Modeling Bird Flight Formations Using Diffusion Adaptation

Federico S. Cattivelli, *Member, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

***Abstract*—Flocks of birds self-organize into V-formations when they need to travel long distances. It has been shown that this formation allows the birds to save energy, by taking advantage of the upwash generated by the neighboring birds. In this work we use a model for the upwash generated by a flying bird, and show that a flock of birds can self-organize into a V-formation if every bird were to process spatial and network information through an adaptive diffusive process. The diffusion algorithm requires the birds to obtain measurements of the upwash, and also to use information from neighboring birds. The result has interesting implications. First, a simple diffusion algorithm can account for self-organization in birds. The algorithm is fully distributed and runs in real time. Second, according to the model, that birds can self-organize based on the upwash generated by the other birds. Third, that some form of information sharing among birds is necessary to achieve flight formation. We also propose a modification to the algorithm that allows birds to organize into a U-formation, starting from a V-formation. We show that this type of formation leads to an equalization effect, where every bird in the flock observes approximately the same upwash.**

***Index Terms*—Adaptive networks, bird flight, distributed estimation, diffusion LMS, diffusion networks, U-formation, V-formation, self-organization.**

I. Introduction

SELF-ORGANIZATION is a remarkable property of nature and it has been observed in several physical and biological systems. Examples include fish joining together in schools, chemicals forming spirals, and sand grains assembling into rippling dunes [2]. In self-organizing systems, a global pattern emerges from the interactions of the individual components of the system.

Biologically inspired techniques have been advanced in the literature. For example, Ant Colony Optimization [3] is based on how ants organize in order to find the shortest path to food, and Particle Swarm Optimization [4] is based on how birds flock to find food. Both algorithms have been applied to solve different optimization problems [5]. Algorithms based on how fireflies synchronize have also been proposed for wireless network synchronization [6], [7]. Flocking behavior, vehicle formation and multi-agent systems have been studied before in the control literature [8], [9]. The work [9] studies flocking behavior of multi-agent systems, where the agents attempt to stay close to each other and avoid obstacles. The specific formation used by the flock is not considered. The work [8] studies flock formations based on distributed, consensus-based control strategies. The major difference between these earlier works and the approach we pursue in this paper is our emphasis on the role of adaptation over networks. The previous works (e.g., [8] and [9]) focus on determining motion control mechanisms that lead to flocking behavior, but do not deal with estimation and tracking from noisy measurements. In comparison, this paper focuses on real-time, local information processing and local information sharing among network agents, where adaptation is critical for learning, tracking, and to handle noisy data. What the current work shows is that adaptation, local information processing and local information exchange over networks can lead to flocking in the presence of noisy data.

In this paper, we focus on bird flocks, and specifically on V-shaped formations obtained during bird flight. It has been argued before [10] that birds form into V-shapes in order to save energy. The reason, although yet debated, is that a flying bird generates a pair of trailing vortices, which a trailing bird can use to maintain its altitude and save energy. Still, what algorithm is employed by the birds to get into this formation is unknown.

In order to emulate these formations, we shall employ a distributed estimation algorithm over cognitive, adaptive networks that is based on diffusion LMS adaptation as proposed and studied in [13], [14], [17], [19], [21], [23], [24]. Diffusion algorithms are based on the principle that nodes should obtain their estimates by sharing information only with their neighbors. This is achieved in two steps: local processing and merging of information using convex combinations of adaptive estimates [14], [24]. Diffusion algorithms only require local communications between neighboring nodes, and can attain good estimation performance compared to centralized solutions. These algorithms are useful for minimizing cost functions in a distributed manner and their performance has been studied in some detail in [13] and [14]. In this work, we focus on the Adapt-then-Combine (ATC) version of the diffusion LMS algorithm proposed in [14]. The diffusion LMS algorithm was originally proposed in [17], [23], and [24], and later generalized in [14], which also introduced the ATC version by reverting the order in which adaptation and combination are performed.

Manuscript received May 12, 2010; revised November 02, 2010; accepted December 29, 2010. Date of publication January 24, 2011; date of current version April 13, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Roberto Lopez-Valcarce. This material was based on work supported in part by the National Science Foundation under awards ECS-0725441, CCF-0942936, and CCF-1011918. A short version of this work appeared in the conference publication *Proceedings of the International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP),* Aruba, Dutch Antilles, December 2009, pages 49–52.

The authors are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: fcattiv@ee.ucla.edu; sayed@ee.ucla.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/TSP.2011.2107907

The subsequent work [25] is related to the ATC diffusion LMS algorithm without information exchange. Distributed estimation algorithms based on consensus strategies have also been considered in [26] and [27]. However, diffusion strategies generally provide superior adaptation and learning abilities, as already indicated in [14] and [28]. In this paper, we show how to develop an appropriate model for flight formations that fits well with diffusion strategies. We also study the performance and show how to organize in V- and U-formations.

This paper is organized as follows. We start by reviewing the diffusion LMS algorithm in Section II and then describe the aerodynamic model to be employed in Section III. The proposed model is based on a horseshoe vortex with a Burnham-Hallock profile, with some simplifications. Then, in Section IV, we show how we can use the diffusion LMS algorithm to solve the problem of collectively estimating the optimal distance of a trailing bird with respect to its leading bird. Subsequently, we provide an algorithm for V-formations based on this estimation technique. In Section V we analyze the steady-state behavior of the algorithm, and present simulation results in Section VI, showing that if birds were to employ diffusion adaptation, they would be able to form in V-formations. Finally, in Section VII, we study U-formations as well and propose an algorithm for self-organization into a U-shape, if the flock starts from a V-shape. We also provide some insight into why these formations might be useful in nature.

II. The Diffusion LMS Algorithm

The diffusion LMS algorithm [13], [14], [18], [19], [23], [24] is a distributed estimation scheme that allows every node in a network to estimate an unknown parameter from local measurements and local interactions with neighboring nodes. In this work we use the ATC version of the diffusion LMS algorithm proposed in [14], [18], though we will refer to it simply as diffusion LMS. We use bold notation for random quantities and regular font for nonrandom quantities. The symbol $\mathbb{1}$ indicates a vector of all ones.

Consider a set of N nodes distributed over some region. We say that two nodes are connected if they can communicate directly with each other. Every node is always connected to itself. The set of nodes connected to node k is called the *neighborhood* of node k, and is denoted by $\mathcal{N}_k$. It is assumed that at every time instant i, every node k measures a scalar $d_k(i)$, drawn from some random process $\boldsymbol{d}_k(i)$, and a row regression vector $u_{k,i}$ of size M, drawn from a random process $\boldsymbol{u}_{k,i}$, which are related to an unknown vector w^o of size M as follows:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i). \tag{1}$$

It is assumed that $\boldsymbol{u}_{k,i}$ is wide-sense stationary with mean zero and covariance matrix $R_{u,k} = \mathrm{E}\boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i} > 0$, and $\boldsymbol{v}_k(i)$ is a scalar, zero-mean random process, independent of $\boldsymbol{u}_{k,i}$ and uncorrelated in time and space, i.e., $\mathrm{E}\boldsymbol{v}_k(i)\boldsymbol{v}_l(j) = \delta_{kl}\delta_{ij}\sigma_{v_k}^2$. The operator E denotes expectation, * denotes conjugate transposition, and δ_{kl} is the Kronecker delta. We estimate w^o by seeking the minimizer of the sum of the mean-square errors across all nodes [13], [14], [17], i.e.,

$$w^o = \arg\min_w \sum_{k=1}^{N} \mathrm{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w|^2.$$

The diffusion LMS algorithm allows every node in the network to obtain an estimate of the unknown parameter w^o from a linear observation model as in (1), by communicating only with their neighbors. The estimate obtained by node k at time i is denoted by $w_{k,i}$. The algorithm uses a so-called *diffusion* matrix A of size N by N, with nonnegative real entries $a_{l,k}$ satisfying

$$a_{l,k} = 0 \quad \text{if } l \notin \mathcal{N}_k \quad \mathbb{1}^T A = \mathbb{1}^T. \tag{2}$$

The ATC version of the algorithm without measurement exchange is shown below for convenience. Notice that nodes only need to communicate to their neighbors the vectors $\psi_{k,i}$ of size M.

Algorithm 1: ATC Diffusion LMS (no measurement exchange) [14]

Start with $\{w_{l,-1} = 0\}$ for all l. Given nonnegative real coefficients $\{a_{l,k}\}$ satisfying (2), for each time $i \geq 0$ and for each node k, repeat:

Step 1: Incremental step

$$\psi_{k,i} = w_{k,i-1} + \mu_k u_{k,i}^* (d_k(i) - u_{k,i} w_{k,i-1})$$

Every bird k exchanges $\psi_{k,i}$ with its neighbors

Step 2: Diffusion step

$$w_{k,i} = \textstyle\sum_{l \in \mathcal{N}_k} a_{l,k} \psi_{l,i}$$

Thus, in the incremental step of Algorithm 1, node k obtains an intermediate estimate of w^o, denoted by $\psi_{k,i}$, by running an LMS-type update on the data $\{d_k(i), u_{k,i}\}$. In the diffusion step, nodes exchange their intermediate estimates with their neighbors, and combine these estimates in a convex manner in order to obtain the final estimate $w_{k,i}$.

III. Aerodynamic Model

A. Vortex Model

A flying bird generates trailing vortices due to the interaction between its wings and the atmosphere [29]–[31]. Several models for these vortices have been considered before in the literature. One of the most basic and widely used vortex models is the so-called horseshoe model [32], [33]. In this model, the vortex sheet behind the bird is modeled as a pair of infinite trailing vortices, together with a bound vortex at the location of the wing. This configuration is shown schematically in Fig. 1, where Γ is the vortex circulation in m^2/s, V is the bird velocity, and a is the separation between the trailing vortices. A reasonable value for the separation of the trailing vortices is $a = b\pi/4 \approx 0.785b$, where b is the wingspan of the bird [30]. Thus, we will use this approximation in our model, and assume that a bird generates two semi-infinite trailing vortices with the same circulation and opposite directions, spaced $a = b\pi/4$ apart. Moreover, since we expect neighboring birds to be at a

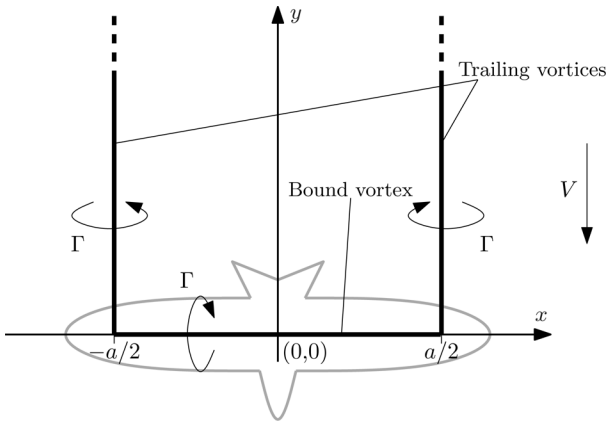

Fig. 1. Horseshoe vortex model.

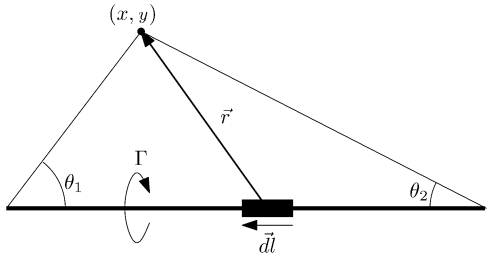

Fig. 2. Filament vortex.

certain safe distance from each other, we will also ignore the contribution of the bound vortex to the total model.

A semi-infinite vortex line will generate a field of induced velocities in its proximity. Several models for the induced velocity have been proposed before in the literature. One of the simplest vortex models is derived from Biot-Savart's law as follows. Consider a filament of vortex as shown in Fig. 2. According to Biot–Savart's law the velocity induced by an infinitesimal vortex of length $\mathrm{d}l$ and circulation Γ is given at location $\vec{r}$ by [29], [33]

$$\mathrm{d}\vec{v}(\vec{r}) = \frac{\Gamma}{4\pi} \cdot \frac{\vec{\mathrm{d}l} \times \vec{r}}{|\vec{r}|^3}.$$

Thus, the total velocity induced at a point located in the (x, y) plane at a distance h from a single vortex line is given by [29]

$$v_{\text{single}}(h, \theta_1, \theta_2) = \frac{\Gamma}{4\pi h} \cdot (\cos\theta_1 + \cos\theta_2). \tag{3}$$

The direction of this velocity is perpendicular to the plane of the page in Fig. 2. For a single semi-infinite vortex line starting at point $(0, 0)$ and extending in the direction of positive y, we have $\theta_2 = 0$ and therefore

$$v_{\text{single}}(x, y) = \frac{\Gamma}{4\pi x} \cdot \left(1 + \frac{y}{\sqrt{x^2 + y^2}}\right). \tag{4}$$

At an infinite distance from the bound vortex (as $y \to \infty$), we have that the induced velocity is of the form

$$v_{\text{single}}(x, \infty) = \frac{\Gamma}{2\pi x}. \tag{5}$$

Although (5) is an appropriate model when x is large, it produces infinite velocities at the location $x = 0$, and therefore is not physically meaningful in the proximity of the vortex. For this reason, other velocity models have been proposed in the literature, including the Rankine model [34]

$$v_{\text{single}}(x, \infty) = \begin{cases} \frac{\Gamma}{2\pi x} & |x| > r_c \\ \frac{\Gamma x}{2\pi r_c^2} & |x| < r_c, \end{cases} \tag{6}$$

the NASA–Burnham–Hallock model [35]

$$v_{\text{single}}(x, \infty) = \frac{\Gamma}{2\pi} \cdot \frac{x}{r_c^2 + x^2}, \tag{7}$$

and the Lamb–Oseen model [36]

$$v_{\text{single}}(x, \infty) = \frac{\Gamma}{2\pi x} \cdot (1 - e^{-x^2/r_c^2}) \tag{8}$$

where in all cases, r_c is a parameter known as the core radius. Notice that the Lamb–Oseen model approaches the Burnham–Hallock model for small x, and both models approach the model obtained through Biot–Savart's law for large x [37].

In this work, we use a model based on the NASA–Burnham–Hallock model, since it has been shown to provide good results in practice [38]. In order to account for the vortex decay in the proximity of the originating wing (i.e., as $y \to 0$), we incorporate a term $(1 + \cos\theta_1)$ as in (4). This leads to the model [35], [38]

$$v_{\text{single}}(x, y) = \frac{\Gamma}{2\pi} \cdot \frac{x}{r_c^2 + x^2} \left(1 + \frac{y}{\sqrt{x^2 + y^2}}\right). \tag{9}$$

Also, in order to simplify our subsequent analysis, we will modify the velocity (9) to be separable in the x and y coordinates, i.e., to obtain a velocity of the form $v(x, y) = v_x(x)v_y(y)$. In order to accomplish this, we will approximate the quantity $\frac{y}{\sqrt{x^2+y^2}}$ as if it were felt by a bird at location $x = \pm b/2$. We shall see later that this is close to the optimal location of the wingtip of a trailing bird. Moreover, it is not reasonable in practice to assume that the vortex will extend infinitely in the y direction [39]. Thus, we will assume a Gaussian vortex strength decay along the y direction, by multiplying by a factor of the form

$$e^{-\frac{(y-\beta)^2}{2\sigma}}$$

where β and σ are some parameters. Putting these facts together, the model for a semi-infinite vortex starting at position $(0, 0)$ and extending in the direction of positive y takes the form:

$$v_{\text{single}}(x, y) = \frac{\Gamma}{2\pi} \frac{x}{r_c^2 + x^2} \left(1 + \frac{y}{\sqrt{(b/2)^2 + y^2}}\right) e^{-\frac{(y-\beta)^2}{2\sigma}}. \tag{10}$$

Notice that we can also write

$$v_{\text{single}}(x, y) = v_x(x)v_y(y) \tag{11}$$

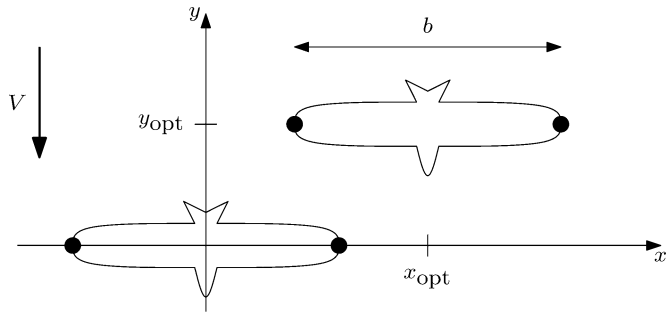

Fig. 3. Bird reference system with respect to leading bird.

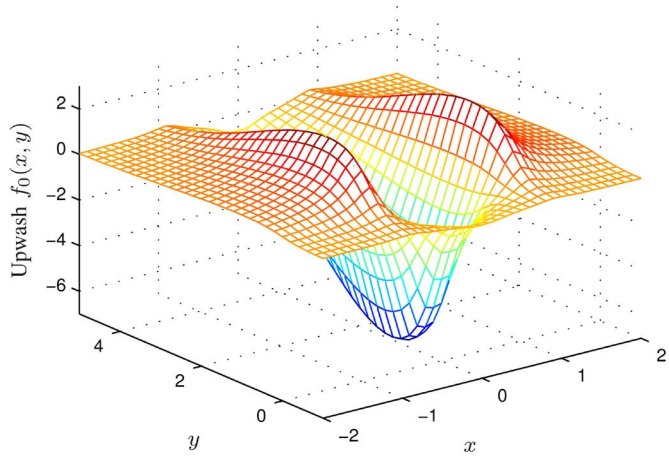

Fig. 4. Upwash generated by a bird located at position $x = 0$, $y = 0$, using $b = 1$, $a = \pi/4$, $r_c = 0.1$, $\beta = 0.7$ and $\sigma = 4$.

where

$$v_x(x) \triangleq \frac{\Gamma}{2\pi} \cdot \left(\frac{x}{r_c^2 + x^2} \right)$$

$$v_y(y) \triangleq \left(1 + \frac{y}{\sqrt{(b/2)^2 + y^2}} \right) e^{-\frac{(y-\beta)^2}{2\sigma}}$$

Taking into account the two vortices generated by a wing centered at position $(x, y) = (0, 0)$ as shown in Fig. 1, we obtain that the velocity generated at a position (x, y) is given by

$$\boxed{\begin{aligned} v(x,y) &= v_{\text{single}}(x - a/2, y) - v_{\text{single}}(x + a/2, y) \\ &= v_y(y)[v_x(x - a/2) - v_x(x + a/2)]. \end{aligned}} \tag{12}$$

The direction of this velocity is perpendicular to the (x, y)-plane as given by Fig. 1.

B. Upwash Model

The velocity field induced by a flying bird can be exploited by a trailing bird by locating itself in the regions where the velocity points in the upward direction. This upward velocity, also known as upwash, will help the trailing bird save energy required to maintain its position.

Consider a reference system as shown in Fig. 3. It is assumed that birds fly at a constant velocity V in the direction of negative y. The wingspan of a bird is denoted by b and is assumed constant for all birds. The leading bird is located at position (0, 0), and moves in the direction of negative y. The trailing bird is located at position (x, y) relative to the position of the leading bird, and will attempt to move to a location such that the upwash it experiences is maximized, as we now proceed to explain.

The average upwash experienced by a trailing bird of wingspan b, located at a position (x, y) with respect to the center of the horseshoe vortex of the leading bird is found by integrating the velocity field along the bird's wingspan b and dividing by b [10]. This upwash is given by

$$\begin{aligned} f_0(x,y) &= \frac{1}{b} \int_{x-b/2}^{x+b/2} v(\eta, y)\, d\eta \\ &= v_y(y) \frac{1}{b} \int_{x-b/2}^{x+b/2} [v_x(\eta - a/2) - v_x(\eta + a/2)]\, d\eta. \end{aligned}$$

We now define

$$f_x(x) \triangleq \frac{1}{b} \int_{x-b/2}^{x+b/2} v_x(\eta) d\eta \tag{13}$$

$$= \frac{\Gamma}{4\pi b} \cdot \ln \frac{(x + b/2)^2 + r_c^2}{(x - b/2)^2 + r_c^2} \tag{14}$$

and therefore[1]

$$f_0(x,y) = v_y(y)[f_x(x - a/2) - f_x(x + a/2)] \tag{15}$$

or, equivalently,

$$\boxed{\begin{aligned} f_0(x,y) = \tfrac{\Gamma}{4\pi b} \cdot \Big(&\ln \tfrac{(x-a/2+b/2)^2 + r_c^2}{(x-a/2-b/2)^2 + r_c^2} \\ &- \ln \tfrac{(x+a/2+b/2)^2 + r_c^2}{(x+a/2-b/2)^2 + r_c^2} \Big) \\ &\cdot \left(1 + \tfrac{y}{\sqrt{(b/2)^2 + y^2}} \right) e^{-\frac{(y-\beta)^2}{2\sigma}} \end{aligned}} \tag{16}$$

Fig. 4 shows the upwash generated by a flying bird located at position $(0, 0)$, and flying in the direction of negative y, using $r_c = 0.1$, $b = 1$, $a = \pi/4$, $\beta = 0.7$ and $\sigma = 4$. It is evident from the figure that the upwash is maximized at some locations $(\pm\Delta x_0^o, \Delta y_0^o)$, defined as

$$\{\Delta x_0^o, \Delta y_0^o\} = \arg \max_{\Delta x \geq 0, \Delta y} f_0(\Delta x, \Delta y). \tag{17}$$

Notice that we are defining Δx_0^o to be positive, and due to the even symmetry of $f_0(x, y)$ with respect to x, the optimal position of a trailing bird is given by either $(\Delta x_0^o, \Delta y_0^o)$ or $(-\Delta x_0^o, \Delta y_0^o)$. For the upwash shown in Fig. 4, these values are given by $\Delta x_0^o = 0.9097$ and $\Delta y_0^o = 1.0413$. Thus, for these parameters, a trailing bird should position itself about one wingspan away from its leading bird, both in the x and y directions.

IV. Diffusion Algorithms for V-Formations

In this section, we show how the diffusion LMS algorithm (Algorithm 1) can be employed to solve the problem of estimating the optimal position relative to the leading bird, such

[1]The terms $v_y(y) f_x(x - a/2)$ and $v_y(y) f_x(x + a/2)$ correspond to the right and left railing vortices, respectively, as shown in Fig. 1.

that the upwash is maximized. To do so, we linearize the upwash model (16) such that it is of the form (1).

A. Flock Configuration

Assume there are N birds, located at time i at positions $\{(x_{k,i}, y_{k,i})\}$, $k = 1, \ldots, N$, with respect to some global system of coordinates. Since all birds in the flock are moving in the direction of negative y, we define the leader of the flock at time i to be the bird with the smallest value of $y_{k,i}$, and denote this leader with index $k_{0,i}$. More precisely,

$$k_{0,i} = \arg\min_{k} y_{k,i}$$

Moreover, we assume that every bird in the flock, except for the flock leader, will select a reference bird according to some criterion, and will follow this reference bird, either to its left or to its right. The criterion to select a reference bird is based on the smallest weighted distance to every other bird in the flock, as discussed later in Section IV-D. For now, it suffices to say that a certain bird will select its reference among one of its closest birds. We say that a *left bird* will follow its reference bird to the left, while a *right bird* will follow its reference to its right (in the direction of movement). Thus, we define the quantity

$$p_k = \begin{cases} 1 & \text{if bird } k \text{ is a left bird} \\ -1 & \text{if bird } k \text{ is a right bird.} \end{cases} \tag{18}$$

We assume that for bird k, the reference bird at time i is positioned at $(x_{k,i}^{\mathrm{ref}}, y_{k,i}^{\mathrm{ref}})$. The leader of the flock at time i does not follow any other bird and will therefore not have a reference bird. Notice that we allow the leader of the flock and the references of every bird to change with time.

The neighbors of a bird k are defined as the two birds closest to bird k, together with bird k itself. The set of neighbors of node k (including itself) is called the neighborhood of bird k and is denoted by $\mathcal{N}_k$. Notice that with this definition, $l \in \mathcal{N}_k$ does not imply $k \in \mathcal{N}_l$, that is, k may select bird l to be one of its neighbors, while bird l need not select bird k as one of its neighbors. The reference bird can also be chosen as a neighbor.

B. Model Linearization

The upwash generated at a point (x, y) by a bird flying at position $(0, 0)$ in the direction of negative y is given by (15). Assume again that the N birds are located at some positions $\{(x_l, y_l)\}$ for $l = 1, \ldots, N$, with respect to some global system of coordinates, where we drop the dependence on time i to simplify the notation. Then, for an arbitrary bird k located at position (x, y), the upwash contribution due to each one of the remaining birds will depend on the relative coordinates $(x - x_l, y - y_l)$:

$$f(x, y) = \sum_{l=1, l\neq k}^{N} f_0(x - x_l, y - y_l) \tag{19}$$

where the upwash inflicted by the bird upon itself, $f_0(0, 0)$ is a constant which does not depend on the location of the bird and is therefore omitted from (19). Notice that as the birds move with time, the total upwash (19) will also change with time, and will in general be different for every bird.

Let $(x^{\mathrm{ref}}, y^{\mathrm{ref}})$ denote the reference bird of the bird under consideration (located at (x, y)), and let us center the coordinate system at $(x^{\mathrm{ref}}, y^{\mathrm{ref}})$. Relative to this new coordinate system, the (x, y) bird is now located at $(\Delta x, \Delta y)$, where $\Delta x = x - x^{\mathrm{ref}}$ and $\Delta y = y - y^{\mathrm{ref}}$. We also define $\Delta x_l = x_l - x^{\mathrm{ref}}$, $\Delta y_l = y_l - y^{\mathrm{ref}}$. Therefore, (19) in this new coordinate system becomes

$$\begin{aligned} f_c(\Delta x, \Delta y) &\triangleq f(\Delta x + x^{\mathrm{ref}}, \Delta y + y^{\mathrm{ref}}) \\ &= \sum_{l=1, l\neq k}^{N} f_0(\Delta x - \Delta x_l, \Delta y - \Delta y_l). \end{aligned} \tag{20}$$

Ideally, a bird would attempt to locate itself at a position $(\Delta x, \Delta y)$, relative to its reference bird, such that (20) is maximized. However, note that (20) will generally have multiple local maxima (for example, when birds are far away from each other, there will be two local maxima behind each bird). Therefore, a bird will attempt to find instead a *local* maximum of (20) in its proximity. Thus, a left bird will attempt to locate itself at a local maximum $(\Delta x^o, \Delta y^o)$, while a right bird will attempt to locate itself at a local maximum $(-\Delta x^o, \Delta y^o)$. Notice that we are defining Δx^o to be positive. In general, the bird will attempt to estimate the locally optimal position $(p\Delta x^o, \Delta y^o)$, where $p = 1$ if the bird is a left bird, and $p = -1$ otherwise.

Notice that the true upwash (20) will not be maximized at position $(\pm\Delta x^o, \Delta y^o)$ in general, since this is a local maximum in the proximity of the bird. Also, due to the decay of the upwash f_0 in the x and y directions shown in Fig. 4, we expect $(\Delta x^o, \Delta y^o)$ to be close to the maximizer of the function f_0, given by $(\Delta x_0^o, \Delta y_0^o)$ (see Section VI). In other words, it is reasonable to assume that the upwash at bird k will be dominated by the upwash generated by the bird leading bird k. Since all birds will attempt to find approximately the same estimate, the use of diffusion adaptation is justified.

We now explain how diffusion adaptation can be used to estimate the optimal displacement vector $(\Delta x^o, \Delta y^o)$. To do so, we first explain the data that are available to each bird and what kind of information they share. To begin with, let us assume that the bird located at position $(\Delta x, \Delta y)$ has some initial estimates of the values of Δx^o and Δy^o, and let us denote these estimates by $\widehat{\Delta x}$ and $\widehat{\Delta y}$, respectively. We define the error quantities

$$\begin{aligned} \widetilde{\Delta x} &\triangleq p(\Delta x^o - \widehat{\Delta x}) \\ \widetilde{\Delta y} &\triangleq \Delta y^o - \widehat{\Delta y}. \end{aligned}$$

For small $\widetilde{\Delta x}$ and $\widetilde{\Delta y}$, we can perform a first-order Taylor series expansion as follows:

$$\begin{aligned} f_c(\Delta x + \widetilde{\Delta x}, \Delta y + \widetilde{\Delta y}) &\approx f_c(\Delta x, \Delta y) \\ &+ \left.\frac{\partial f_c(x, y)}{\partial x}\right|_{(\Delta x, \Delta y)} \widetilde{\Delta x} + \left.\frac{\partial f_c(x, y)}{\partial y}\right|_{(\Delta x, \Delta y)} \widetilde{\Delta y}. \end{aligned}$$

Thus, if we define

$$\begin{aligned}\bar{f}_c(\Delta x,\Delta y,\widehat{\Delta x},\widehat{\Delta y}) \triangleq\ & f_c(\Delta x+\widetilde{\Delta x},\Delta y+\widetilde{\Delta y})\\ & - f_c(\Delta x,\Delta y)\\ & + p\frac{\partial f_c(x,y)}{\partial x}\bigg|_{(\Delta x,\Delta y)}\widehat{\Delta x}\\ & + \frac{\partial f_c(x,y)}{\partial y}\bigg|_{(\Delta x,\Delta y)}\widehat{\Delta y}\end{aligned}$$

we can rewrite (21) as

$$\begin{aligned}\bar{f}_c(\Delta x,\Delta y,\widehat{\Delta x},\widehat{\Delta y}) \approx\ & p\frac{\partial f_c(x,y)}{\partial x}\bigg|_{(\Delta x,\Delta y)}\Delta x^o\\ & + \frac{\partial f_c(x,y)}{\partial y}\bigg|_{(\Delta x,\Delta y)}\Delta y^o. \quad (21)\end{aligned}$$

Notice further that the term $f_c(\Delta x+\widetilde{\Delta x},\Delta y+\widetilde{\Delta y})$ is maximized (locally) at location $(\widehat{\Delta x},\widehat{\Delta y})$. Thus, since the bird will attempt to move to location $(\widehat{\Delta x},\widehat{\Delta y})$, we have that when $\Delta x = p\widehat{\Delta x}$ and $\Delta y = \widehat{\Delta y}$,

$$\begin{aligned}& f_c(\Delta x+\widetilde{\Delta x},\Delta y+\widetilde{\Delta y})\Big|_{\Delta x=p\widehat{\Delta x},\Delta y=\widehat{\Delta y}}\\ &\quad = f_c(p\Delta x^o,\Delta y^o)\\ &\quad \triangleq f_{\max}\end{aligned}$$

where $f_{\max}$ is the maximum local upwash. *A priori*, a bird will not have access to this upwash. However, it can set this value to the maximum upwash it has observed so far. That is, if $f_{k,i}$ denotes the upwash observed by bird k at time i, then we set[2]

$$f_{\max,k,i} \triangleq \max_{j=0,\ldots,i} f_{k,j}. \quad (22)$$

Expression (21) now allows us to write the data model at each bird k. First we collect the optimal displacement vector $(\Delta x^o,\Delta y^o)$ into an unknown vector w^o that we wish to estimate, as shown below. Then, for a generic bird k located at time i at position $(\Delta x_{k,i},\Delta y_{k,i})$ relative to its reference, we define

$$w^o = \begin{bmatrix}\Delta x^o & \Delta y^o\end{bmatrix}^T \quad (23)$$

$$\begin{aligned}u_{k,i} = & \left[p_k\frac{\partial f_c(x,y)}{\partial x}\bigg|_{(\Delta x_{k,i},\Delta y_{k,i})}\right.\\ & \left.\times\ \frac{\partial f_c(x,y)}{\partial y}\bigg|_{(\Delta x_{k,i},\Delta y_{k,i})}\right] \quad (24)\end{aligned}$$

$$\begin{aligned}d_k(i) = & f_{\max,k,i} - f_c(\Delta x,\Delta y)\\ & + p_k\frac{\partial f_c(x,y)}{\partial x}\bigg|_{(\Delta x_{k,i},\Delta y_{k,i})}\widehat{\Delta x}_{k,i}\\ & + \frac{\partial f_c(x,y)}{\partial y}\bigg|_{(\Delta x_{k,i},\Delta y_{k,i})}\widehat{\Delta y}_{k,i} + v_k(i) \quad (25)\end{aligned}$$

[2]An alternative approach is to add a forgetting factor to (22), obtaining an update of the form $f_{\max,k,i} = \max(\lambda f_{\max,k,i-1}, f_{k,i})$, where $0 < \lambda \leq 1$.

where $v_k(i)$ is a noise process that takes into account model and measurement uncertainties, assumed to be zero-mean of variance $\sigma_{v,k}^2$, $\Delta x_{k,i} = x_{k,i} - x_{k,i}^{\text{ref}}$ and $\Delta y_{k,i} = y_{k,i} - y_{k,i}^{\text{ref}}$. It is easy to verify that with these definitions, the approximations (21)–(22) have the same form as the observation model (1). Therefore, every bird k is assumed to have access to the data $\{d_k(i),u_{k,i}\}$ for every i. We then have a network of birds with access to data that satisfy model (1), with the unknown being the optimal displacement vector w^o relative to their respective leading birds. We can now use diffusion adaptation to estimate w^o. Notice that the flock leader does not have a reference, and therefore does not obtain measurements $\{d_k(i),u_{k,i}\}$. However, it will still be able to obtain estimates of w^o by communicating with its neighbors.

All that remains is to define the variables $\widehat{\Delta x}$ and $\widehat{\Delta y}$ for bird k at time i so that the quantities $\{d_k(i),u_{k,i}\}$ are readily available. From the definition of w^o, we can see that these variables can be obtained at time i from the first two entries of the previous estimate $w_{k,i-1}$, i.e.,

$$\widehat{\Delta x}_{k,i} = e_1^T w_{k,i-1} \quad \widehat{\Delta y}_{k,i} = e_2^T w_{k,i-1} \quad (26)$$

where e_m is a vector with a one at position m and zeros elsewhere.

The partial derivatives of $f_0(x,y)$ in (15) with respect to x and y are given by

$$\begin{aligned}&\frac{\partial f_0(x,y)}{\partial x}\\ &\quad = \frac{v_y(y)}{b}\left[v_x(x+b/2-a/2) - v_x(x-b/2-a/2)\right.\\ &\qquad \left. -v_x(x+b/2+a/2) + v_x(x-b/2+a/2)\right] \quad (27)\\ &\frac{\partial f_0(x,y)}{\partial y}\\ &\quad = \frac{\partial v_y(y)}{\partial y}\cdot\left[f_x(x-a/2) - f_x(x+a/2)\right] \quad (28)\end{aligned}$$

where

$$\begin{aligned}\frac{\partial v_y(y)}{\partial y} = e^{-\frac{(y-\beta)^2}{2\sigma}}\left[\frac{b^2/4}{(b^2/4+y^2)^{3/2}}\right.\\ \left. -\frac{y-\beta}{\sigma}\left(1+\frac{y}{\sqrt{b^2/4+y^2}}\right)\right].\end{aligned}$$

C. Interpretation

We now provide a physical interpretation for the quantities in (25) and (24). The vector $u_{k,i}$ contains the derivatives of the upwash experienced by the bird at its current location. It is the therefore assumed that the bird can "feel" these derivatives at its current location. The term $f_c(\Delta x,\Delta y)$ is the upwash felt by the bird at its current location, while the term quantity $f_{\max,k,i}$ is the maximum upwash the bird has experienced up to time i.

D. Motion Model

Consider two birds located at positions (x_1, y_1) and (x_2, y_2). We define the weighted distance between these two birds as follows:

$$d_\xi^2(x_1, y_1, x_2, y_2) = (x_1 - x_2)^2 + \xi(y_1 - y_2)^2 \tag{29}$$

with $\xi > 0$. The flock leader is denoted by $k_{0,i}$ and does not have a reference bird. For every bird k that is not the flock leader, we define a set of reference coordinates given by those of the closest leading bird, i.e.,

$$\left(x_{k,i}^{\text{ref}}, y_{k,i}^{\text{ref}}\right) = \arg \min_{x_{l,i}, y_{l,i}} d_\xi^2(x_{k,i}, y_{k,i}, x_{l,i}, y_{l,i}) \quad \text{subject to } y_{k,i} - y_{l,i} > 0. \tag{30}$$

Notice that d_ξ corresponds to a weighted Euclidean distance. For $\xi < 1$, more distance weight will be given in the x direction rather than in the y direction, and therefore a bird will tend to follow leaders in the y direction (since they are closer with respect to this weighted distance). For example, consider bird 1 located at $(0, 0)$, bird 2 located at $(-c, 0)$, and bird 3 located at $(0, -c)$, for some $c > 0$. If we use Euclidean distance to select the leading bird, both birds 2 and 3 would be at the same distance to node 1. In contrast, using the weighted distance we have that bird 2 is at a distance c from bird 1, while bird 3 is at a distance $\sqrt{\xi}c < c$ from bird 1. Therefore, bird 3 will be selected as the reference of bird 1. The rationale for using a weighted distance is to provide more flexibility to the flock formation. It is reasonable to assume that birds will have a tendency to follow birds directly in front of them rather than to their sides. This choice of distance will be useful later when we analyze the stability of the V-formation.

After a bird updates its estimate $w_{k,i}$ of the relative position with respect to its reference bird, it must move itself in the direction given by this estimate. Consider again bird k at time i, located at position $(x_{k,i}, y_{k,i})$, and having obtained a new estimate, $w_{k,i}$, of the optimal position. This bird will move to a new position $x_{k,i+1}, y_{k,i+1}$ based on the estimate $w_{k,i}$. Ideally, if the new estimate were perfect, a bird would move to the new location given by the coordinates $x_{k,i+1} = x_{k,i}^{\text{ref}} + e_1^T w_{k,i} p_k$ and $y_{k,i+1} = y_{k,i}^{\text{ref}} + e_2^T w_{k,i}$. However, since neither the new estimate is perfect, nor the bird can move too fast to a desired location, we use an update that combines in a convex manner the previous position and the desired position. Thus, the position update for a bird that is not a leader is given by

$$\begin{cases} x_{k,i+1} = \gamma x_{k,i} + (1-\gamma)(x_{k,i}^{\text{ref}} + e_1^T w_{k,i} p_k) + \nu_{k,i} \\ y_{k,i+1} = \gamma y_{k,i} + (1-\gamma)(y_{k,i}^{\text{ref}} + e_2^T w_{k,i}) - s_{k,i} + \zeta_{k,i} \end{cases} \tag{31}$$

where $0 < \gamma < 1$, $s_{k,i}$ is a deterministic positive quantity that determines the step taken by bird k at time i in the direction of negative y, and $\nu_{k,i}$ and $\zeta_{k,i}$ are noise processes that model the uncertainty in the movement of the bird. These processes are assumed zero-mean, independent in time and space, with variances σ_ν^2 and σ_η^2, respectively. For the leading bird of the flock, we set

$$\begin{cases} x_{k_0,i+1} = x_{k_0,i} + \nu_{k_0,i} \\ y_{k_0,i+1} = y_{k_0,i} - s_{k_0,i} + \zeta_{k_0,i} \end{cases}. \tag{32}$$

When all birds move in the direction of negative y with a velocity V, we set

$$s_{k,i} = V \cdot \Delta T \quad \text{for all } k \text{ and } i$$

where ΔT is the discrete time-step. Later we shall study formations where the step $s_{k,i}$ changes for every bird and every time instant.

E. Summary of Algorithm

The algorithm is as follows. At time i, bird k is located at position $(x_{k,i}, y_{k,i})$ and has an estimate $w_{k,i-1}$ of the optimal position with respect to its reference bird. The bird measures (or "feels") the upwash $f(x_{k,i}, y_{k,i})$ at its current location, together with its partial derivatives. It also has access to the maximum upwash it has experienced so far. In essence, bird k at time i has access to $d_k(i)$ and $u_{k,i}$, and can therefore use the diffusion LMS algorithm (Algorithm 1) to compute the new estimate of the best relative position, $w_{k,i}$. Notice that the bird will need to communicate its estimate with its neighbors in order to run the diffusion algorithm. Then, the bird will move to a new position $(x_{k,i+1}, y_{k,i+1})$ using (31), and the entire process is repeated in the next time instant. The complete algorithm is summarized below.

Algorithm 2: Self-Organization for V-Formations Using Diffusion LMS

Initialize $w_{k,-1}$, $x_{k,-1}$, $y_{k,-1}$ and p_k randomly (see remark below) for every bird $k = 1, \ldots, N$. At every time instant $i \geq 0$, do:

1) For every bird k that is not the flock leader, set a reference by finding the leading bird l closest to bird k using (30). Then set $x_{k,i}^{\text{ref}} = x_{l,i}$ and $y_{k,i}^{\text{ref}} = y_{l,i}$.
2) For every bird k that is not the flock leader, obtain $d_k(i)$ and $u_{k,i}$ using (25) and (24), where $\widehat{\Delta x}$ and $\widehat{\Delta y}$ are obtained from (26).
3) Use the diffusion LMS algorithm (Algorithm 1) to obtain $w_{k,i}$ for every bird k:
   a) Every bird k that is not the flock leader adapts its estimate using:
   $$\psi_{k,i} = w_{k,i-1} + \mu_k u_{k,i}^* \left(d_k(i) - u_{k,i} w_{k,i-1}\right)$$
   b) The flock leader does not update its estimate:
   $$\psi_{k_0,i} = w_{k_0,i-1}$$
   c) Every bird k sends $\psi_{k,i}$ to its neighbors and receives $\psi_{l,i}$ from its neighbors.
   d) Then bird k computes:
   $$w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{l,k} \psi_{l,i}$$
4) For every bird k, compute the new position using (31)–(32), where $s_{k,i} = V \cdot \Delta T$.

Remark: Initialization of the algorithm is important. If the birds are initially too far apart, or their initial estimates $w_{k,-1}$

are too large, they will not be able to feel the influence from other birds, and will not be able to form. Thus, it is assumed that the initial position is such that the birds are influenced by other birds. One choice is to draw the first element of $w_{k,-1}$ uniformly distributed between 0 and $2\Delta x_0^o$ and the second element of $w_{k,-1}$ uniformly distributed between 0 and $2\Delta y_0^o$. Although far less critical, the initial position of the birds should also be initialized in such a way that the formation can be obtained. One choice is to draw the positions uniformly distributed between $-Nb/4$ and $Nb/4$ in the horizontal direction, and between 0 and $Nb/2$ in the vertical direction. Finally, the coefficients p_k, which determine whether a bird is a left bird or a right bird are drawn randomly to be 1 or -1 with equal probability.

V. Performance Analysis

We now study the convergence properties of Algorithm 2, and provide a justification for convergence to a V-formation. The algorithm combines a rule for finding leading birds, a linearized observation model, the diffusion LMS algorithm, and a motion equation. Thus, the algorithm dynamics are complex and challenging to analyze exactly. However, the formation of V-shapes can be justified by resorting to some reasonable assumptions. In what follows we show that if we assume that the neighborhoods do not change with time, and that the diffusion LMS algorithm converges in steady-state to any one of the two local maxima behind its leading bird (which is reasonable for small step-sizes and provided birds do not get too close to each other), then the formation will converge to one with a uniform absolute separation between birds and their references. Subsequently, we show that formations other than V-formations are unstable. These results are in agreement with the simulation results of Section VI. When the initial conditions are such that birds are too close to each other, or their initial estimates of the optimal relative position are small, then the diffusion LMS algorithm may converge to a local minimum or some other local maximum which is not necessarily one of the two trailing maxima. In this situation, a V-formation will not be achieved. Thus, initialization of the algorithm in such a way that birds are not too close, and their estimates are not too small is important. These initial values would correspond to a practical limitation of birds not being able to get close to each other during flight without hitting each other.

A. Topology Convergence

We now analyze the limiting behavior of the motion equations, given by (31) and (32), and show that the expected value of the relative position of a bird with respect to its leader converges when the neighborhoods do not change with time. Thus, let

$$\boldsymbol{z}_{k,i} = \begin{bmatrix} \boldsymbol{x}_{k,i} \\ \boldsymbol{y}_{k,i} \end{bmatrix} \quad \boldsymbol{z}_{k,i}^{\mathrm{ref}} = \begin{bmatrix} \boldsymbol{x}_{k,i}^{\mathrm{ref}} \\ \boldsymbol{y}_{k,i}^{\mathrm{ref}} \end{bmatrix} \quad P_k = \begin{bmatrix} p_k & 0 \\ 0 & 1 \end{bmatrix}$$
$$b_{k,i} = \begin{bmatrix} 0 \\ -V \cdot \Delta T \end{bmatrix} \quad \eta_{k,i} = \begin{bmatrix} \nu_{k,i} \\ \zeta_{k,i} \end{bmatrix}$$

and, for the leading bird, we define

$$\boldsymbol{z}_{k_0,i}^{\mathrm{ref}} = \boldsymbol{z}_{k_0,i} - P_{k_0}\boldsymbol{w}_{k_0,i}. \tag{33}$$

We also introduce the extended quantities

$$\boldsymbol{z}_i = \mathrm{col}\{\boldsymbol{z}_{1,i}, \ldots, \boldsymbol{z}_{N,i}\} \tag{34}$$
$$\boldsymbol{z}_i^{\mathrm{ref}} = \mathrm{col}\left\{\boldsymbol{z}_{1,i}^{\mathrm{ref}}, \ldots, \boldsymbol{z}_{N,i}^{\mathrm{ref}}\right\} \tag{35}$$
$$\boldsymbol{\eta}_i = \mathrm{col}\{\boldsymbol{\eta}_{1,i}, \ldots, \boldsymbol{\eta}_{N,i}\}$$
$$\boldsymbol{w}_i = \mathrm{col}\{P_1\boldsymbol{w}_{1,i}, \ldots, P_N\boldsymbol{w}_{N,i}\} \tag{36}$$
$$b_i = \mathrm{col}\{b_{1,i}, \ldots, b_{N,i}\}. \tag{37}$$

The position update (31) for the entire flock can be written in the form

$$\boldsymbol{z}_{i+1} = \gamma\boldsymbol{z}_i + (1-\gamma)[\boldsymbol{z}_i^{\mathrm{ref}} + \boldsymbol{w}_i] + b_i + \boldsymbol{\eta}_i. \tag{38}$$

Notice that (38) also holds for the leading bird k_0. For the leading bird, we have

$$\boldsymbol{z}_{k_0,i+1} = \boldsymbol{z}_{k_0,i} + b_{k_0,i} + \boldsymbol{\eta}_{k_0,i}.$$

The reference vector can be written as

$$\boldsymbol{z}_i^{\mathrm{ref}} = \boldsymbol{H}_i\boldsymbol{z}_i + \left(\boldsymbol{e}_{k_0} \otimes \boldsymbol{z}_{k_0,i}^{\mathrm{ref}}\right)$$

where $\boldsymbol{H}_i = \bar{\boldsymbol{H}}_i \otimes I_2$, I_2 is the 2×2 identity matrix, and $\bar{\boldsymbol{H}}_i$ is $N \times N$ with entries 0 or 1 indicating which birds are references of other birds. For example, for the case $N = 3$, a matrix of the form

$$\bar{\boldsymbol{H}}_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

would indicate that at time i, bird 2 is the reference of bird 1, bird 3 is the reference of bird 2, and bird 3 is the leader.

Multiplying (38) by $\boldsymbol{H}_{i+1}$ and adding $(\boldsymbol{e}_{k_0} \otimes \boldsymbol{z}_{k_0,i+1}^{\mathrm{ref}})$, we obtain

$$\boldsymbol{z}_{i+1}^{\mathrm{ref}} = \gamma\boldsymbol{H}_{i+1}\boldsymbol{z}_i + (1-\gamma)\boldsymbol{H}_{i+1}\boldsymbol{z}_i^{\mathrm{ref}}$$
$$+(1-\gamma)\boldsymbol{H}_{i+1}\boldsymbol{w}_i + \boldsymbol{H}_{i+1}(b_i + \boldsymbol{\eta}_i) + \left(\boldsymbol{e}_{k_0} \otimes \boldsymbol{z}_{k_0,i+1}^{\mathrm{ref}}\right). \tag{39}$$

In order to proceed with the analysis, we will assume now that the neighborhoods do not change with time, and therefore $\boldsymbol{H}_i = H$ and $\boldsymbol{e}_{k_0} = e_{k_0}$. This assumption is reasonable once the flock has stabilized, and is supported by the simulation results of Section VI. Then, (39) can be rewritten as

$$\begin{aligned} \boldsymbol{z}_{i+1}^{\mathrm{ref}} &= \gamma H\boldsymbol{z}_i + (1-\gamma)H\boldsymbol{z}_i^{\mathrm{ref}} + (1-\gamma)H\boldsymbol{w}_i \\ &\quad + H(b_i + \boldsymbol{\eta}_i) + \left[e_{k_0} \otimes \left(\boldsymbol{z}_{k_0,i}^{\mathrm{ref}} + b_{k_0,i} + \boldsymbol{\eta}_{k_0,i}\right.\right. \\ &\quad \left.\left. + P_{k_0}\boldsymbol{w}_{k_0,i} - P_{k_0}\boldsymbol{w}_{k_0,i+1}\right)\right] \\ &= \gamma\boldsymbol{z}_i^{\mathrm{ref}} + (1-\gamma)\left[H\boldsymbol{z}_i^{\mathrm{ref}} + \left(e_{k_0} \otimes z_{k_0,i}^{\mathrm{ref}}\right)\right] \\ &\quad + (1-\gamma)H\boldsymbol{w}_i + b + H\boldsymbol{\eta}_i \\ &\quad + \left[e_{k_0} \otimes \left(\boldsymbol{\eta}_{k_0,i} + P_{k_0}\boldsymbol{w}_{k_0,i} - P_{k_0}\boldsymbol{w}_{k_0,i+1}\right)\right]. \end{aligned} \tag{40}$$

Subtracting (40) from (38), and defining

$$\tilde{\boldsymbol{z}}_i \triangleq \boldsymbol{z}_i - \boldsymbol{z}_i^{\mathrm{ref}}$$

we obtain

$$\tilde{\boldsymbol{z}}_{i+1} = [\gamma I + (1-\gamma)H]\tilde{\boldsymbol{z}}_i + (1-\gamma)[I - H]\boldsymbol{w}_i$$
$$+[I-H]\boldsymbol{\eta}_i - \left[e_{k_0} \otimes \left(\boldsymbol{\eta}_{k_0,i} + P_{k_0}\boldsymbol{w}_{k_0,i} - P_{k_0}\boldsymbol{w}_{k_0,i+1}\right)\right]. \tag{41}$$

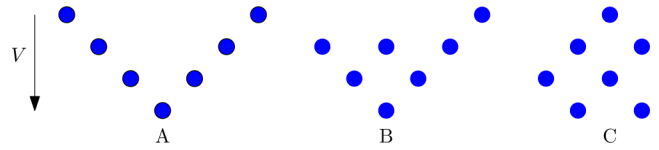

Fig. 5. Example topologies that satisfy (44).

Now, all eigenvalues of the matrix H are zero, and therefore the matrix $Q \triangleq \gamma I + (1-\gamma)H$ has eigenvalues γ. Since $0 < \gamma < 1$, the matrix Q is stable. Taking expectations of (41), we arrive at

$$\mathrm{E}\tilde{\boldsymbol{z}}_{i+1} = [\gamma I + (1-\gamma)H]\mathrm{E}\tilde{\boldsymbol{z}}_i + (1-\gamma)[I-H]\mathrm{E}\boldsymbol{w}_i. \quad (42)$$

In steady state, assuming the diffusion LMS algorithm converges in the mean-square sense, we have

$$\lim_{i\to\infty} \mathrm{E}\boldsymbol{w}_i = \begin{bmatrix} P_1 w^o \\ \vdots \\ P_N w^o \end{bmatrix}$$

and therefore

$$\lim_{i\to\infty} \mathrm{E}\tilde{\boldsymbol{z}}_i = (1-\gamma)[I-Q]^{-1}[I-H] \lim_{i\to\infty} \mathrm{E}\boldsymbol{w}_i. \quad (43)$$

Noticing that $[I-Q]^{-1} = (1-\gamma)^{-1}[I-H]^{-1}$, we conclude that

$$\lim_{i\to\infty} \mathrm{E}\tilde{\boldsymbol{z}}_i = \lim_{i\to\infty} \mathrm{E}\boldsymbol{w}_i. \quad (44)$$

Expression (44) indicates that, in steady state, we have

$$\left|\mathrm{E}\left(\boldsymbol{x}_{k,i} - \boldsymbol{x}_{k,i}^{\mathrm{ref}}\right)\right| = e_1^T w^o$$
$$\left|\mathrm{E}\left(\boldsymbol{y}_{k,i} - \boldsymbol{y}_{k,i}^{\mathrm{ref}}\right)\right| = e_2^T w^o$$

and shows that the expected relative distance of every bird in the flock to its reference is the same, and determined by the entries of w^o, assuming the neighborhoods do not change. Thus, once the flock has stabilized, it will stabilize into a formation that has this property. Topologies other than a V-shape have this property, and therefore, in the next section we will show that these topologies are unstable.

B. Unstable Topologies

Expression (44) indicates that in steady state, the horizontal and vertical distances between every node and its reference are constant across nodes. Fig. 5 shows some topologies that satisfy this property, including the V-formation.

In this section, we will show that the three formations shown in Fig. 6 are unstable. Consider first topology A, where node 3 is a left node following node 1, and node 2 is also assumed to be a left node. Without loss of generality, we assume that node 2 is also following node 1. Equation (44) shows that in steady-state, node 2 must be at a location relative to node 1 given by the entries of w^o. Thus, node 2 will attempt to position itself in the location of node 3. As soon as node 2 moves closer to node 3, the latter node will start following node 2, since it is now closer than node 1. Finally, since the topology does not change any longer, and all relative distances must be equal in steady state, the resulting topology is the one shown to the right

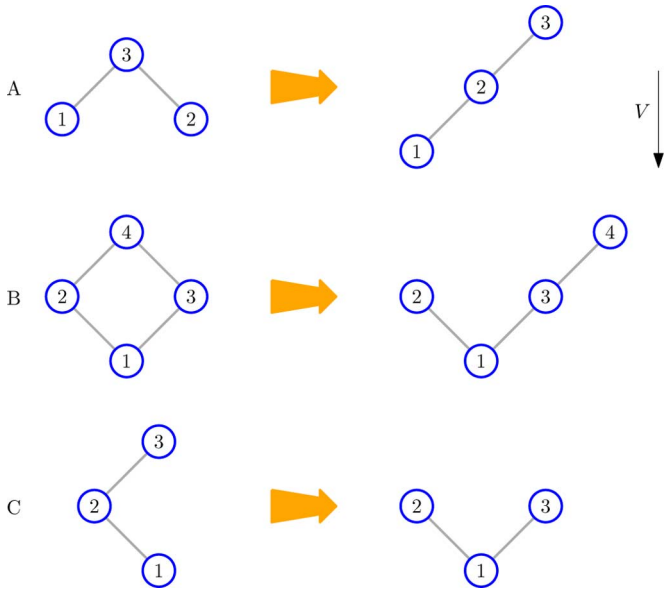

Fig. 6. Unstable topologies.

of topology A. This topology is a special case of a V-formation without right nodes.

Consider now topology B. Clearly, node 4 must either follow node 2 or node 3. If node 4 follows node 3, and is assumed to be a left node, it will move to the left of node 3 (in the flying direction), and the topology shown to the right of topology B is obtained. If node 4 is following node 2, it will, at some point in time, move closer to node 3 than to node 2 due to the noise component in the position of every node, given by the noisy estimates $\boldsymbol{w}_{k,i}$. Again, the topology shown to the right of topology B is obtained.

Finally, consider topology C. Using the weighted distance (29), node 3 will get closer to node 1 than to node 2 (in the sense of the weighted distance), and since node 3 is a left node, it will move to the left of node 1 (in the flying direction), as shown in the topology to the right of topology C. More formally, let the horizontal distance between nodes 2 and 3 be Δx_0^o plus noise, and let the vertical distance between node 1 and 2, and between node 2 and 3 be Δy_0^o plus noise. Then, the weighted distance between nodes 2 and 3 is given by

$$d_\xi^2(2,3) = (\Delta x_0^o)^2 + \xi(\Delta y_0^o)^2$$

while the weighted distance between nodes 3 and 1 is

$$d_\xi^2(1,3) = 4\xi(\Delta y_0^o)^2.$$

Thus, we will have $d_\xi^2(1,3) < d_\xi^2(2,3)$ if

$$4\xi(\Delta y_0^o)^2 < (\Delta x_0^o)^2 + \xi(\Delta y_0^o)^2$$

or, equivalently,

$$\xi < \frac{(\Delta x_0^o)^2}{3(\Delta y_0^o)^2.}$$

For the special case $\Delta x_0^o \approx \Delta y_0^o$ (which is the case in our simulations), we need $\xi < 1/3$. Notice that slightly higher values of ξ can be used, and the birds will still get into the V-formation.

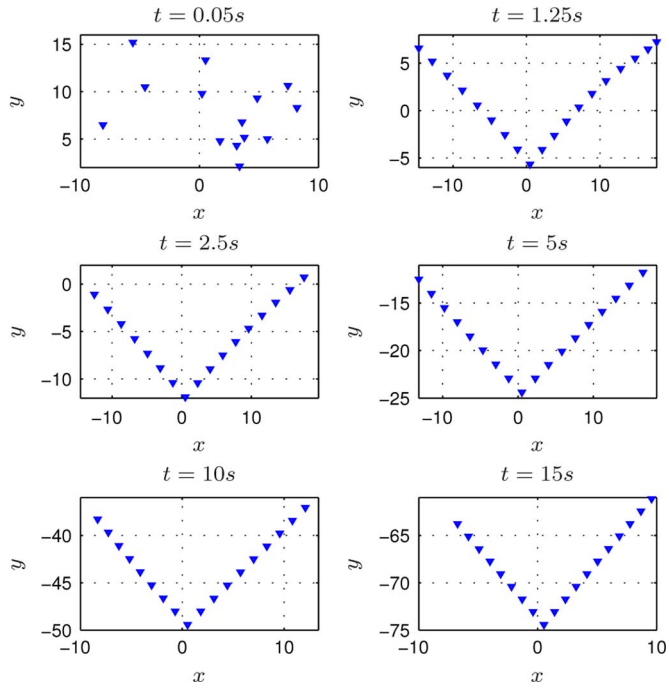

Fig. 7. Bird positions at different time instants.

The reason is that the noise introduced in the bird positions due to the noisy estimate of $\boldsymbol{w}_{k,i}$ will make node 3 get closer to node 1 with high probability for an infinite number of realizations.

VI. Simulation Results

We now present a simulation that illustrates the performance of the self-organizing algorithm. We use a total of $N = 19$ birds, with a wingspan $b = 1$, vortex separation $a = \pi/4$, vortex radius $r_c = 0.1$, $\beta = 0.7$ and $\sigma = 4$. The values of x and y that maximize the upwash due to a single bird are $\Delta x_0^o = 0.9097$ and $\Delta y_0^o = 1.0413$. For the motion, we use a flock velocity $V = 5$ m/s with sampling time $\Delta T = 0.05$ s, $\gamma = 0.5$, and the weighted distance coefficient is $\xi = 1/3$. For the diffusion LMS algorithm, the noise variance observed by every node is $\sigma_{v,k}^2 = 0.001$, and the step-size is $\mu_k = 0.002$ for all k, except for the flock leader which uses $\mu_k = 0$ (since it does not update its estimate). The noise variance in the position update is $\sigma_\eta^2 = \sigma_\zeta^2 = 10^{-6}$. The neighbors of a bird are defined as the two closest birds, plus itself. For the diffusion matrix, we use uniform weights

$$a_{l,k} = \begin{cases} 1/n_k & \text{if } l \in \mathcal{N}_k \\ 0 & \text{otherwise} \end{cases}$$

where n_k is the number of neighbors of node k (including itself). The initial positions $(x_{k,-1}, y_{k,-1})$ are drawn uniformly between $-Nb$ and Nb in the x direction, and between 0 and Nb in the y direction. The initial estimates $w_{k,-1}$ are drawn uniformly, with the first entry being between $-2b$ and $2b$ and the second entry being between 0 and $2b$.

Fig. 7 shows the resulting bird formations obtained through Algorithm 2 at different time instants throughout the simulation. It can be observed that after about 100 iterations ($t = 5$ seconds), the bird flock has converged to a V-shape. Fig. 8 shows the resulting total upwash generated by the 19 birds.

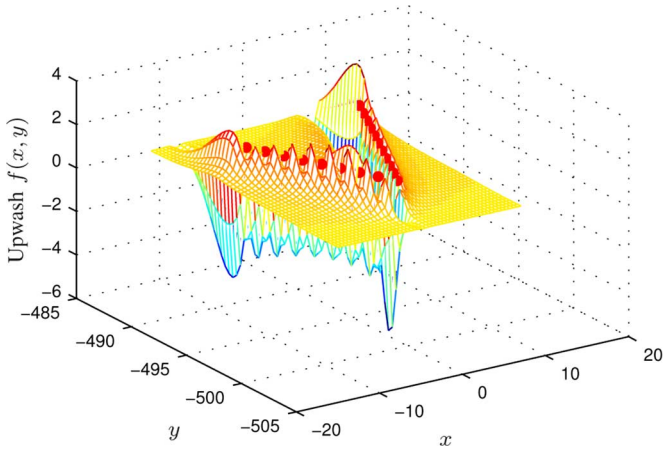

Fig. 8. Upwash generated by birds in steady-state.

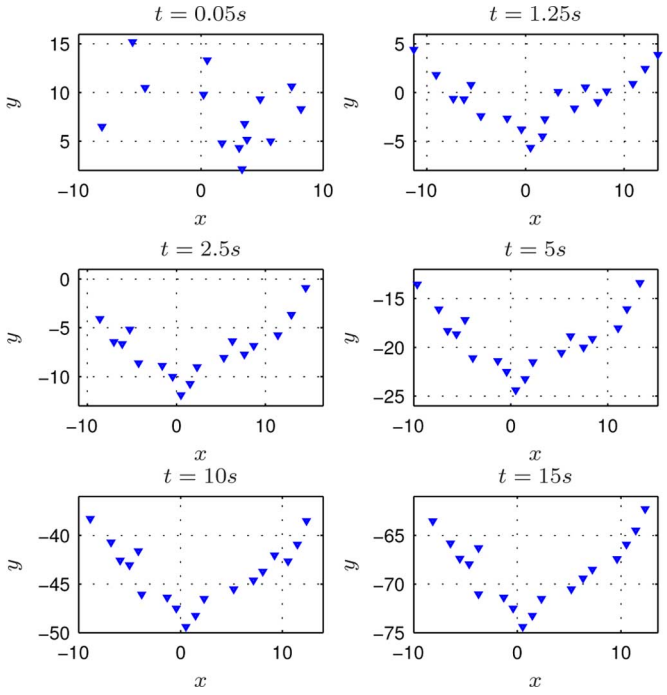

Fig. 9. Bird positions at different time instants, when there is no communication between birds.

We now study the case where nodes do not communicate with each other. We still perform an LMS iteration at every bird, but remove the diffusion step, or, equivalently, set

$$a_{l,k} = \begin{cases} 1, & \text{if } l = k \\ 0, & \text{otherwise.} \end{cases}$$

Thus, birds are only influenced by neighboring birds through the upwash, but do not communicate their estimates in any way. Fig. 9 shows the resulting formations for different time instants. Observe that now the birds do not organize into a V shape. This result would suggest that communication is critical to achieve V formations.

The performance of the diffusion LMS algorithm (Algorithm 1) was studied in detail in [14] for the case of linear observation models of the form (1). It was shown that Algorithm 1 will be asymptotically unbiased for sufficiently

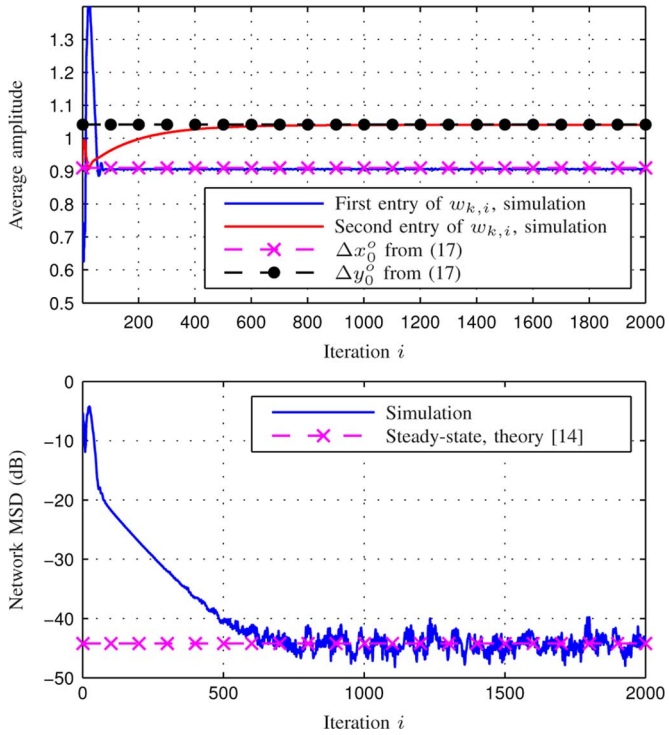

Fig. 10. Average value of the entries of $w_{k,i}$ (top) and the mean-square deviation (bottom).

small step-sizes μ. Moreover, expressions for the steady-state mean-square deviation (MSD) were also provided for the case of independent regressors [14, eq. (48)]. We now provide simulation results to show that these claims are also reasonable in the case of our linearized observation model. The top plot of Fig. 10 shows the values of the entries of $w_{k,i}$ as a function of the iteration, averaged over all nodes and over ten experiments. The dashed curves represent the entries of the vector $(\Delta x_0^o, \Delta y_0^o)$ that maximizes the individual upwash $f_0(x,y)$ due to a single bird. Notice that the estimates obtained by the birds are very close to these values, although there is a small bias due to the fact that the total upwash is not truly maximized at $(\Delta x_0^o, \Delta y_0^o)$ once all birds are taken into account.

The bottom plot of Fig. 10 shows the network instantaneous MSD, defined as

$$e(i) = \frac{1}{N}\sum_{k=1}^{N} \|w_k^{\rm ss} - w_{k,i}\|^2$$

where $w_k^{\rm ss}$ denotes the steady-state value of the estimate of bird k. Notice that we are using $w_k^{\rm ss}$ instead of w^o to compute the error because of the small bias of $w_{k,i}$. The dashed curves in Fig. 10 represent the theoretical values of the steady-state MSD obtained using [14, eq. (48)], averaged over all nodes and over ten experiments, where the regressor covariances $R_{u,k}$ were empirically estimated by averaging over realizations of the regressors $u_{k,i}$. Notice that the error is close to that predicted by the theoretical expressions.

VII. U-Shaped Formations

Formations other than V-shapes are usually observed in nature, including echelon formations, J-formations, inverted V formations, and U formations [40]. Echelon formations and J-formations are a special case of V-formations where the number of birds to one side of the leader is different than the number of birds to the other side of the leader (for echelon formations, there are no birds to one side of the flock leader). In this section, we show how we can modify our self-organization algorithm (Algorithm 2) in order to obtain U-shaped formations. These formations have some interesting properties as we now show.

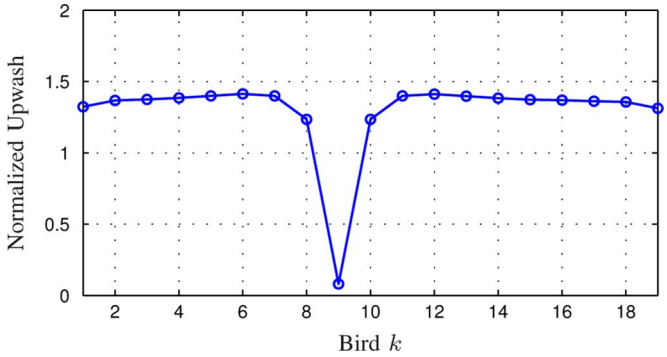

Fig. 11. Upwash observed by birds flying in V-formation.

It has been argued before that U-shaped formations help distribute the energy savings uniformly across birds [10], [41]. As we shall see, in V-formations, the leader bird experiences the least amount of upwash compared to other birds in the flock. This causes the leading bird to get more tired than the other birds, and start slowing down. This process of slowing down will tend to shape the V into a U.

Thus, we now introduce appropriate modifications to Algorithm 2 in order to model the effect of birds getting tired, and see how this model can predict U-shaped formations where the upwash is evenly distributed across all birds.

To begin with, we define the normalized upwash observed by a bird as follows:

$$f_{\rm norm}(x_{k,i}, y_{k,i}) = \frac{f(x_{k,i}, y_{k,i}) - f_{\rm solo}}{f_{0,\max}}$$

where $f(x_{k,i}, y_{k,i})$ is given by (19), $f_{\rm solo} = f_0(0,0)$ is the upwash (or downwash) generated by a bird flying alone upon itself, and $f_{0,\max}$ is the maximum upwash generated by a single flying bird, corresponding to the maximum value of $f_0(x,y)$. Fig. 11 shows the normalized upwash, $f_{\rm norm}(x_{k,i}, y_{k,i})$, in steady-state for the V-formation of Fig. 7, where nodes are numbered from left to right. It can be observed that the leading bird is not observing high energy savings due to the formation, and also that the birds next to the leader and on the tips of the V observe a lower upwash than the remaining birds. This type of curve has been pointed out before in [10].

When a bird observes a small upwash, it will spend more energy than the remaining birds, and it is therefore reasonable to assume that the bird will slow down. This can be reflected in the motion equation (31) by reducing the value of the step $s_{k,i}$. We propose the following update for the step, based on the upwash observed by a bird k and its neighbors

$$s_{k,i+1} = s_{k,i} + \nu\left[f(x_{k,i}, y_{k,i}) - f\left(x_{k,i}^{\rm ref}, y_{k,i}^{\rm ref}\right)\right], \quad \text{if } k \neq k_0 \qquad (45)$$

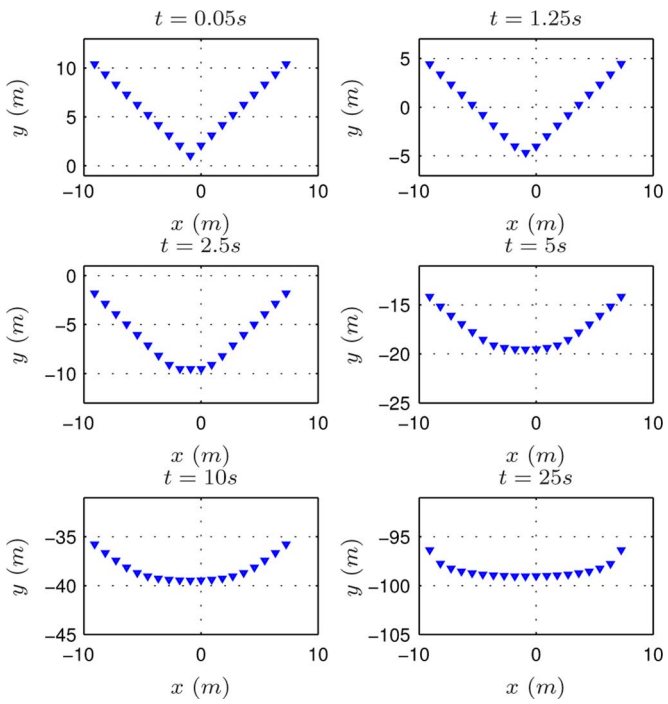

Fig. 12. Bird positions at different time instants, adaptive steps.

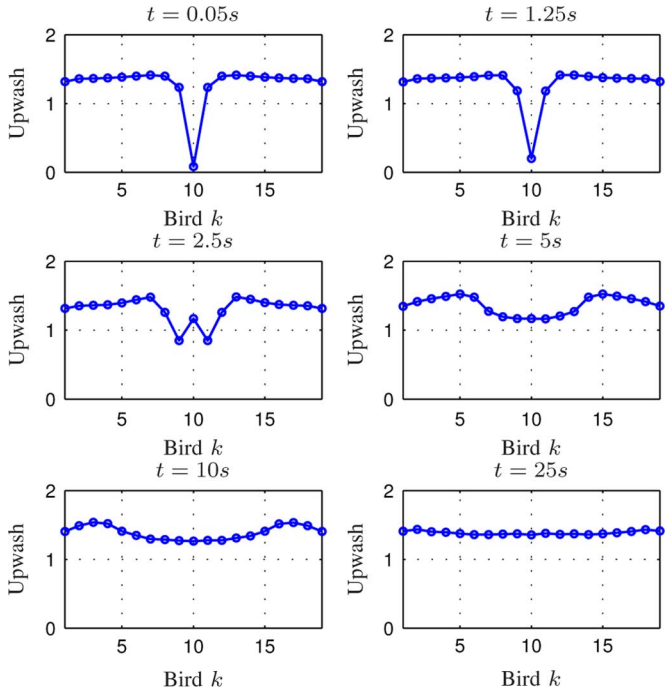

Fig. 13. Upwash observed by birds flying in U-formation.

$$s_{k_0,i+1} = s_{k_0,i} + \nu \left[f(x_{k_0,i}, y_{k_0,i}) - \frac{f(x_{k_1,i}, y_{k_1,i}) + f(x_{k_2,i}, y_{k_2,i})}{2} \right] \tag{46}$$

where ν is a positive step-size, and k_1 and k_2 are used to denote the two birds that follow the flock leader in a V or U formation. Thus, if the upwash observed by bird k at time i is less than the upwash observed by its reference bird, it will decrease the value of $s_{k,i}$, which will in turn make the bird fly farther away from its leader. If, on the contrary, the upwash of bird k is higher than that of its reference bird, it will increase $s_{k,i}$, moving closer to its reference.

The complete algorithm for obtaining U- formations is tabulated as Algorithm 3.

Algorithm 3: Self-organization for U-formations

Start from a V-formation and set $s_{k,-1} = V \cdot \Delta T$ for all k.
At every $i \geq 0$, repeat:

1) For every bird k, update $s_{k,i}$ using (45)–(46).
2) For every bird k, compute the new position using (31)–(32).

The resulting simulations, starting from a V-shape, are shown in Fig. 12, using $\nu = 0.0025$ and $\gamma = 0.9$. Notice how the V-shape deforms into a U after some iterations. In order to obtain Fig. 12, we added the constraint that a bird will never overtake its leader. This constraint allows us to avoid the effect of birds getting too close to their leader. Without this constraint, the formation will be unstable, with birds in close proximity trading places continuously.

Fig. 13 shows the resulting normalized upwash observed by every bird in the flock at different time instants, and for the formations of Fig. 12, where the birds are numbered from left to right. Notice how the upwash gets progressively equalized, and how after 500 iterations ($t = 25$ seconds), all birds in the flock observe nearly the same upwash.

Figs. 7–9 compare the algorithm with and without cooperation, for a single experiment. There exist different cases of interest that are not captured by these plots. First, it is important to study whether V-formations will still be achieved for different initial bird positions. Second, for the case without cooperation, it is of interest to see whether V-formations can be achieved for some initial configurations, and as we increase the number of iterations. A third case of interest is to vary the wingspans and vortex strengths of each bird, which is a more reasonable case in practice, and study whether V-formations can still be achieved. Fig. 14 addresses these three issues using simulations. In order to simplify the plots, we now use $N = 11$ birds. The plots show the bird formations after 2000 iterations ($t = 100$ s), for five independent experiments, both for the cases with and without cooperation. Each column corresponds to a different experiment, and both plots in each column (with and without cooperation) use the same initial flock configuration. Both the wingspan of each bird and the vortex strength generated by each bird are drawn uniformly between 0.5 and 1.5 for each experiment. The plots indicate that V-formations are still achievable in the case with cooperation, even with variations in the wingspan and strength. Moreover, V-formations are not attained without cooperation in most cases, which supports our claim that cooperation is necessary to achieve them. A final remark concerning the size of the neighborhoods is due. In our

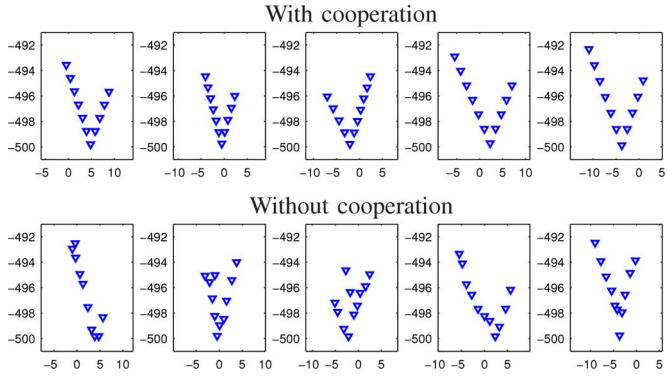

Fig. 14. Steady-state formations after 2000 iterations ($t = 100$ s) with and without cooperation. The initial formation is the same for both plots in each column.

simulations we used three neighbors for each bird (including itself). Our simulations indicate that increasing the number of neighbors tends to slightly increase the speed of convergence, but it does not affect the steady-state formation in a significant way.

VIII. Discussion and Conclusions

We presented an algorithm for self-organization in bird formations which uses the diffusion LMS algorithm of [13], [14], [17], and [24] to estimate the optimal position, relative to the closest leading bird. The estimation is performed using measurements from the upwash generated by neighboring birds, and by communicating the positioning estimates in the diffusion step.

Our results indicate that birds can form into a V-shape by running the proposed algorithm, which is fully distributed and runs in real time. The results also indicate that birds would be able to form into a V-shape based on upwash measurements and local communications. Finally, it appears that the information sharing between neighboring birds is necessary to achieve the V-shape.

We also showed that birds can form into a U-shape if, starting from a V-shape, the birds that observe lower upwash start slowing down. We proposed an algorithm that allows the birds to form into a U-formation, and showed that the upwash gets equalized across all birds in the flock.

Acknowledgment

The authors would like to acknowledge Ph.D. student X. Zhao for pointing out a result in connection with Fig. 11.

References

[1] F. Cattivelli and A. H. Sayed, "Self organization in bird flight formations using diffusion adaptation," in *Proc. Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (CAMSAP)*, Aruba, Dutch Antilles, Dec. 2009, pp. 49–52.

[2] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-Organization in Biological Systems*. Princeton, NJ: Princeton Univ. Press, 2003.

[3] M. Dorigo and T. Stutzle, *Ant Colony Optimization*. Cambridge: MIT Press, 2004.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Conf. Neural Netw.*, Piscataway, NJ, 1995, pp. 1942–1948.

[5] D. Merkle and M. Middendorf, "Swarm intelligence and signal processing," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 152–158, Nov. 2008.

[6] R. Mathar and J. Mattfeldt, "Pulse-coupled decentral synchronization," *SIAM J. App. Math.*, vol. 56, no. 4, pp. 1094–1106, Aug. 1996.

[7] Y. W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 5, pp. 1085–1099, May 2005.

[8] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, 2006.

[9] 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, 2004.

[10] D. Hummel, "Aerodynamic aspects of formation flight in birds," *J. Theor. Biol.*, vol. 104, pp. 321–347, 1983.

[11] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[12] A. H. Sayed, *Adaptive Filters*. New York: Wiley, 2008.

[13] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[14] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[15] V. H. Nascimento, M. Silva, R. Candido, and J. Arenas-Garcia, "A transient analysis for the convex combination of adaptive filters," in *Proc. IEEE Statist. Signal Process. Workshop*, Cardiff, Wales, Sep. 2009, pp. 53–56.

[16] J. Arenas-Garcia, A. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[17] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fund. Electron., Commun., Comput. Sci.*, vol. E90-A, no. 8, pp. 1504–1510, Aug. 2007.

[18] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS algorithms with information exchange," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 2008, pp. 251–255.

[19] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[20] L. Li and J. A. Chambers, "Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology," in *Proc. IEEE Statist. Signal Process. Workshop*, Cardiff, Wales, Sep. 2009, pp. 757–760.

[21] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 8, no. 9, pp. 4795–4810, Sep. 2010.

[22] S. Chouvardas, K. Slavakis, and S. Theodoridis, "A novel adaptive algorithm for diffusion networks using projections onto hyperslabs," in *Proc. IAPR Workshop Cognitive Inf. Process.*, Jun. 2010, pp. 393–398.

[23] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," presented at the Adapt. Sensor Array Process. Workshop, Lexington, MA, Jun. 2006.

[24] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Honolulu, HI, Apr. 2007, pp. 917–920.

[25] S. S. Stankovic, M. S. Stankovic, and D. S. Stipanovic, "Decentralized parameter estimation by consensus based stochastic approximation," in *Proc. 46th Conf Decision Control*, New Orleans, LA, Dec. 2007, pp. 1535–1540.

[26] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, vol. 8, no. 6, pp. 2365–2381, Jun. 2009.

[27] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Quantized data and random link failures," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1383–1400, 2010.
[28] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.
[29] L. M. Milne-Thomson, *Theoretical Aerodynamics*. New York: D. Van Nostrand, 1948.
[30] J. J. L. Higdon and S. Corrsin, "Induced drag of a bird flock," *Amer. Naturalist*, vol. 112, no. 986, pp. 727–744, Aug. 1978.
[31] N. J. Hallock, "Aircraft wake vortices: An assessment of the current situation," U.S. Dept. of Transportation, Rep. DOT-FAA-RD-90-29, 1991.
[32] L. Prandtl and O. G. Tietjens, *Fundamentals of Hydro and Aeromechanics*. New York: Dover, 1957.
[33] J. D. Anderson, *Fundamentals of Aerodynamics*. New York: McGraw Hill, 1984.
[34] W. J. M. Rankine, *A Manual of Applied Mechanics*. London, U.K.: Griffin, 1921.
[35] W. Blake and D. Multhopp, "Design, performance and modeling considerations for close formation flight," in *Proc. AIAA Guidance, Navigation and Control Conf.*, Philadelphia, PA, Aug. 1998, pp. 476–486.
[36] H. Lamb, *Hydrodynamics*. Cambridge, U.K.: Cambridge Univ. Press, 1932.
[37] M. J. Bhawat and J. G. Leishman, "Generalized viscous vortex model for application to free-vortex wake and aeroacoustic calculations," presented at the 58th Annu. Forum Technol. Displa Amer. Helicopter Soc. Int., Montreal, QC, Canada, Jun. 2002.
[38] P. Binetti, K. B. Ariyur, M. Krsti, and F. Bernelli, "Formation flight optimization using extremum seeking feedback," *J. Guidance, Control, Dynam.*, vol. 26, no. 1, pp. 132–142, Feb. 2003.
[39] G. C. Greene, "An approximate model of vortex decay in the atmosphere," *J. Aircraft*, vol. 23, pp. 566–573, 1986.
[40] F. H. Heppner, "Avian flight formations," *Bird-Banding*, vol. 45, no. 2, pp. 160–169, 1974.
[41] M. Andersson and J. Wallander, "Kin selection and reciprocity in flight formation?," *Behavior.Ecol.*, vol. 15, no. 1, pp. 158–162, 2004.

Federico S. Cattivelli (M'04) received the B.S. degree in electronics engineering from the Universidad ORT, Uruguay, in 2004 and the M.S. degree in 2006 and the Ph.D. degree in 2010 from the University of California, Los Angeles, both in electrical engineering.

In 2010, he joined Broadcom Corporation, where he develops and implements signal processing algorithms for communications. His research interests are in the areas of signal processing, adaptive filtering, estimation theory and optimization. Current and past research activities include distributed estimation and detection over adaptive networks, frequency estimation for space applications (in collaboration with the NASA JPL), and state estimation and modeling in biomedical systems (in collaboration with the UCLA School of Medicine).

Mr. Cattivelli was recipient of the UCLA Graduate Division and UCLA Henry Samueli SEAS Fellowships.

Ali H. Sayed (F'01) is Professor of Electrical Engineering at the University of California, Los Angeles (UCLA), where he directs the Adaptive Systems Laboratory (www.ee.ucla.edu/asl). He has published widely in the areas of adaptation and learning, statistical signal processing, estimation theory, adaptive and bio-inspired networks, and signal processing for communications. He has authored or coauthored several books, including *Adaptive Filters* (Wiley 2008), *Fundamentals of Adaptive Filtering* (Wiley 2003), and *Linear Estimation* (Prentice-Hall, 2000).

Dr. Sayed has served on the editorial boards of several journals, including as Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. His work has received several recognitions, including the 1996 IEEE Donald G. Fink Award, 2003 Kuwait Prize, 2005 Terman Award, and 2002 Best Paper Award, and 2005 Young Author Best Paper Award from the IEEE Signal Processing Society. He has served as a 2005 Distinguished Lecturer of the same society and as General Chairman of the International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2008. He is serving as the Vice-President of Publications of the IEEE Signal Processing Society.