2}(e_R b_R + e_I b_I) - \varepsilon_R\sqrt{|a| + a_R} - \varepsilon_I \mathrm{sgn}(a_I)\sqrt{|a| - a_R})$, $\sigma_1 = -(1/2\sqrt{2}|\varepsilon|^2)(\sqrt{2}(e_R b_R + e_I b_I) + \varepsilon_R\sqrt{|a| + a_R} + \varepsilon_I \mathrm{sgn}(a_I)\sqrt{|a| - a_R})$, $a_R = b_R^2 - b_I^2 - 4(\varepsilon_R\lambda_R - \varepsilon_I\lambda_I)$, $a_I = 2b_R b_I - 4(\varepsilon_R\lambda_I + \varepsilon_I\lambda_R)$, and $|a| = \sqrt{a_R^2 + a_I^2}$. The deployments in each dimension become decoupled if $|a| = 0$.

Clearly, selecting the appropriate PDE model (10)—specifically the coefficients ε, b, and λ required for a desired deployment family—is not as straightforward as in the 1-D case due to the complicated expressions for θ and σ given in (14)–(16). However, interesting deployments can be found with some effort. For example, by selecting $\varepsilon = 1/2 + j/2$, $b = \pi - j\pi$, and $\lambda = 0$, the deployment (14) becomes

$$\begin{bmatrix}\bar{x}(\alpha)\\ \bar{y}(\alpha)\end{bmatrix} = \begin{bmatrix}a_0 & -a_1\\ a_1 & a_0\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} + \begin{bmatrix}a_2 & -a_3\\ a_3 & a_2\end{bmatrix}\begin{bmatrix}-\sin(2\pi\alpha)\\ \cos(2\pi\alpha)\end{bmatrix} \tag{17}$$

which represents a circle deployment centered about the point (a_0, a_1). By simply changing the scalars a_0 and a_1, the deployment can be moved about the plane. In contrast, a circle deployment can also be formed using two independent, open-loop unstable 1-D reaction–diffusion equations with $\lambda = 4\pi^2$, whose equilibria correspond to a family of ellipses centered about the origin. The deployments (15) and (16) model families of spiral-like deployments.

C. Design Procedure for Desired Deployment Profiles

To achieve leader-enabled deployment onto possibly nonlinear (in α) planar curves using either two 1-D deployments or one complex-valued deployment, the user applies the following steps.

1) Select a desired *deployment family*, i.e., a family of basis functions.
2) Select *specific basis functions* by choosing the coefficients of the appropriate PDE model(s).
3) Choose *deployment coefficients* to generate a specific planar deployment.
4) Choose the desired deployment *convergence rate*.
5) Discretize the PDE model(s) spatially to obtain implementable control laws for the leader, anchor, and follower agents.

This procedure encompasses both feasibility (steps 1–3) and stability (steps 4 and 5). We have discussed deployment families for agents with feedback laws derived from PDE model(s), which guarantee feasibility, but not stability, of the deployments. For stable deployment, we design the control laws $U_0(t)$ and $U_1(t)$ for the anchor and the leader, and for the leader, observers to estimate the agents' positions. We now focus on these designs.

III. Leader Feedback Design

We employ PDE backstepping boundary control [34] for the complex-valued PDE model (10) and (11) since leader-based control naturally leads to formulations with actuation at the boundary. PDE backstepping succeeds in deriving closed-form controllers that achieve exponential stability with only boundary actuation. This approach is more elegant than other boundary methods that produce complicated controllers, which require solving operator Riccati equations. This design also applies, as a special case, to the real-valued PDE model (4) with boundary conditions (5) by treating $z(\alpha, t)$, ε, b, and λ as real-valued and setting $\varepsilon = 1$.

First, we introduce the deployment profile error

$$\begin{aligned}\zeta(\alpha, t) &= [x(\alpha, t) + jy(\alpha, t)] - [\bar{x}(\alpha) + j\bar{y}(\alpha, t)]\\ &= z(\alpha, t) - \bar{z}(\alpha, t)\end{aligned} \tag{18}$$

to shift the equilibrium of (10) to the origin. Substituting (18) into (10) and (11) yields

$$\zeta_t = \varepsilon\zeta_{\alpha\alpha} + b\zeta_\alpha + \lambda\zeta \tag{19}$$

$$\zeta_t(0) = U_0(t) \tag{20}$$

$$\zeta_t(1) = U_1(t) \tag{21}$$

and we remind the reader that the coefficients ε, b, and λ are complex and $\mathrm{Re}\{\varepsilon\} > 0$. Next, we substitute the change of variable [34]

$$v(\alpha) = \zeta(\alpha)e^{\frac{b}{2\varepsilon}\alpha} \tag{22}$$

into (19)–(21) to eliminate the advection term and obtain

$$v_t = \varepsilon v_{\alpha\alpha} + \left(\lambda - \frac{b^2}{4\varepsilon}\right)v \tag{23}$$

$$v_t(0) = U_0(t) \tag{24}$$

$$v_t(1) = e^{\frac{b}{2\varepsilon}}U_1(t). \tag{25}$$

Now, let $w(\alpha,t)$ be a new state that is defined by the coordinate transformation

$$w(\alpha,t)=v(\alpha,t)-\int_0^\alpha k(\alpha,\beta)v(\beta,t)d\beta \qquad (26)$$

where the kernel $k(\alpha,\beta)$ defined on $\mathcal{T}=\{(\alpha,\beta):0\leq\beta\leq\alpha\leq 1\}$ is given by [34]

$$k(\alpha,\beta)=-\gamma\beta\frac{I_1\left(\sqrt{\gamma(\alpha^2-\beta^2)}\right)}{\sqrt{\gamma(\alpha^2-\beta^2)}} \qquad (27)$$

with $\gamma=(1/\varepsilon)(c+\lambda-(b^2/4\varepsilon))$, $c>0$, In (27), I_1 denotes the first-order modified Bessel function of the first kind. The variable (26) is shown to satisfy the target PDE system

$$\begin{aligned} w_t &= \varepsilon w_{\alpha\alpha}-cw-\varepsilon k_\beta(\alpha,0)w(0) && (28)\\ w_t(0) &= -cw(0) && (29)\\ w_t(1) &= -cw(1) && (30)\end{aligned}$$

and this transformation can be inverted to obtain

$$v(\alpha,t)=w(\alpha,t)+\int_0^\alpha l(\alpha,\beta)w(\beta,t)d\beta \qquad (31)$$

where the inverse gain kernel $l\in\mathcal{C}^2(\mathcal{T})$ [34]. As will be seen in Section IV, the parameter c, which is selected by the user, determines the convergence rate of the deployment.

From (18), (22), (26), and the boundary conditions (24) and (29), we obtain the anchor's control law

$$U_0(t)=-c\zeta(0,t)=-cz(0,t)+c\bar{z}(0). \qquad (32)$$

For the leader's control law, we introduce the operator $\mathcal{K}\{\cdot\}$ acting on the function $\xi(\alpha,t)$ as

$$\begin{aligned}\mathcal{K}\{\xi\}(t) =& \varepsilon\sqrt{\gamma}I_1(\sqrt{\gamma})e^{-\frac{b}{2\varepsilon}}\xi(0,t)\\ &+\left(c+\frac{\varepsilon\gamma^2}{8}-\frac{\varepsilon\gamma}{2}+\frac{b\gamma}{4}\right)\xi(1,t)+\varepsilon\frac{\gamma}{2}\xi_\alpha(1,t)\\ &+\varepsilon\gamma^2\int_0^1 e^{-\frac{b}{2\varepsilon}(1-\beta)}\Pi(\beta)\xi(\beta,t)d\beta \qquad (33)\\ \Pi(\beta)=&\gamma\beta^3\frac{I_3\left(\sqrt{\gamma(1-\beta^2)}\right)}{(\gamma(1-\beta^2))^{\frac{3}{2}}}-3\beta\frac{I_2\left(\sqrt{\gamma(1-\beta^2)}\right)}{\gamma(1-\beta^2)}\\ &+\beta\frac{I_1\left(\sqrt{\gamma(1-\beta^2)}\right)}{\sqrt{\gamma(1-\beta^2)}} \qquad (34)\end{aligned}$$

where I_2 and I_3 indicate the second- and third-order modified Bessel functions of the first kind, respectively. From (18), (22), (26), and the boundary conditions (25) and (30), we arrive at the leader's control law

$$U_1(t)=-\mathcal{K}\{\zeta\}(t)=-\mathcal{K}\{z\}(t)+\mathcal{K}\{\bar{z}\} \qquad (35)$$

that, with (32), stabilizes the deployment profile $\bar{z}(\alpha)$.

Of note, the control laws (32) and (35) both contain a feedback term and a constant bias term, whose value is determined by the desired formation and can be computed prior to deployment. By simply changing the bias terms—without changing the feedback terms or the follower agents' control strategy—different deployment profiles can be induced and stabilized by the anchor and the leader. To achieve a specific formation, the user selects the deployment coefficients $a_0,\ldots,a_3$ to compute the bias $\mathcal{K}\{\bar{z}\}$. If the bias terms are zero, rendezvous at the origin is achieved.

However, if the user has no knowledge of the deployment family and instead changes the bias values directly—i.e., employs the boundary conditions $z_t(0)=-cz(0)+\bar{U}_0$ and $z_t(1)=-\mathcal{K}\{z\}(t)+\bar{U}_1$, where $\bar{U}_0,\bar{U}_1\in\mathbb{C}$ denote bias terms set by the user—the agents will stabilize the deployment profile $\bar{z}(\alpha)=e^{-(b/2\varepsilon)\alpha}(\bar{w}(\alpha)+\int_0^\alpha l(\alpha,\beta)\bar{w}(\beta)d\beta)$. This profile is found by applying the change of variable, $v(\alpha,t)=z(\alpha,t)e^{(b/2\varepsilon)\alpha}$, and the transformation (26) to the system (10) and (11) to obtain the target PDE system (28) with boundary conditions $w_t(0,t)=-cw(0,t)+\bar{U}_0$ and $w_t(1,t)=-cw(1,t)+e^{b/2\varepsilon}\bar{U}_1$. The equilibrium of (28) with these boundary conditions is

$$\begin{aligned}\bar{w}(\alpha)=&\left(\cosh(\Theta\alpha)-\frac{\cosh\Theta}{\sinh\Theta}\sinh(\Theta\alpha)\right)\frac{\bar{U}_0}{c}-\frac{\sqrt{\gamma}\bar{U}_0}{c\Theta}\\ &\times\left(\int_0^\alpha\frac{1}{\tau}\sinh\left(\Theta(\alpha-\tau)\right)I_1(\sqrt{\gamma}\tau)d\tau\right.\\ &\left.-\frac{\sinh(\Theta\alpha)}{\sinh\Theta}\int_0^1\frac{1}{\tau}\sinh\left(\Theta(1-\tau)\right)I_1(\sqrt{\gamma}\tau)d\tau\right)\\ &+\frac{e^{\frac{b}{2\varepsilon}}\bar{U}_1}{c\sinh\Theta}\sinh(\Theta\alpha) \qquad (36)\end{aligned}$$

where $\Theta=\sqrt{c(\varepsilon_R-j\varepsilon_I)}/|\varepsilon|$. We use the inverse transformation (31) and change of variable, $v(\alpha,t)=z(\alpha,t)e^{(b/2\varepsilon)\alpha}$, to obtain $\bar{z}(\alpha)$.

IV. Closed-Loop Stability

Due to the dynamic character of the boundary conditions (29) and (30), there are several aspects in which the stability analysis here differs from [34] and other work on PDE boundary control. A boundary value-dependent perturbation term arises on the right-hand side of (28), and the dynamic boundary conditions necessitate that the analysis be conducted in the Sobolev space, $H^1=\{f\in\mathcal{C}^1[0,1]|||f||^2=|f(0)|^2+|f(1)|^2+\int_0^1|f(\alpha)|^2d\alpha+\int_0^1|f_\alpha(\alpha)|^2d\alpha<\infty\}$, rather than the function space, $L^2=\{f\in\mathcal{C}[0,1]|||f||^2=\int_0^1|f(\alpha)|^2d\alpha<\infty\}$.

Theorem 1: The system (10) with boundary conditions (11) and control laws (32) and (35) is exponentially stable in the

H^1 norm at the equilibrium $z(\alpha,t) \equiv \bar{z}(\alpha)$, i.e., there exists $M > 0$ such that for all $t \geq 0$, $\Omega(t) \leq Me^{-ct}\Omega(0)$, where $c > 0$, $c \neq (b^2/4\varepsilon) - \lambda$, and

$$\Omega(t)=|\zeta(0,t)|^2+|\zeta(1,t)|^2+\int_0^1 |\zeta(\alpha,t)|^2\, d\alpha+\int_0^1 |\zeta_\alpha(\alpha,t)|^2 d\alpha. \tag{37}$$

Proof: We begin by proving exponential stability of the target system (28)–(30). Let $V(t)$ be the Lyapunov functional

$$V(t) = \frac{m}{2}|w(0,t)|^2 + \frac{1}{2}|w(1,t)|^2 + \frac{1}{2}\int_0^1 |w_\alpha(\alpha,t)|^2\, d\alpha \tag{38}$$

where m is a positive scalar to be determined. In the sequel, we omit the arguments (α,t) unless needed for clarity. Taking the time derivative of $V(t)$ gives

$$\dot{V} = \frac{1}{2}\left[m w_t(0)w(0)^* + m w(0)w_t(0)^* + w_t(1)w(1)^* \right. \\ \left. + w(1)w_t(1)^* + \int_0^1 (w_{\alpha t}w_\alpha^* + w_\alpha w_{\alpha t}^*)\, d\alpha \right] \tag{39}$$

where w^* denotes the complex conjugate of w. Substituting (30) yields

$$\dot{V} = -cm|w(0)|^2 - c|w(1)|^2 + \mathrm{Re}\left\{ \int_0^1 w_{\alpha t}w_\alpha^* d\alpha \right\}. \tag{40}$$

Integrating by parts and substituting (28)–(30) gives

$$\begin{aligned}\dot{V} =& -cm|w(0)|^2 - c|w(1)|^2 \\ & - \mathrm{Re}\left\{ cww_\alpha^*|_0^1 \right. \\ & \left. + \int_0^1 (\varepsilon w_{\alpha\alpha} - cw - \varepsilon k_\beta(\alpha,0)w(0)) \times w_{\alpha\alpha}^* d\alpha \right\} \\ =& -cm|w(0)|^2 - c|w(1)|^2 - c\int_0^1 |w_\alpha|^2 d\alpha - \varepsilon_R \int_0^1 |w_{\alpha\alpha}|^2 d\alpha \\ & + \mathrm{Re}\left\{ w(0)\int_0^1 \varepsilon k_\beta(\alpha,0)w_{\alpha\alpha}^* d\alpha \right\} \\ \leq& -cm|w(0)|^2 - c|w(1)|^2 - c\int_0^1 |w_\alpha|^2 d\alpha - \varepsilon_R \int_0^1 |w_{\alpha\alpha}|^2 d\alpha \\ & + |w(0)|\left| \int_0^1 \varepsilon k_\beta(\alpha,0)w_{\alpha\alpha}^* d\alpha \right|. \end{aligned} \tag{41}$$

We now apply the Cauchy-Schwarz and Young's inequalities with the parameter $\rho > 0$ to obtain

$$\begin{aligned}\dot{V} \leq& -cm|w(0)|^2 - c|w(1)|^2 - c\int_0^1 |w_\alpha|^2 d\alpha \\ & - \varepsilon_R \int_0^1 |w_{\alpha\alpha}|^2 d\alpha + \frac{\rho}{2}|w(0)|^2 \int_0^1 |\varepsilon k_\beta(\alpha,0)|^2 d\alpha \\ & + \frac{1}{2\rho}\int_0^1 |w_{\alpha\alpha}|^2 d\alpha \\ =& -\left(c - \frac{\rho\Lambda}{2m}\right) m|w(0)|^2 - c|w(1)|^2 - c\int_0^1 |w_\alpha|^2 d\alpha \\ & - \left(\varepsilon_R - \frac{1}{2\rho}\right)\int_0^1 |w_{\alpha\alpha}|^2 d\alpha \end{aligned} \tag{42}$$

where $\Lambda = \int_0^1 |\varepsilon k_\beta(\alpha,0)|^2 d\alpha$. Selecting the parameters $m = \rho\Lambda/c$ and $\rho > (1/2\varepsilon_R)$, we find

$$\dot{V} \leq -\frac{c}{2}m|w(0)|^2 - c|w(1)|^2 - c\int_0^1 |w_\alpha|^2 d\alpha \leq -cV. \tag{43}$$

(The choice of m assumes that $c \neq (b^2/4\varepsilon) - \lambda$ so that $\gamma \neq 0$ and $\Lambda > 0$.)

From the Comparison Lemma [36] and Lemma 4 in the Appendix, we have

$$\begin{aligned}\Omega(t) &\leq \frac{1}{p_1}\Psi(t) \leq \frac{1}{p_1q_1}V(t) \leq \frac{1}{p_1q_1}e^{-ct}V(0) \\ &\leq \frac{q_2}{p_1q_1}e^{-ct}\Psi(0) \leq \frac{p_2q_2}{p_1q_1}e^{-ct}\Omega(0)\end{aligned} \tag{44}$$

where $q_1 = (1/16)\min(1,(8\rho\Lambda/c))$, $q_2 = (1/2)\max(1,(\rho\Lambda/c))$, and p_1, p_2 are shown in (103) and (104). The stability result is obtained from (44) with $M = p_2q_2/p_1q_1$. ■

From Theorem 1, we see that the leader-enabled continuum design achieves exponential stability with a decay rate that can be arbitrarily set by the user, namely, the gain c.

V. Leader Observer Design

In the previous sections, we have assumed that the leader agent has knowledge of all the agents' positions. Since the leader agent is acting as a boundary actuator, it is also natural to use it as a boundary sensor, providing measurements to an observer that estimates the positions of all the agents. We now consider two scenarios: 1) the leader knows the position of itself, its nearest neighbor, and the anchor agent, and 2) the leader knows the position of only itself and its nearest neighbor. In both cases, the leader agent also knows the anchor's bias term. For these scenarios, we use backstepping for PDEs with boundary sensing [35] to design exponentially stable observers of the follower agents' positions for use in the leader agent's controller (35). As with boundary control, PDE backstepping leads to observer designs with closed-form observer gains, whereas other methods lead to more complicated designs.

A. Leader, Neighbor, and Anchor Measurements

Consider the observer for the PDE system (10) and (11)

$$\begin{aligned} \hat{z}_t =& \varepsilon \hat{z}_{\alpha\alpha} + b\hat{z}_\alpha + \lambda \hat{z} + p_1(\alpha)\left[z_\alpha(1) - \hat{z}_\alpha(1)\right] \\ &+ p_2(\alpha)\left[z(1) - \hat{z}(1)\right] \qquad (45) \\ \hat{z}_t(0) =& -c\hat{z}(0) + c\bar{z}(0) + p_{00}\left[z(0) - \hat{z}(0)\right] \qquad (46) \\ \hat{z}_t(1) =& U_1(t) + p_{10}\left[z(1) - \hat{z}(1)\right] \qquad (47) \end{aligned}$$

where $\hat{z}(\alpha, t)$ is the position estimate of agent α at time t, $U_1(t)$ is the leader agent's control input, and $z(0)$, $z(1)$, and $z_\alpha(1)$ are measurements. The functions, $p_1(\alpha)$ and $p_2(\alpha)$, and the constants, p_{00} and p_{10}, are observer gains to be determined.

Define the observer error variable $\tilde{z} = z - \hat{z}$ and consider the PDE error system

$$\begin{aligned} \tilde{z}_t =& \varepsilon \tilde{z}_{\alpha\alpha} + b\tilde{z}_\alpha + \lambda \tilde{z} - p_1(\alpha)\tilde{z}_\alpha(1) - p_2(\alpha)\tilde{z}(1) \qquad (48) \\ \tilde{z}_t(0) =& -(c + p_{00})\tilde{z}(0) \qquad (49) \\ \tilde{z}_t(1) =& -p_{10}\tilde{z}(1). \qquad (50) \end{aligned}$$

We eliminate the advection term by substituting the change of variable $\tilde{v}(\alpha) = \tilde{z}(\alpha)e^{(b/2\varepsilon)\alpha}$ into (48)–(50) to obtain the reaction–diffusion equation

$$\begin{aligned} \tilde{v}_t =& \varepsilon \tilde{v}_{\alpha\alpha} + \left(\lambda - \frac{b^2}{4\varepsilon}\right)\tilde{v} - e^{-\frac{b}{2\varepsilon}(1-\alpha)} p_1(\alpha)\tilde{v}_\alpha(1) \\ &+ e^{-\frac{b}{2\varepsilon}(1-\alpha)}\left(p_2(\alpha) - \frac{b}{2\varepsilon}p_1(\alpha)\right)\tilde{v}(1) \qquad (51) \\ \tilde{v}_t(0) =& -(c + p_{00})\tilde{v}(0) \qquad (52) \\ \tilde{v}_t(1) =& -p_{10}\tilde{v}(1). \qquad (53) \end{aligned}$$

Now, define the coordinate transformation [35]

$$\tilde{v}(\alpha) = \tilde{w}(\alpha) - \int_\alpha^1 p(\alpha, \beta)\tilde{w}(\beta)d\beta \qquad (54)$$

where the kernel $p(\alpha,\beta)$ defined on $\mathcal{S} = \{(\alpha, \beta) : 0 \leq \alpha \leq \beta \leq 1\}$ is given by $p(\alpha, \beta) = -\delta\alpha(I_1(\sqrt{\delta(\beta^2 - \alpha^2)})/\sqrt{\delta(\beta^2 - \alpha^2)})$ with $\delta = (1/\varepsilon)(d + \lambda - (b^2/4\varepsilon))$, $d > 0$. Using (54), we transform the error system (51)–(53) to the target system

$$\begin{aligned} \tilde{w}_t =& \varepsilon \tilde{w}_{\alpha\alpha} - d\tilde{w} \qquad (55) \\ \tilde{w}_t(0) =& -d\tilde{w}(0) \qquad (56) \\ \tilde{w}_t(1) =& -d\tilde{w}(1) \qquad (57) \end{aligned}$$

to determine the observer gains. From (51), (54), and (55), we find

$$p_1(\alpha) = -\varepsilon\delta\alpha \frac{I_1\left(\sqrt{\delta(1-\alpha^2)}\right)}{\sqrt{\delta(1-\alpha^2)}} e^{\frac{b}{2\varepsilon}(1-\alpha)} \qquad (58)$$

$$\begin{aligned} p_2(\alpha) =& \varepsilon\delta^2\alpha \frac{I_2\left(\sqrt{\delta(1-\alpha^2)}\right)}{\delta(1-\alpha^2)} e^{\frac{b}{2\varepsilon}(1-\alpha)} \\ &- \frac{1}{2}(b + \varepsilon\delta)\delta\alpha \frac{I_1\left(\sqrt{\delta(1-\alpha^2)}\right)}{\sqrt{\delta(1-\alpha^2)}} e^{\frac{b}{2\varepsilon}(1-\alpha)} \qquad (59) \end{aligned}$$

and from (52)–(54), (56), and (57), we have $p_{00} = d - c$ and $p_{10} = d$.

Theorem 2: The observer (45)–(47) with gains (58) and (59), $p_{00} = d - c$, and $p_{10} = d$ converges exponentially in the H^1 norm to the state $z(\alpha, t)$, i.e., there exists $N > 0$ such that for all $t \geq 0$, $\Sigma(t) \leq Ne^{-2dt}\Sigma(0)$, where c, $d > 0$ and

$$\Sigma(t) = |\tilde{z}(0,t)|^2 + |\tilde{z}(1,t)|^2 + \int_0^1 |\tilde{z}(\alpha,t)|^2 d\alpha + \int_0^1 |\tilde{z}_\alpha(\alpha,t)|^2 d\alpha. \qquad (60)$$

Proof: The transformation (54) can be inverted to obtain $\tilde{w}(\alpha, t) = \tilde{v}(\alpha, t) + \int_\alpha^1 q(\alpha, \beta)\tilde{v}(\beta, t)d\beta$, where the inverse gain kernel $q \in \mathcal{C}^2(\mathcal{S})$ [34], so we begin by proving exponential stability of the target system (55)–(57). Let $W(t)$ be the Lyapunov functional

$$W(t) = \frac{1}{2}|\tilde{w}(0,t)|^2 + \frac{1}{2}|\tilde{w}(1,t)|^2 + \frac{1}{2}\int_0^1 |\tilde{w}_\alpha(\alpha,t)|^2 d\alpha. \qquad (61)$$

In the sequel, we omit the arguments (α, t) unless needed for clarity.

Computing the time derivative of $W(t)$, integrating by parts, and substituting (55)–(57) yields

$$\begin{aligned} \dot{W} =& -d|\tilde{w}(0)|^2 - d|\tilde{w}(1)|^2 \\ &- \mathrm{Re}\left\{ d\, ww_\alpha^*\big|_0^1 + \int_0^1 (\varepsilon\tilde{w}_{\alpha\alpha} - d\tilde{w})\tilde{w}_{\alpha\alpha}^* d\alpha \right\}. \qquad (62) \end{aligned}$$

Integrating by parts gives

$$\begin{aligned} \dot{W} =& -d\left(|\tilde{w}(0)|^2 + |\tilde{w}(1)|^2 + \int_0^1 |\tilde{w}_\alpha|^2 d\alpha\right) - \varepsilon_R \int_0^1 |\tilde{w}_{\alpha\alpha}|^2 d\alpha \\ \leq& -2dW. \qquad (63) \end{aligned}$$

From the Comparison Lemma [36] and Lemma 5 in the Appendix, we have

$$\begin{aligned} \Sigma(t) \leq& \frac{1}{r_1}\Phi(t) \leq \frac{16}{r_1}W(t) \leq \frac{16}{r_1}e^{-2dt}W(0) \\ \leq& \frac{8}{r_1}e^{-2dt}\Phi(0) \leq \frac{8r_2}{r_1}e^{-2dt}\Sigma(0) \qquad (64) \end{aligned}$$

where r_1 and r_2 are shown in (105) and (106). The stability result is obtained from (64) with $N = 8r_2/r_1$. ■

B. Leader and Neighbor Measurements

From $p_{00} = d - c$, we see that choosing the target system (55)–(57) with $d = c$ results in the observer gain $p_{00} = 0$, nullifying the anchor agent position measurement. Thus, the transformation (54) transforms the error system (51)–(53) to the target system, $\tilde{w}_t = \varepsilon\tilde{w}_{\alpha\alpha} - c\tilde{w}$, with boundary conditions $\tilde{w}_t(0) = -c\tilde{w}(0)$, $\tilde{w}_t(1) = -c\tilde{w}(1)$, and yields an observer that requires communication/sensing of only the leader agent's nearest neighbor

$$\begin{aligned} \hat{z}_t =& \hat{z}_{\alpha\alpha} + b\hat{z}_\alpha + \lambda\hat{z} + p_1(\alpha)\left[z_\alpha(1) - \hat{z}_\alpha(1)\right] \\ &+ p_2(\alpha)\left[z(1) - \hat{z}(1)\right] \qquad (65) \end{aligned}$$

$$\hat{z}_t(0) = -c\hat{z}(0) + c\bar{z}(0) \tag{66}$$

$$\hat{z}_t(1) = U_1(t) + c\left[z(1) - \hat{z}(1)\right] \tag{67}$$

where the observer gains $p_1(\alpha)$ and $p_2(\alpha)$ are given by (58) and (59) by setting $\delta = \gamma$. From Theorem 2, we have exponential stability in the H^1 norm; specifically, $\Sigma(t) \leq Ne^{-2ct}\Sigma(0)$. Note how the decay rate is tied to the control gain c and, hence, cannot be arbitrarily set without knowledge of the anchor agent's position.

C. Output Feedback

Equipped with the stabilizing feedback (35) and observer (45)–(47), one can now pursue output feedback for the leader agent to stabilize the deployment profile, namely, let $U_1(t) = -\mathcal{K}\{\hat{z}\}(t) + \mathcal{K}\{\bar{z}\}$ be the control input for the leader agent. To prove stability with output feedback, we must prove stability of the observer and observer-error system $(\hat{z}, \tilde{z})$, which can be converted to the target system $(\hat{w}, \tilde{w})$ using the change of variables $\hat{v}(\alpha) = (\hat{z}(\alpha) - \bar{z}(\alpha))e^{(b/2\varepsilon)\alpha}$ and $\tilde{v}(\alpha) = \tilde{z}(\alpha)e^{(b/2\varepsilon)\alpha}$, and the transformations $\hat{w}(\alpha) = \hat{v}(\alpha) - \int_0^\alpha k(\alpha,\beta)\hat{v}(\beta)d\alpha$ and (54). The target system $(\hat{w}, \tilde{w})$ is

$$\begin{aligned}\hat{w}_t =& \varepsilon\hat{w}_{\alpha\alpha} - c\hat{w} - \varepsilon k_\beta(\alpha,0)\hat{w}(0)\\ &+ \varepsilon\left(r(\alpha) - p_\beta(\alpha,1)\right)\tilde{w}(1)\\ &+ \varepsilon\left(s(\alpha) + p(\alpha,1)\right)\tilde{w}_\alpha(1)\end{aligned} \tag{68}$$

$$\hat{w}_t(0) = -c\hat{w}(0) + (d-c)\tilde{w}(0) \tag{69}$$

$$\hat{w}_t(1) = -c\hat{w}(1) + \left(d + \varepsilon r(1)\right)\tilde{w}(1) + \varepsilon s(1)\tilde{w}_\alpha(1) \tag{70}$$

where the $\tilde{w}$-system is given by (55)–(57), $r(\alpha) = \int_0^\alpha k(\alpha,\beta)p_\xi(\beta,1)d\beta$, and $s(\alpha) = -\int_0^\alpha k(\alpha,\beta)p(\beta,1)d\beta$. This system is a cascade connection where the exponentially stable $\tilde{w}$-system drives the $\hat{w}$-system, which is also exponentially stable when unforced. Consequently, one would expect stability of this system to be in the H^1 or higher norm.

VI. Discretized Agent Control Laws

Implementable control laws for a finite number of agents are obtained by spatially discretizing the PDE model (10), the leader agent's controller (35), and if employed, the observer (45)–(47). The anchor agent's controller (32) does not require modification since it utilizes only the anchor's own position. When we spatially discretize the system, the state variable $z(\alpha, t)$ becomes $z(ih, t)$, where $i \in \{0, \ldots, n\}$, $h = 1/n$, and $(n+1)$ is the total number of agents. We denote the position of the anchor, follower, and leader agents as z_0, z_i, and z_n.

The discretization process gives rise to two questions: 1) How many agents must be deployed, i.e., how fine must the spatial discretization be to retain stability in the closed-loop system? 2) Is the equilibrium of the spatially discretized system the same as the equilibrium of the continuous system? One can address the first question by transforming the spatially discretized system using the transformation, $w(ih,t) = u(ih,t) - h\sum_{j=0}^{i-1} k(ih,jh)u(jh,t)$, where k is the gain kernel (27), and performing a Lyapunov analysis on the resulting target system. Due to space limitations, we do not address this issue in this paper, but similar results can be found in [37]–[39]. Moreover, when compared to other design options, the benefits of this multiagent control design, which allows the user to deploy large numbers of agents into various planar formations by communicating with only two agents, are mitigated when deploying few agents. Concerning the second question, we present a method that minimizes the approximation error due to the spatial discretization at the desired deployment $\bar{z}(ih)$. Hence, the achieved deployment $\bar{z}_i$ closely approximates $\bar{z}(ih)$. Note, however, that both questions 1) and 2) become irrelevant as the number of agents deployed is increased.

Using three-point central differencing to approximate the spatial derivatives in (10), we obtain the discretized follower agent control laws

$$U_i(t) = \varepsilon\frac{z_{i+1} - 2z_i + z_{i-1}}{h^2} + b\frac{z_{i+1} - z_{i-1}}{2h} + \lambda z_i. \tag{71}$$

For the leader agent, we define the operator $\mathcal{K}_D\{\cdot\}$, a discretized version of (33), acting on the vector $[\xi_0, \cdots, \xi_n]^T$, where $\xi_i(t) = \xi(ih, t)$, as

$$\begin{aligned}\mathcal{K}_D\{\xi\}(t) =& \varepsilon\sqrt{\gamma}I_1(\sqrt{\gamma})e^{-\frac{b}{2\varepsilon}}\xi_0(t)\\ &+ \left(c + \frac{\varepsilon\gamma^2}{8} - \frac{\varepsilon\gamma}{2} + \frac{b\gamma}{4}\right)\xi_n(t)\\ &+ \varepsilon\frac{\gamma}{2}\left(\frac{\xi_n(t) - \xi_{n-1}(t)}{h}\right)\\ &+ \varepsilon\gamma^2\frac{h}{2}\left(f_n(t) + 2\sum_{i=1}^{n-1} f_i(t)\right)\end{aligned} \tag{72}$$

where $f_i(t) = e^{-(b/2\varepsilon)(1-ih)}\Pi(ih)\xi_i(t)$, $f_n(t) = ((\gamma/48) + (1/8))\xi_n(t)$, and Π is shown in (34). To obtain (72), we use the trapezoidal rule and the two-point backward difference to approximate the integral and $z_\alpha(1)$ terms in (33). The leader's discretized control law is simply

$$U_n(t) = -\mathcal{K}_D\{z\}(t) + \mathcal{K}_D\{\bar{z}\}. \tag{73}$$

Together, the control laws (32), (71), and (73) govern the dynamics of the agents according to the linear ODE system

$$\dot{z}_0 = U_0(t) \quad \dot{z}_i = U_i(t) \quad \dot{z}_n = U_n(t). \tag{74}$$

We see directly from (71) that the spatial discretization imposes a fixed communication topology on the follower agents, specifically a chain graph, since the followers depend only on their nearest neighbors on the graph. The leader agent requires global information to stabilize the deployment as seen by (73). However, the observers designed in Section V can be discretized in a similar manner so that if output feedback were employed, the leader agent's control law is

$$U_n(t) = -\mathcal{K}_D\{\hat{z}\}(t) + \mathcal{K}_D\{\bar{z}\} \tag{75}$$

which requires only local information.

Due to the approximation error inherent in the spatial discretization, the equilibrium of (74), $\bar{z}_i$, will not equal the desired deployment, $\bar{z}(ih)$. Evaluating (74) at $z_i = \bar{z}(ih)$, we have $U_0(t) = -c\bar{z}(0) + c\bar{z}(0) = 0$, $U_n(t) = -\mathcal{K}_D\{\bar{z}\}(t) + \mathcal{K}_D\{\bar{z}\} = 0$, and

$$\begin{aligned} U_i(t) &= \varepsilon\frac{\bar{z}_{i+1} - 2\bar{z}_i + \bar{z}_{i-1}}{h^2} + b\frac{\bar{z}_{i+1} - \bar{z}_{i-1}}{2h} + \lambda\bar{z}_i \\ &= \underbrace{\varepsilon\bar{z}_{\alpha\alpha}(ih) + b\bar{z}_\alpha(ih) + \lambda\bar{z}(ih)}_{=0} + O(h^2) \end{aligned} \tag{76}$$

where $O(h^2)$ is the accuracy of the three-point central difference [40]. Because the desired deployment is encoded in the leader and anchor agents' bias terms, $U_0(t) = U_n(t) = 0$ when $z_i = \bar{z}(ih)$. The followers, however, are unaware of $\bar{z}(ih)$, and at $z_i = \bar{z}(ih)$, $U_i(t) = O(h^2) \neq 0$.

To make $\bar{z}_i$ closely approximate $\bar{z}(ih)$, we need to minimize the approximation error $O(h^2)$, but we must do so for the entire deployment family, not for specific deployments. Otherwise, the error minimization will adversely affect the flexibility of the design and will require the follower agents to know the desired deployment. One method is to simply deploy more agents since $O(h^2) \to 0$ as $n \to \infty$, but we seek a method to decrease the error for a fixed n. We exploit our knowledge of the basis functions (13) that comprise the desired deployment family, allowing us to characterize a measure of the approximation error for the entire family.

Define vectors for the family of basis function spatial derivatives as $\mathbf{\Psi}_{\alpha\alpha} = [\psi_{0,\alpha\alpha}(h), \ldots, \psi_{0,\alpha\alpha}(nh - h), \ldots, \psi_{3,\alpha\alpha}(h), \ldots, \psi_{3,\alpha\alpha}(nh - h)]^T$ and $\mathbf{\Psi}_\alpha = [\psi_{0,\alpha}(h), \ldots, \psi_{0,\alpha}(nh-h), \ldots, \psi_{3,\alpha}(h), \ldots, \psi_{3,\alpha}(nh-h)]^T$, where $\psi_{k,\alpha\alpha}(ih)$ and $\psi_{k,\alpha}(ih)$ are the second and first spatial derivatives of the basis function ψ_k for agent i, $i \in \{1, \ldots, n-1\}$. The vectors for the approximations of the spatial derivatives, $\hat{\mathbf{\Psi}}_{\alpha\alpha}$ and $\hat{\mathbf{\Psi}}_\alpha$, are defined similarly, but with the three-point central difference approximations $\hat{\psi}_{k,\alpha\alpha}(ih) = (1/h^2)(\psi_k(ih + h) - 2\psi_k(h) + \psi_k(ih - h))$ and $\hat{\psi}_{k,\alpha}(ih) = (1/2h)(\psi_k(ih + h) - \psi_k(ih - h))$. The actual derivatives and approximations are related by $\psi_{k,\alpha\alpha}(ih) = \hat{\psi}_{k,\alpha\alpha}(ih) + O(h^2)$ and $\psi_{k,\alpha}(ih) = \hat{\psi}_{k,\alpha}(ih) + O(h^2)$.

Now, define the second and first spatial derivative approximation error for the family of basis functions as $\tilde{\mathbf{\Psi}}_{\alpha\alpha} = \mathbf{\Psi}_{\alpha\alpha} - \hat{\mathbf{\Psi}}_{\alpha\alpha}$ and $\tilde{\mathbf{\Psi}}_\alpha = \mathbf{\Psi}_\alpha - \hat{\mathbf{\Psi}}_\alpha$, and characterize the error as

$$\|\tilde{\mathbf{\Psi}}_{\alpha\alpha}\|_2 = \mu_0 \leq \bar{\mu}_0 \quad \|\tilde{\mathbf{\Psi}}_\alpha\|_2 = \mu_1 \leq \bar{\mu}_1 \tag{77}$$

where $\bar{\mu}_0$ and $\bar{\mu}_1$ represent user-determined thresholds for the approximation error, and $\|\cdot\|_2$ denotes the Euclidean norm. If both inequalities are satisfied, the achieved deployment $\bar{z}_i$ will match the desired deployment $\bar{z}(ih)$ according to the user's specifications.

However, what if one or both of the above inequalities are not satisfied? We propose the modified follower agent control law

$$U_i(t) = \breve{\varepsilon}\frac{z_{i+1} - 2z_i + z_{i-1}}{h^2} + \breve{b}\frac{z_{i+1} - z_{i-1}}{2h} + \lambda z_i \tag{78}$$

where $\breve{\varepsilon} = \varepsilon(1 + \nu_0)$, $\breve{b} = b(1 + \nu_1)$, $\nu_0 = \arg\min_\nu f_0(\nu)$, $f_0(\nu) = \|\hat{\mathbf{\Psi}}_{\alpha\alpha}\nu - \tilde{\mathbf{\Psi}}_{\alpha\alpha}\|_2^2$, $\nu_1 = \arg\min_\nu f_1(\nu)$, and $f_1(\nu) = \|\hat{\mathbf{\Psi}}_\alpha\nu - \tilde{\mathbf{\Psi}}_\alpha\|_2^2$. Hence, (78) is minimal in a least squares sense at $z_i = \bar{z}(ih)$ because the numerical approximation error for the family of basis function spatial derivatives has been minimized by using the approximations $(1 + \nu_0)\hat{\psi}_{k,\alpha\alpha}(ih)$, $(1 + \nu_1)\hat{\psi}_{k,\alpha}(ih)$ for the actual derivatives $\psi_{k,\alpha\alpha}(ih)$, $\psi_{k,\alpha}(ih)$. Since $f_0(\nu)$ and $f_1(\nu)$ are minimized over a single parameter, the follower agents can still achieve a family of deployments while employing a single set of gains (78). For 1-D deployments, the procedure to find ν_0 and ν_1 is the same, but the minimization is done using two basis functions instead of four.

If $\mu_0 \leq \bar{\mu}_0$ and/or $\mu_1 \leq \bar{\mu}_1$ is satisfied, ν_0 and/or ν_1 can be simply set to zero. For example, the three-point central difference is exact for linear deployments, so $\nu_0 = \nu_1 = 0$. If after computing ν_0 and/or ν_1, the inequalities $f_0(\nu_0)\bar{\mu}_0$ and/or $f_1(\nu_1) \leq \bar{\mu}_1$ are not satisfied, improved deployment accuracy can be achieved only by deploying more agents. As an example, consider a 1-D deployment modeled by $x_t = x_{\alpha\alpha} + 4\pi^2 x$, which has basis functions $(\cos(2\pi\alpha), \sin(2\pi\alpha))$. For deployments with 10 agents ($n = 9$), the approximation error is $\|\tilde{\mathbf{\Psi}}_{\alpha\alpha}\|_2 = 4.4622$ when the followers employ (71) and is reduced to $f_0(\nu_0) = 1.2841 \times 10^{-13}$, where $\nu_0 = 0.0416$, when implementing (78) instead, indicating that the achieved deployment $\bar{z}_i$ is a much closer approximation of $\bar{z}(ih)$ when using the feedback laws (78).

While increasing the number of agents causes $\bar{z}_i$ to better approximate $\bar{z}(ih)$ because the discretization step-size h decreases, this smaller h leads to large gains in the follower agents feedback laws (78). However, we can simply scale all the control laws to prevent disproportionately large gains and, hence, velocity commands. Specifically, we implement the system $\dot{z}_0 = \kappa U_0(t)$, $\dot{z}_i = \kappa U_i(t)$, $\dot{z}_n = \kappa U_n(t)$, which has a decay rate of κc instead of c. This scaling is meant for the implementation of the control strategies on velocity-actuated systems. Alternatively, this deployment paradigm could be used as a planning algorithm where $z(\alpha, t)$ represents a nonphysical consensus-like variable such that no physical actuation is required, making this scaling unnecessary.

VII. Simulations

We present a variety of deployment examples to demonstrate the flexibility of our results. All deployments, unless otherwise stated, are simulated with 10 agents ($n = 9$), where the anchor and follower agents employ (32) and (78), and the leader agent employs output feedback (75) using the observer (45)–(47) with $d = 10$. The agents use the scaling $\kappa = h/4$, and the leader and anchor agents select the control gain $c = 7.5$. we provide the parameter set $(\varepsilon, b, \lambda, \nu_0, \nu_1)$ for each PDE model used. In Figs. 4–6, the trajectories are shaded from light to dark as the agents move from one desired planar deployment to another. The desired deployments are also projected onto the xy plane when needed for clarity.

Our first example focuses on the family of elliptical deployments that are centered about the origin, which is derived from identical, unstable reaction–diffusion equations $(1, 0, 4\pi^2, 0.0416, 0)$. In Fig. 3(a), the agents' initial positions are sampled from the Gaussian distribution $\mathcal{N}(0, 1)$ (zero mean with unity variance) and rendezvous at the origin due to the movements of the anchor and the leader

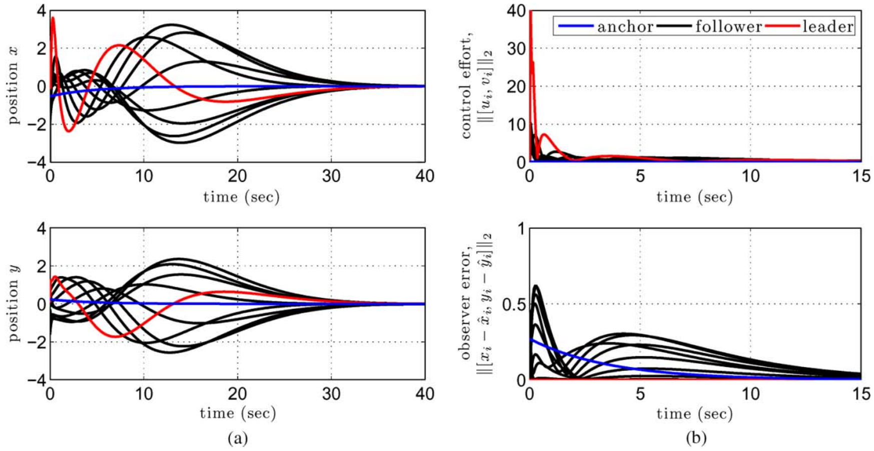

Fig. 3. Agents in the ellipse deployment family (a) rendezvous at the origin with (b) the corresponding control effort (top) and observer error (bottom). The agents converge to the origin after the observer error is negligible (≈14 s). In (b), we plot the control effort and observer error for $t \in [0, 15]$ s to highlight the transients since all the curves decay to zero after 15 s.

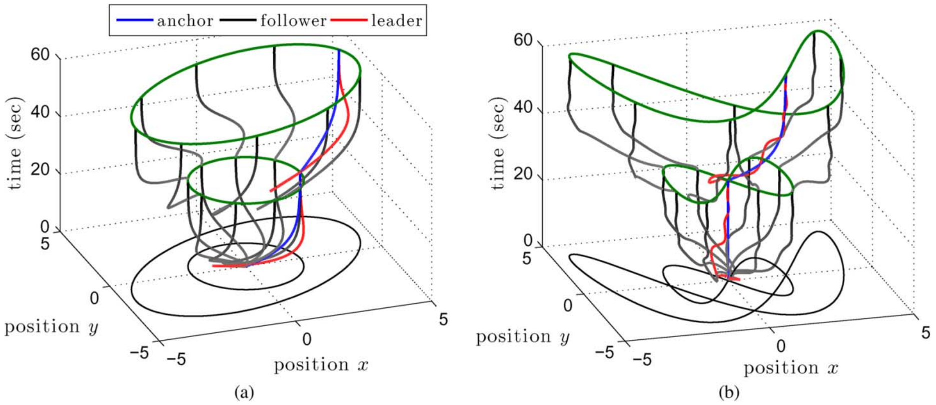

Fig. 4. Agent trajectories for (a) the ellipse deployment family when deploying to a circle and then to a rotated ellipse, and (b) the Lissajous figure-8 deployment family when deploying to a figure-8 and a distorted figure-8.

agents, whose bias terms are set to zero. The observer's initial condition is $(\hat{x}_i(0), \hat{y}_i(0)) = (x_i(0) + \xi_i, y_i(0) + \eta_i)$, $i \in \{0, \ldots, n-1\}$, where (ξ_i, η_i) is sampled from the distribution $\mathcal{N}(0, 0.04)$, and $(\hat{x}_n(0), \hat{y}_n(0)) = (x_n(0), y_n(0))$. Fig. 3(b) shows the control effort $\|[u_i, v_i]\|_2$ and the norm of the observer error $\|[x_i - \hat{x}_i, y_i - \hat{y}_i]\|_2$ for each agent. Note how the agents exhibit a transient before converging toward the origin at approximately 14 s, which is when the observer error becomes negligible. In Fig. 4(a), the anchor and leader change their bias terms to deploy the agents onto a circle $[\bar{x}(\alpha) = 2\cos(2\pi\alpha),\ \bar{y}(\alpha) = 2\sin(2\pi\alpha)]$ and then onto an ellipse rotated counterclockwise about the origin by $\phi = \pi/6$ rad $[\bar{x}(\alpha) = (5\sqrt{3}/2)\cos(2\pi\alpha) - (3/2)\sin(2\pi\alpha)$, $\bar{y}(\alpha) = (5/2)\cos(2\pi\alpha) + (3\sqrt{3}/2)\sin(2\pi\alpha)]$.

For the following examples, we could again initialize the agents randomly in the plane with an initial observer error, but instead we assume convergence of the observer and initialize the agents at the origin to depict the agents deploying, and then redeploying, more clearly. Fig. 4(b) depicts the agents deploying onto a figure-8 $[\bar{x}(\alpha) = 2\sin(2\pi\alpha), \bar{y}(\alpha) = 2\sin(4\pi\alpha)]$ and a distorted figure-8 $[\bar{x}(\alpha) = 3\cos(2\pi\alpha) + 6\sin(2\pi\alpha),\ \bar{y}(\alpha) = 6\sin(4\pi\alpha)]$, which are in, what we term, the Lissajous figure-8 family. This family is modeled using two unstable reaction–diffusion equations, where the parameters $(1, 0, 4\pi^2, 0.0416, 0)$ model the x-axis deployment and $(1, 0, 16\pi^2, 0.1796, 0)$ the y-axis deployment. The deployments depicted in Fig. 5(a) are formed using the advection–diffusion equation $(1, 5, 0, -0.0253, -0.0496)$ for the x-axis deployment and an unstable reaction–diffusion equation $(1, 0, 2\pi^2, 0.0206, 0)$ for the y-axis deployment. At $t = 30$ s, the agents stabilize $\bar{x}(\alpha) = (2/e^{-5} - 1)(-e^{-5} - 1 + 2e^{-5\alpha}), \bar{y}(\alpha) = 3\sin(\sqrt{2}\pi\alpha)$ before redeploying to $\bar{x}(\alpha) = (5/e^{-5} - 1)(-e^{-5} - 1 + 2e^{-5\alpha})$, $\bar{y}(\alpha) = 3\cos(\sqrt{2}\pi\alpha) + 4\sin(\sqrt{2}\pi\alpha)$.

We now provide some examples that utilize Ginzburg–Landau PDE models. In Fig. 5(b), the agents deploy to a circle centered at (0, 0) $[\bar{x}(\alpha) = -(5/2)\sin(2\pi\alpha)$, $\bar{y}(\alpha) = (5/2)\cos(2\pi\alpha)]$ and redeploy to a larger circle centered at (4, −4) $[\bar{x}(\alpha) = 4 - 5\sin(2\pi\alpha), \bar{y}(\alpha) = -4 + 5\cos(2\pi\alpha)]$ using the complex-valued reaction–advection equation $(1/2 + j/2, \pi - j\pi, 0, 0.0416, 0.0861)$, which generates circles centered about (a_0, a_1). The ability to move the circles about the plane makes this deployment applicable to capture and escort missions [17], [41]. A logarithmic spiral $(\bar{x}(\alpha) + j\bar{y}(\alpha) = -2(\sin(6.28\alpha) + j\cos(6.28\alpha))e^{-2.49\alpha})$ is stabilized in Fig. 6(a) by using a complex-valued reaction–advection–diffusion equation $(1 - j, 25/4 - j3/4, 9\pi^2/4 -$

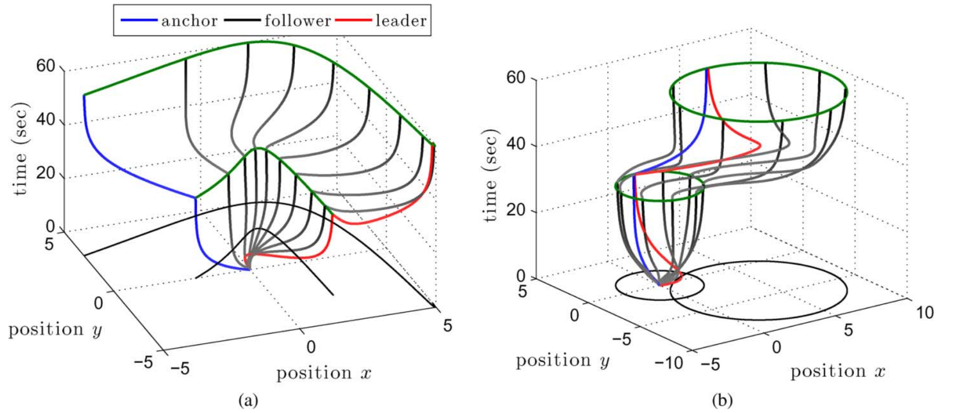

Fig. 5. Agent trajectories (a) using an advection–diffusion equation and a reaction–diffusion equation for the x- and y-deployments, respectively, and (b) for the complex-valued circle deployment family when deploying to a circle centered at (0, 0) and to another centered at (4, −4).

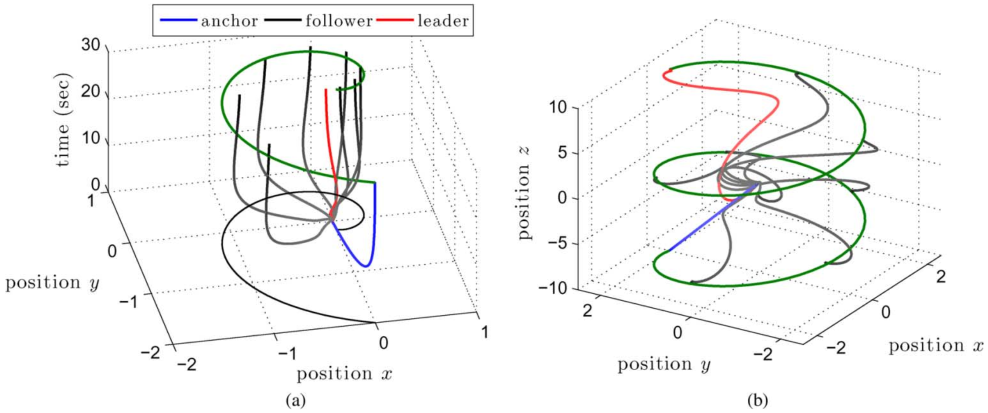

Fig. 6. Agent trajectories for (a) the complex-valued spiral deployment family when deploying to a logarithmic spiral, and (b) a helical deployment with $t \in [0, 30]$ s.

$j11\pi^2/4, 0.301, 0.486)$. For this deployment family, the desired deployment is not precisely reached due to the approximation error incurred by deploying 10 agents (whereas the deployment error in the other examples is negligible for 10 agents). This error is significantly reduced by deploying 15 agents and is indiscernible when deploying 30 agents. The final leader-enabled example [Fig. 6(b)] is a 3-D deployment to a helix $[\bar{x}(\alpha) = -2\sin(4\pi\alpha)$, $\bar{y}(\alpha) = 2\cos(4\pi\alpha)$, $\bar{z}(\alpha) = -10 + 20\alpha]$, which is created by combining a deployment derived from a complex-valued advection–diffusion equation $(1/2 + j/2, 2\pi - j2\pi, 0, 0.699, 0.1481)$ with a linear deployment generated by the 1-D heat equation.

Requiring a leader agent to achieve these deployments does give rise to questions about the method's robustness regarding deployment stabilization when faced with agent failures. We categorize a failure as an agent that stops moving prematurely, which leads to this failed position being continually used in the feedback laws of the failed agent's neighbors. Consequently, the failure of an agent can cause an unstable collective response when employing an open-loop unstable PDE model. If the failed agent is the anchor, however, its failure has the same effect as selecting a different value for its bias term, causing the agents to stabilize another (unintended) deployment. If the PDE model is open-loop stable, the agents achieve a stable, but again unintended, deployment no matter which agent fails since a failed agent acts like an anchor, splitting the deployment into two formations—one between the anchor and the failed agent and another between the failed agent and the leader. In short, the failure of any agent may prevent the agents from achieving the desired deployment, but an unstable response is possible only when an open-loop unstable PDE model is employed.

If the agents have the ability to detect failed agents (that are not the leader), remove them from the topology, and reconnect the topology, the agents would stabilize a deployment that differs slightly from the desired one since the agents' gains are based on the initial number of deployed agents. If the leader were to fail, the agents could elect a new leader, but the user would have to provide this new leader (a former follower agent) with the necessary desired deployment information.

Another concern is collision avoidance. To prevent collisions, potential field-based controllers (or other suitable controllers) could be appended to the controllers used for deployment, but modifications would be needed for desired deployments that have collocated agents. As mentioned at the end of Section VI, this methodology can also be used as a planning algorithm where $z(\alpha, t)$ is a nonphysical variable that converges to the desired deployment $\bar{z}(\alpha)$ while the agents are stationary. Once $z(\alpha, t)$ has converged, each agent effectively knows its desired position and may deploy to this position using any controller that prevents collisions.

VIII. Deployment Onto Circular Arcs

The previous sections have discussed families of planar curves that are stabilized by leader feedback, where the leader is responsible for: 1) achieving closed-loop stability; 2) selecting a specific deployment from the feasible family; and 3) setting the convergence rate of the deployment. If we do not require a specific convergence rate and restrict ourselves to stable systems, two anchor agents can deploy the agents to a deployment family in some cases. For instance, linear deployments can be formed by moving two anchor agents about the plane [3], [11]. The complex-valued circle deployment shown in Section VII also does not require leader feedback for stability, but if two anchor agents are collocated, the agents form a circle whose radius depends on the agents' initial positions, which is undesirable. Many stable schemes have been presented for the stable formation of circles/regular polygons without leader agents, such as cyclic pursuit [5], [6], which does not produce stationary formations, and consensus-based results [2]–[4], which require each agent to know the desired formation. We now explore an alternative approach that utilizes a nonlinear PDE model to achieve a family of circular arc deployments, including full circles, where only the desired radius R needs to be specified for each agent.

Denote the position of agent α at time t as $(x(\alpha,t), y(\alpha,t))$ where $\alpha \in [0,1]$, and consider agents with single integrator dynamics, namely, $x_t(\alpha,t) = u(\alpha,t), y_t(\alpha,t) = v(\alpha,t)$ where u and v are the agents' controllers. In the sequel, we omit (α,t) for brevity unless needed for clarity. We propose controllers u and v, which lead to the nonlinear PDE model

$$\begin{aligned} x_t = &- \frac{c}{2}x(x^2+y^2-R^2) - \varepsilon y \frac{xy_{\alpha\alpha} - yx_{\alpha\alpha}}{x^2+y^2} \\ &- 2\varepsilon y \frac{xy\left(x_\alpha^2 - y_\alpha^2\right) - x_\alpha y_\alpha (x^2-y^2)}{(x^2+y^2)^2} \end{aligned} \tag{79}$$

$$\begin{aligned} y_t = &- \frac{c}{2}y(x^2+y^2-R^2) + \varepsilon x \frac{xy_{\alpha\alpha} - yx_{\alpha\alpha}}{x^2+y^2} \\ &+ 2\varepsilon x \frac{xy\left(x_\alpha^2 - y_\alpha^2\right) - x_\alpha y_\alpha (x^2-y^2)}{(x^2+y^2)^2} \end{aligned} \tag{80}$$

where c, ε, $R > 0$, and R is the desired radius of the circular formation. Two anchor agents ($\alpha = 0$, $\alpha = 1$) represent the boundary conditions for (79) and (80) and have the control laws

$$\begin{aligned} x_t(i) = &- \frac{c}{2}x(i)\left(x^2(i)+y^2(i)-R^2\right) \\ &+ \varepsilon y(i)\left(\arctan\frac{y(i)}{x(i)} - \arctan\frac{\bar{y}(i)}{\bar{x}(i)}\right) \end{aligned} \tag{81}$$

$$\begin{aligned} y_t(i) = &- \frac{c}{2}y(i)\left(x^2(i)+y^2(i)-R^2\right) \\ &- \varepsilon x(i)\left(\arctan\frac{y(i)}{x(i)} - \arctan\frac{\bar{y}(i)}{\bar{x}(i)}\right) \end{aligned} \tag{82}$$

where $\bar{x}(i) = R\cos\theta_i$, $\bar{y}(i) = R\sin\theta_i$, θ_i is the anchor's desired angular position on the circle, and $i = 0$ or 1. The control laws (79)–(82) cause the agents to deploy to a circular arc from θ_0 to θ_1 with radius R, provided no agent (including the anchors) is initialized at the origin.

This deployment is seen more readily by converting (79)–(82) to polar coordinates, i.e., $x(\alpha,t) = r(\alpha,t)\cos\theta(\alpha,t)$, $y(\alpha,t) = r(\alpha,t)\sin\theta(\alpha,t)$, where $r(\alpha,t) = \sqrt{x^2(\alpha,t)+y^2(\alpha,t)}$ and $\theta(\alpha,t) = \arctan(y(\alpha,t)/x(\alpha,t))$. In polar coordinates, the control laws are

$$\begin{aligned} u(\alpha,t) = &\frac{c}{2}r(\alpha,t)\cos\theta(\alpha,t)\left(R^2 - r^2(\alpha,t)\right) \\ &- \varepsilon r(\alpha,t)\sin\theta(\alpha,t)\theta_{\alpha\alpha}(\alpha,t) \end{aligned} \tag{83}$$

$$\begin{aligned} v(\alpha,t) = &\frac{c}{2}r(\alpha,t)\sin\theta(\alpha,t)\left(R^2 - r^2(\alpha,t)\right) \\ &+ \varepsilon r(\alpha,t)\cos\theta(\alpha,t)\theta_{\alpha\alpha}(\alpha,t) \end{aligned} \tag{84}$$

which result in the r^2- and θ-subsystems

$$(r^2)_t = - cr^2(r^2 - R^2) \tag{85}$$

$$(r^2)_t(i) = - cr^2(i)\left(r^2(i) - R^2\right) \tag{86}$$

$$\theta_t = \varepsilon\theta_{\alpha\alpha} \tag{87}$$

$$\theta_t(i) = - \varepsilon\left(\theta(i) - \theta_i\right). \tag{88}$$

It should be noted that (85)–(86) is an α-parameterized family of ODEs, whereas (87)–(88) is a PDE.

Lemma 1: For each α, the r^2-subsystem (85) and (86) has the solution

$$r^2(\alpha,t) = \frac{r_0^2(\alpha)R^2}{r_0^2(\alpha) - \tilde{r}_0^2(\alpha)e^{-cR^2t}} \tag{89}$$

where $r_0(\alpha) = r(\alpha,0)$, $\tilde{r}_0^2(\alpha) = r_0^2(\alpha) - R^2$. For every $r_0(\alpha) > 0$, $r(\alpha,t) \to R$ as $t \to \infty$.

Analyzing the θ-subsystem (87) and (88) is more subtle. The equilibrium curves are linear in α, and from the boundary conditions, we have $\theta(i,t) = e^{-\varepsilon t}(\theta(i,0) - \theta_i) + \theta_i$, which implies that the linear deployment $\bar{\theta}(\alpha)$ lies between θ_0 and θ_1. However, $\theta(\alpha,t) = \theta(\alpha,t) + 2k\pi$, $k \in \mathbb{N}$, so there exist countably many equilibria

$$\bar{\theta}(\alpha) = \theta_0 \pm 2k\pi + (\theta_1 - \theta_0 \pm 2l\pi)\alpha, \qquad k,l \in \mathbb{N} \tag{90}$$

that lie "between" the final anchor agent positions.

Lemma 2: The θ-subsystem (87) and (88) is exponentially stable in the H^1 norm at the equilibrium, $\theta(\alpha,t) \equiv \bar{\theta}(\alpha)$, i.e., for all $t \geq 0$, $\Gamma(t) \leq e^{-(\varepsilon t/4)}\Gamma(0)$, where $\varepsilon > 0$

$$\Gamma(t) = \tilde{\theta}^2(0,t) + \tilde{\theta}^2(1,t) + \int_0^1 \tilde{\theta}^2(\alpha,t)d\alpha + \int_0^1 \tilde{\theta}_\alpha^2(\alpha,t)d\alpha \tag{91}$$

$$\tilde{\theta}(\alpha,t) = \theta(\alpha,t) - \bar{\theta}(\alpha). \tag{92}$$

Proof: Working in the error variable $\tilde{\theta}(\alpha,t)$, we obtain the error system $\tilde{\theta}_t(\alpha,t) = \varepsilon\tilde{\theta}_{\alpha\alpha}(\alpha,t)$, with boundary conditions $\tilde{\theta}_t(i,t) = -\varepsilon\tilde{\theta}(i,t)$. Using (91) as a Lyapunov functional, we take the time derivative, substitute the error system, integrate by parts, and apply the Poincaré inequality to obtain

$$\begin{aligned} \dot{\Gamma} \leq &- 2\varepsilon\left(\tilde{\theta}^2(0) + \frac{3}{4}\tilde{\theta}^2(1) + \frac{1}{8}\int_0^1 \tilde{\theta}^2 d\alpha + \frac{1}{2}\int_0^1 \tilde{\theta}_\alpha^2 d\alpha\right) \\ \leq &- \frac{\varepsilon}{4}\Gamma \end{aligned} \tag{93}$$

which implies the stability result by application of the Comparison Lemma [36]. ■

Before providing a stability result for the agents in the xy plane, we state the following lemma.

Lemma 3: $\sup_{\alpha \in [0,1]} |\tilde{\theta}(\alpha)| \leq \sqrt{2\Gamma(t)}$.

Proof: The proof follows after noting $\int_0^\alpha \partial_\alpha((1/2)\tilde{\theta}^2)d\alpha = \int_0^\alpha \tilde{\theta}\tilde{\theta}_\alpha d\alpha$. ■

Proving stability of the r^2-subsystem (85) and (86) via Lyapunov analysis is difficult because the subsystem has two equilibria: $r^2 = 0$, which is unstable, and $r^2 = R^2$, which is stable. We could remove the equilibrium at the origin by modifying the control laws (79) and (80), but then the control laws become singular at the origin. Instead, we consider agents in the domain $\mathcal{D} = \{(\xi, \eta) \in \mathbb{R}^2 | \xi^2 + \eta^2 \geq \epsilon^2, \epsilon > 0\}$ and utilize the explicit solution (89) to give a pointwise convergence result, whose bounds depend on an agent's initial position.

Theorem 3: Each agent α in the system (79) and (80), with boundary conditions (81) and (82), exponentially converges to its equilibrium position on the circle $(x(\alpha,t), y(\alpha,t)) \equiv (R\cos\bar{\theta}(\alpha), R\sin\bar{\theta}(\alpha))$, where $R > 0$ and $\bar{\theta}(\alpha)$ is given in (90). Specifically, for all initial profiles such that $x^2(\alpha,0) + y^2(\alpha,0) \geq \epsilon^2 > 0$, the following bound holds for each agent α for all $t \geq 0$:

$$\max\left(|\tilde{x}(\alpha,t)|, |\tilde{y}(\alpha,t)|\right) \leq F(\alpha)e^{-cR^2 t} + G(\alpha)e^{-\frac{\varepsilon t}{8}} \quad (94)$$

where $\tilde{x}(\alpha,t) = x(\alpha,t) - R\cos\bar{\theta}(\alpha)$, $\tilde{y}(\alpha,t) = y(\alpha,t) - R\sin\bar{\theta}(\alpha)$, and

$$F(\alpha) = \begin{cases} \frac{R}{2\epsilon^2}\left|r_0^2(\alpha) - R^2\right|, & r_0(\alpha) < R \\ \left|r_0(\alpha) - R\right|, & r_0(\alpha) > R \end{cases} \quad (95)$$

$$G(\alpha) = \sqrt{2\Gamma(0)}R\left(\left|\cos\bar{\theta}(\alpha)\right| + \left|\sin\bar{\theta}(\alpha)\right|\right). \quad (96)$$

Proof: We show the details of bounding $\tilde{x}(\alpha,t)$ only. We rewrite $\tilde{x}(\alpha,t)$ by converting to polar coordinates and using (89), (92), some algebra, and a trigonometric identity as

$$\begin{aligned} \tilde{x} = {} & \frac{r_0 R}{\sqrt{r_0^2 - \tilde{r}_0^2 e^{-cR^2 t}}}\cos(\tilde{\theta} + \bar{\theta}) - R\cos(\tilde{\theta} + \bar{\theta}) \\ & - R\cos(\bar{\theta}) + R\cos(\tilde{\theta} + \bar{\theta}) \\ = {} & \frac{\tilde{r}_0^2 R e^{-cR^2 t}}{\sqrt{r_0^2 - \tilde{r}_0^2 e^{-cR^2 t}}\left(r_0 + \sqrt{r_0^2 - \tilde{r}_0^2 e^{-cR^2 t}}\right)} \\ & \times (\cos\tilde{\theta}\cos\bar{\theta} - \sin\tilde{\theta}\sin\bar{\theta}) - R\cos\bar{\theta}(1 - \cos\tilde{\theta}) \\ & - R\sin\bar{\theta}\sin\tilde{\theta}. \end{aligned} \quad (97)$$

From this expression, we develop the bound

$$\begin{aligned} |\tilde{x}| \leq {} & \left|\frac{1}{\sqrt{r_0^2 - \tilde{r}_0^2 e^{-cR^2 t}}}\right| \left|\frac{1}{r_0 + \sqrt{r_0^2 - \tilde{r}_0^2 e^{-cR^2 t}}}\right| \\ & \times \left|\tilde{r}_0^2\right| R e^{-cR^2 t} + R\left(|\cos\bar{\theta}| + |\sin\bar{\theta}|\right)|\tilde{\theta}|. \end{aligned} \quad (98)$$

When $r_0(\alpha) < R$, the first and second terms of (98) can be bounded by

$$\left|\frac{1}{\sqrt{r_0^2 + (R^2 - r_0^2)e^{-cR^2 t}}}\right| \leq \frac{1}{r_0} \leq \frac{1}{\epsilon} \quad (99)$$

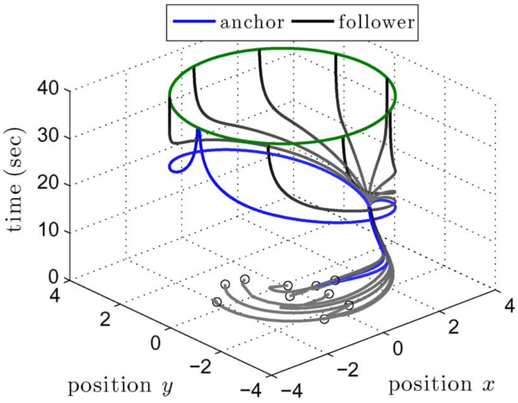

Fig. 7. Agent trajectories for rendezvous on a circle at the point (3,0) using the control laws (79)–(82). At $t = 10$ s, the anchor agents traverse the circle, creating a circle formation.

$$\left|\frac{1}{r_0 + \sqrt{r_0^2 + (R^2 - r_0^2)e^{-cR^2 t}}}\right| \leq \frac{1}{2r_0} \leq \frac{1}{2\epsilon}. \quad (100)$$

If $r_0(\alpha) > R$, we have

$$\left|\frac{1}{\sqrt{r_0^2 - (r_0^2 - R^2)e^{-cR^2 t}}}\right| \leq \frac{1}{R} \quad (101)$$

$$\left|\frac{1}{r_0 + \sqrt{r_0^2 - (r_0^2 - R^2)e^{-cR^2 t}}}\right| \leq \frac{1}{r_0 + R}. \quad (102)$$

Using inequalities (99)–(102) with Lemmas 2 and 3, we obtain the bound $|\tilde{x}| \leq F(\alpha)e^{-cR^2 t} + G(\alpha)e^{-(\varepsilon t/8)}$. ■

We have presented the model (79) and (80) with anchor agents since it enables the agents to deploy onto a circular arc that, by moving the anchor agents, can be rotated about the circle or elongated/shortened on the circle of radius R. Collocating the anchor agents causes the agents to either form a circle or to rendezvous, but the anchors cannot determine which outcome will occur since the system has multiple equilibria. If a circle formation is desired, but the agents' initial positions dictate that the agents rendezvous instead, the anchors can transverse the circle in opposite directions until they are again collocated, dragging the followers into a circle formation.

Fig. 7 depicts this scenario for 10 agents that employ discretized versions of (79)–(82) with $c = \varepsilon = 5h/2$ and $R = 3$. The initial positions (black circles) are sampled from the Gaussian distribution $\mathcal{N}(0,1)$. For $t \in [0,10)$ s, setting $\theta_0 = \theta_1 = 0$ rad, given the initial positions, causes the agents to rendezvous at the point (3,0). For $t \in [10,40]$ s, setting $\theta_0 = \min\{(\pi/15)t - (2\pi/3), \pi\}$ rad and $\theta_1 = -\min\{(\pi/15)t - (2\pi/3), \pi\}$ rad forces the anchors—and, consequently, the followers—to move about the circle until a circle formation is formed. Since the r^2-subsystem is fully decentralized, an agent will obtain the radial position R irrespective of the other agents' motion. The anchor agents (or a failed agent) can affect only the angular distribution of the agents, which means a deployment's radius cannot be changed unless the parameter R is changed for all the agents. In contrast, the agents can achieve circle/ellipse deployments of any size when using the leader-based approach.

The same model, but with periodic boundary conditions, i.e., a ring communication topology, is truly leaderless as it lacks anchor agents and, depending on the agents' initial positions, stabilizes only a circle or a rendezvous on the circle. This system's stability analysis is very much the same since only the boundary conditions differ from (79)–(82). Without anchors, the agents cannot form arcs or change deployments, and only with a leader can the agents stabilize a specific deployment for any initial condition.

IX. Conclusion

We have introduced a PDE-based approach for the deployment of agents onto families of planar curves. The PDE models are a spatial application of the internal model principle, meaning they make a deployment family feasible, but do not guarantee stability. While the standard diffusion-based feedback leads to inherently stable deployment to linear formations, for which leader assistance is not needed, the nonlinear (in α) curves that we pursue may be open-loop unstable. We employ a leader and anchor agent, whose controllers are designed using PDE boundary control techniques, to select and stabilize a desired deployment from the feasible deployments.

We also introduced a family of circular arcs derived from a nonlinear PDE model that does not require leader feedback for stability. This approach extends the stable linear deployments derived from diffusion-based feedback by enabling the agents to stably deploy to a circle. The formation, however, is limited to a fixed, predetermined radius.

It is crucial to observe that, in our framework, the follower agents deploy out and maintain a stable formation not by being commanded some reference positions, but by being induced (indirectly influenced) by the leader's and anchor's motions. Such a paradigm allows a user to control the formation geometry of an entire group of agents by adjusting the bias terms of only two agents. With a leader agent, the agents can stably deploy to the rich geometric family of formations derived from the reaction–advection–diffusion PDE class, which includes several deployments of practical interest—for example, the leader deploying out the majority of agents near a target position (occupied by the anchor) while staying at base, or conversely, deploying out only a few agents; deploying agents out in both directions symmetrically while both the leader and anchor stay at base, creating a "protective shell" for the leader; or deploying the agents to encircle a point of interest for surveillance or escort purposes.

If both the boundary agents are treated as virtual agents, more traditional Dirichlet and Neumann boundary conditions can be used, which may be necessary to extend this work to the stabilization of deployments governed by nonlinear PDE models such as the Burgers equation, which allows for shock-like deployments. Extending this paradigm to planar surface deployments and performing motion planning to control the agents' transient behavior when deploying are also of interest.

Appendix

Lemma 4: The inequalities, $p_1\Omega(t) \leq \Psi(t) \leq p_2\Omega(t)$ and $q_1\Psi(t) \leq V(t) \leq q_2\Psi(t)$ hold, where $\Omega(t)$ and $V(t)$ are given by (37) and (38), $\Psi(t) = |w(0,t)|^2 + |w(1,t)|^2 + \int_0^1 |w(\alpha,t)|^2 d\alpha + \int_0^1 |w_\alpha(\alpha,t)|^2 d\alpha$

$$p_1 = \left[\max\left(2 + \left|\frac{b}{\varepsilon}\right|^2 + \left(12 + \left|\frac{b}{\varepsilon}\right|^2\right) L + 8L_\alpha, 4\right) \times \max\left(1, \left|e^{-\frac{b}{2\varepsilon}}\right|^2\right)\right]^{-1} \tag{103}$$

$$p_2 = \max\left(1, \left|e^{\frac{b}{2\varepsilon}}\right|^2\right) \times \max\left(2 + \left|\frac{b}{\varepsilon}\right|^2 + 8K + 4K_\alpha, 4\right) \tag{104}$$

$q_1 = (1/16)\min(1, 8m)$, $q_2 = (1/2)\max(1, m)$, $L = \sup_{(\alpha,\beta)\in\mathcal{T}} |l(\alpha,\beta)|^2$, $L_\alpha = \sup_{(\alpha,\beta)\in\mathcal{T}} |l_\alpha(\alpha,\beta)|^2$, $K = \sup_{(\alpha,\beta)\in\mathcal{T}} |k(\alpha,\beta)|^2$, and $K_\alpha = \sup_{(\alpha,\beta)\in\mathcal{T}} |k_\alpha(\alpha,\beta)|^2$.

Proof: Using (22), (26), (31), and applying the Cauchy–Schwarz and Young's inequalities several times to bound terms in $\Psi(t)$ and $\Omega(t)$, one can obtain (103) and (104). The scalar q_2 is immediate from $V(t)$ and $\Psi(t)$, and q_1 is found by splitting the last term of $V(t)$ and applying the Poincaré inequality. ■

Lemma 5: The inequalities $r_1\Sigma(t) \leq \Phi(t) \leq r_2\Sigma(t)$ and $\Phi(t) \leq W(t) \leq s_2\Phi(t)$ hold where $\Sigma(t)$ and $W(t)$ are given by (60) and (61), $\Phi(t) = |\tilde{w}(0,t)|^2 + |\tilde{w}(1,t)|^2 + \int_0^1 |\tilde{w}(\alpha,t)|^2 d\alpha + \int_0^1 |\tilde{w}_\alpha(\alpha,t)|^2 d\alpha$

$$r_1 = \left[\max\left(2 + \left|\frac{b}{\varepsilon}\right|^2 + \left(12 + \left|\frac{b}{\varepsilon}\right|^2\right) P + 8P_\alpha, 4\right) \times \max\left(1, \left|e^{-\frac{b}{2\varepsilon}}\right|^2\right)\right]^{-1} \tag{105}$$

$$r_2 = \max\left(1, \left|e^{\frac{b}{2\varepsilon}}\right|^2\right) \max\left(2 + \left|\frac{b}{\varepsilon}\right|^2 + 8Q + 4Q_\alpha, 4\right) \tag{106}$$

$s_1 = 1/16$, $s_2 = 1/2$, $P = \sup_{(\alpha,\beta)\in\mathcal{S}} |p(\alpha,\beta)|^2$, $P_\alpha = \sup_{(\alpha,\beta)\in\mathcal{S}} |p_\alpha(\alpha,\beta)|^2$, $Q = \sup_{(\alpha,\beta)\in\mathcal{S}} |q(\alpha,\beta)|^2$, and $Q_\alpha = \sup_{(\alpha,\beta)\in\mathcal{S}} |q_\alpha(\alpha,\beta)|^2$.

Proof: The proof is analogous to the proof of Lemma 4. ■

References

[1] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1347–1363, 1999.

[2] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[3] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Trans. Autom. Control*, vol. 50, no. 1, pp. 121–127, Jan. 2005.

[4] G. Lafferriere, A. Williams, J. Caughman, and J. J. P. Veerman, "Decentralized control of vehicle formations," *Syst. Control Lett.*, vol. 54, no. 9, pp. 899–910, 2005.

[5] J. A. Marshall, M. E. Broucke, and B. A. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Trans. Autom. Control*, vol. 49, no. 11, pp. 1963–1974, Nov. 2004.

[6] M. Pavone and E. Frazzoli, "Decentralized policies for geometric pattern formation and path coverage," *J. Dynam. Syst., Meas., Control*, vol. 129, pp. 633–643, 2007.

[7] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. Robot. Autom.*, vol. 19, no. 6, pp. 933–941, Dec. 2003.

[8] E. W. Justh and P. S. Krishnaprasad, "Equilibria and steering laws for planar formations," *Syst. Control Lett.*, vol. 52, no. 1, pp. 25–38, 2004.

[9] J. Cortés, S. Martínez, T. Karataç, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[10] M. A. Hsieh, V. Kumar, and L. Chaimowicz, "Decentralized controllers for shape generation with robotic swarms," *Robotica*, vol. 26, no. 5, pp. 691–701, 2008.

[11] R. Carli and F. Bullo, "Quantized coordination algorithms for rendezvous and deployment," *SIAM J. Control Optim.*, vol. 48, no. 3, pp. 1251–1274, 2009.

[12] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. IEEE Conf. Decision Control*, Orlando, FL, Dec. 2001, pp. 2968–2973.

[13] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, Belgium, May 1998, pp. 2864–2869.

[14] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 443–455, Jun. 2004.

[15] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader-follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.

[16] S. Björkenstam, M. Ji, M. Egerstedt, and C. Martin, Allerton House, UIUC, "Leader-based multi-agent coordination through hybrid optimal control," in *Proc. Allerton Conf. Commun., Control, Comput.*, Sep. 2006, pp. 1352–1357.

[17] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa, "Containment control in mobile networks," *IEEE Trans. Autom. Control*, vol. 53, no. 8, pp. 1972–1975, Sep. 2008.

[18] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005 [Online]. Available: http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2005/Fullpapers/04550.pdf

[19] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic consensus estimators," in *Proc. IEEE Conf. Decision Control*, San Diego, CA, Dec. 2006, pp. 338–343.

[20] P. Yang, R. A. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2480–2496, Dec. 2008.

[21] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of planar collective motion with limited communication," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 706–719, Apr. 2008.

[22] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of planar collective motion: All-to-all communication," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 811–824, May 2007.

[23] E. W. Justh and P. S. Krishnaprasad, "Steering laws and continuum models for planar formations," in *Proc. IEEE Conf. Decision Control*, Maui, HI, Dec. 2003, pp. 3609–3614.

[24] A. L. Bertozzi, M. Kemp, and D. Marthaler, "Determining environmental boundaries: Asynchronous communication and physical scales," in *Cooperative Control*, ser. Lecture Notes in Control and Information Sciences, V. Kumar, N. E. Leonard, and A. S. Morse, Eds. Berlin, Germany: Springer, 2004, vol. 309, pp. 25–42.

[25] S. Kalantar and U. R. Zimmer, "Distributed shape control of homogeneous swarms of autonomous underwater vehicles," *Auton. Robots*, vol. 22, no. 1, pp. 37–53, 2007.

[26] L. C. A. Pimenta, N. Michael, R. C. Mesquita, G. A. S. Pereira, and V. Kumar, "Control of swarms based on hydrodynamic models," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, May 2008, pp. 1948–1953.

[27] P. Barooah, P. G. Mehta, and J. P. Hespanha, "Mistuning-based control design to improve closed-loop stability margin of vehicular platoons," *IEEE Trans. Autom. Control*, vol. 54, no. 9, pp. 2100–2113, Sep. 2009.

[28] A. Sarlette and R. Sepulchre, "A PDE viewpoint on basic properties of coordination algorithms with symmetries," in *Proc. IEEE Int. Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 5139–5144.

[29] G. Ferrari-Trecate, A. Buffa, and M. Gati, "Analysis of coordination in multi-agent systems through partial difference equations," *IEEE Trans. Autom. Control*, vol. 51, no. 6, pp. 918–928, Jun. 2006.

[30] L. Moreau, "Stability of continuous-time distributed consensus algorithms," in *Proc. IEEE Conf. Decision Control*, Atlantis, Paradise Island, Bahamas, Dec. 2004, pp. 3998–4003.

[31] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[32] J. Kim, K.-D. Kim, V. Natarajan, S. D. Kelly, and J. Bentsman, "PdE-based model reference adaptive control of uncertain heterogeneous multiagent networks," *Nonlinear Anal., Hybrid Syst.*, vol. 2, no. 4, pp. 1152–1167, 2008.

[33] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.

[34] A. Smyshlyaev and M. Krstic, "Closed-form boundary state feedbacks for a class of 1-D partial integro-differential equations," *IEEE Trans. Autom. Control*, vol. 49, no. 12, pp. 2185–2204, Dec. 2004.

[35] A. Smyshlyaev and M. Krstic, "Backstepping observers for a class of parabolic PDEs," *Syst. Control Lett.*, vol. 54, no. 7, pp. 613–625, 2005.

[36] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[37] A. Balogh and M. Krstic, "Infinite dimensional backstepping-style feedback transformations for a heat equation with an arbitrary level of instability," *Eur. J. Control*, vol. 8, pp. 165–175, 2002.

[38] A. Balogh and M. Krstic, "Stability of partial difference equations governing control gains in infinite-dimensional backstepping," *Syst. Control Lett.*, vol. 51, no. 2, pp. 151–164, 2004.

[39] A. Smyshlyaev and M. Krstic, "Lyapunov adaptive boundary control for parabolic pdes with spatially varying coefficients," in *Proc. Amer. Control Conf.*, Minneapolis, MN, Jun. 2006, pp. 41–48.

[40] C. Pozrikidis, *Numerical Computation in Science and Engineering*, 1st ed. New York: Oxford Univ. Press, 1998.

[41] L. E. Parker, B. Kannan, X. Fu, and Y. Tang, "Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, Oct. 2003, pp. 681–689.

Paul Frihauf (S'09) received the B.S. and M.S. degrees in systems science and engineering from Washington University in St. Louis, MO, in 2005, and is currently pursuing the Ph.D. degree in dynamic systems and control from the University of California, San Diego.

He was an Industrial Engineering Intern with the Boeing Company, Seattle, WA, in 2003; a Summer Research Assistant at MIT Lincoln Laboratory, Lexington, MA, in 2004; and Associate Professional Staff with the Guidance, Navigation, and Control group at the Johns Hopkins University Applied Physics Laboratory, Laurel, MD, from 2005 to 2007. His research interests include noncooperative games, multiagent control, and the control of distributed parameter systems.

Mr. Frihauf is a member of SIAM and is a National Defense Science and Engineering Graduate (NDSEG) Fellow.

Miroslav Krstic (S'92–M'95–SM'99–F'02) received the Ph.D. degree in electrical engineering from the University of California, Santa Barbara, in 1994.

He was an Assistant Professor with the University of Maryland, College Park, until 1997. He is the Daniel L. Alspach Professor and the founding Director of the Cymer Center for Control Systems and Dynamics (CCSD) at the University of California, San Diego. He has held the appointment of Springer Distinguished Visiting Professor of Mechanical Engineering at the University of California, Berkeley. He is a coauthor of eight books: *Nonlinear and Adaptive Control Design* (Wiley, 1995), *Stabilization of Nonlinear Uncertain Systems* (Springer, 1998), *Flow Control by Feedback* (Springer, 2002), *Real-time Optimization by Extremum Seeking Control* (Wiley, 2003), *Control of Turbulent and Magnetohydrodynamic Channel Flows* (Birkhauser, 2007), *Boundary Control of PDEs: A Course on Backstepping Designs* (SIAM, 2008), *Delay Compensation for Nonlinear, Adaptive, and PDE Systems* (Birkhauser, 2009), and *Adaptive Control of Parabolic PDEs* (Princeton Univ. Press, 2009).

Dr. Krstic is a Fellow of IFAC and has received the Axelby and Schuck Paper Prizes, NSF Career, ONR Young Investigator, and PECASE Award.