Self-Organization in Bird Flight Formations Using Diffusion Adaptation

Federico Cattivelli Ali H. Sayed

Department of Electrical Engineering
University of California, Los Angeles, CA 90095
Emails: {fcattiv, sayed}@ee.ucla.edu

***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 simple 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 employ a distributed LMS algorithm, known as diffusion LMS. The algorithm requires the birds to obtain measurements of the upwash, and also to communicate with neighboring birds. The result has interesting implications. First, a simple diffusion LMS-based algorithm can account for the self-organization of birds. The algorithm is fully distributed and runs in real time. Second, that birds can self-organize based on the air pressures generated by the other birds. Third, that some form of communication among birds is crucial to achieve the flight formation.**

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

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 [1]. 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 [2] is based on how ants organize in order to find the shortest path to food, and Particle Swarm Optimization [3] is based on how birds flock to find food. Both algorithms have been applied to solve different optimization problems [4]. Algorithms based on how fireflies synchronize have been proposed for wireless network synchronization [5], [6].

In this work we focus on bird flocks, and specifically on V-shaped formations obtained during bird flight. It has been argued before [7] that birds form into V-shapes in order to save energy. The reason, although yet debated, is that a flying bird generates an upward pressure known as *upwash*, which a trailing bird can use to maintain its altitude and save energy. Still, what type of algorithm is employed by the birds to get into this formation is unknown.

In order to obtain these formations, we employ a distributed estimation algorithm over cognitive, adaptive networks, which is based on the popular LMS algorithm of adaptive filtering [8], [9]. Distributed estimation algorithms are based on the principle that its nodes should obtain some estimate by communicating only with their neighbors. A class of distributed estimation algorithms is known as diffusion algorithms, whereby nodes perform an adaptation step using the available measurements, followed by a diffusion step which requires combining the estimates from the neighboring nodes. Diffusion algorithms based on LMS and RLS have been proposed before [10]-[13].

This material was based on work supported in part by the National Science Foundation under awards ECS-0601266 and ECS-0725441.

In this work we focus on the more general diffusion LMS algorithm of [12]. We show by simulation that if birds were to employ this algorithm to attempt to estimate the best position relative to a reference bird, they would end up in a V-formation. We start by introducing the diffusion LMS algorithm.

A. The diffusion LMS algorithm

The diffusion LMS algorithm [10]-[12] 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 Adapt-Then-Combine (ATC) version of the diffusion LMS algorithm [12], though we will refer to it simply as diffusion LMS.

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.

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 non-negative real entries $a_{l,k}$ satisfying:

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

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.

***ATC Diffusion LMS (no measurement exchange)* [12]**

At every time $i \geq 0$, repeat for every node k:

$$\begin{cases} \psi_{k,i} &= w_{k,i-1} + \mu_k u_{k,i}^* [d_k(i) - u_{k,i} w_{k,i-1}] \\ w_{k,i} &= \displaystyle\sum_{l \in \mathcal{N}_k} a_{l,k} \psi_{l,i} \end{cases} \tag{2}$$

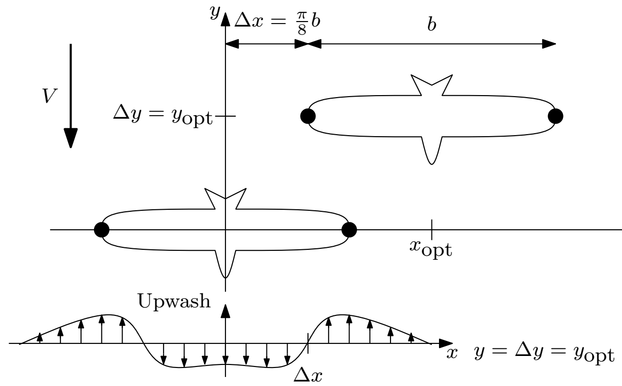

Fig. 1. Bird reference system with respect to leading bird.

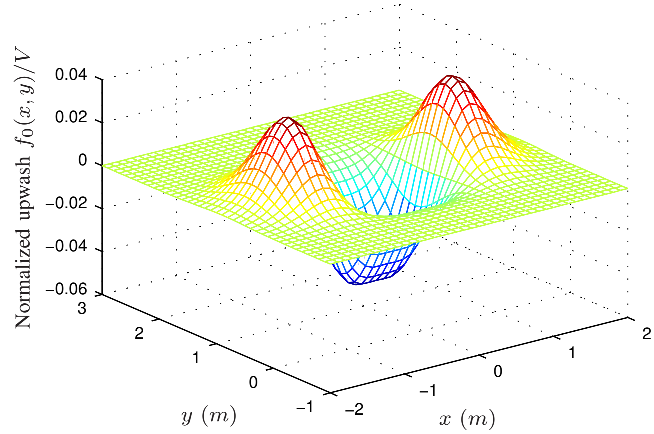

Fig. 2. Upwash generated by a bird located at position $x = 0$, $y = 0$, using $\sigma_x = 0.5$, $\sigma_y = 0.6$, $b = 1.8$, $\Delta x = 0.49b - \sigma_x$ and $\Delta y = 0.9$.

II. Methodology

A. Upwash model

Consider a reference system as shown in Figure 1. 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. With respect to the trailing bird, the leading bird is located at position $(0, 0)$, and moves in the direction of negative y. The trailing bird will locate itself at position $(\pm x_{\text{opt}}, y_{\text{opt}})$ relative to the position of the leading bird. The optimal bird position coordinates, x_{opt} and y_{opt}, will depend on the aerodynamic model employed.

It is well known that the flight of a bird generates vortices, which in turn produce a field of of induced velocities in the proximity of the wings [7], [14], [15]. Upward velocity, also known as *upwash*, is generated near the wingtips of the bird, while downward velocity or downwash is generated near the center of the bird. Near the wingtips, a pair of free vortices form, and it has been shown before that the horizontal distance between the centers of these vortices is $\pi b/4$, where b is the wingspan [7]. Moreover, it has been argued that the optimal position of the bird is therefore $x_{\text{opt}} = \pm(1/2 + \pi/8)b$. In this work we use a simple model for the upwash generated by a bird. Assuming the bird is located at position $x = 0$, $y = 0$, and moving in the vertical direction (towards negative y), then the upwash generated by the bird is given by:

$$\begin{aligned} f_0(x, y, \Delta x, \Delta y) = \quad & \alpha V(x - \Delta x) \cdot g\left(\tfrac{x-\Delta x}{\sigma_x}\right) \cdot g\left(\tfrac{y-\Delta y}{\sigma_y}\right) \\ & -\alpha V(x + \Delta x) \cdot g\left(\tfrac{x+\Delta x}{\sigma_x}\right) \cdot g\left(\tfrac{y-\Delta y}{\sigma_y}\right) \quad (3) \end{aligned}$$

where α is some constant and $g(\cdot)$ is the Gaussian function:

$$g(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

Figure 2 illustrates the upwash generated by a bird located at position $x = 0$, $y = 0$, using the parameters $\alpha = 1$, $\sigma_x = 0.4$, $\sigma_y = 0.5$, $b = 1.8$, $\Delta x = \pi b/8$ and $\Delta y = 0.9$. All positions have units of meters (m), the upwash is in m/s and α is in m^{-1}.

A bird flying on the rear of another bird will choose to position one of its wingtips at the center of one of the vortices generated by the leading bird. Thus, if the leading bird is in location $(0, 0)$, the optimal location for the trailing bird will be

$$x_{\text{opt}} = \pm(\Delta x + b/2) \qquad y_{\text{opt}} = \Delta y \quad (4)$$

Using (4), we can express (3) as a function of x_{opt} and y_{opt}, by replacing:

$$\Delta x = |x_{\text{opt}}| - b/2 \qquad \Delta y = y_{\text{opt}} \quad (5)$$

We will denote by $f_0(x, y, x_{\text{opt}}, y_{\text{opt}})$ the function obtained from $f_0(x, y, \Delta x, \Delta y)$ when using the above replacements. Consider now the case where N birds are present, and let $(x_{k,i}, y_{k,i})$ denote the coordinates of bird k at time i with respect to some arbitrary frame of reference. The overall upwash observed by bird k is given by:

$$f(x_{k,i}, y_{k,i}, x_{\text{opt}}, y_{\text{opt}}) = \sum_{l=1}^{N} f_0(x_{k,i} - x_{l,i}, y_{k,i} - y_{l,i}, x_{\text{opt}}, y_{\text{opt}}) \quad (6)$$

B. Model Linearization

Assume that a bird located at position (x, y) has an estimate of the optimal position $(x_{\text{opt}}, y_{\text{opt}})$, and let us denote this estimate by $(\hat{x}_{\text{opt}}, \hat{y}_{\text{opt}})$. To first order, we can approximate

$$\begin{aligned} f(x, y, x_{\text{opt}}, y_{\text{opt}}) \quad \approx \quad & f(x, y, \hat{x}_{\text{opt}}, \hat{y}_{\text{opt}}) + \left.\frac{\partial f}{\partial x_{\text{opt}}}\right|_{x_{\text{opt}}=\hat{x}_{\text{opt}}} (x_{\text{opt}} - \hat{x}_{\text{opt}}) + \\ & \left.\frac{\partial f}{\partial y_{\text{opt}}}\right|_{y_{\text{opt}}=\hat{y}_{\text{opt}}} (y_{\text{opt}} - \hat{y}_{\text{opt}}) \quad (7) \end{aligned}$$

The partial derivatives of f in (6) with respect to x_{opt} and y_{opt} after replacing (5) are given by

$$\begin{aligned} \frac{\partial f(x, y, x_{\text{opt}}, y_{\text{opt}})}{\partial x_{\text{opt}}} &= \sum_{l=1}^{N} \frac{\partial f_0(x - x_{l,i}, y - y_{l,i}, x_{\text{opt}}, y_{\text{opt}})}{\partial x_{\text{opt}}} \\ \frac{\partial f(x, y, x_{\text{opt}}, y_{\text{opt}})}{\partial y_{\text{opt}}} &= \sum_{l=1}^{N} \frac{\partial f_0(x - x_{l,i}, y - y_{l,i}, x_{\text{opt}}, y_{\text{opt}})}{\partial y_{\text{opt}}} \end{aligned}$$

And the partial derivatives of f_0 are given by

$$\begin{aligned} \frac{\partial f_0(x, y, x_{\text{opt}}, y_{\text{opt}})}{\partial x_{\text{opt}}} = -\alpha V \,\text{sign}(x_{\text{opt}}) g\left(\frac{y - y_{\text{opt}}}{\sigma_y}\right) \cdot \Bigg[\\ \frac{x - \Delta x}{\sigma_x} \cdot g'\left(\frac{x - \Delta x}{\sigma_x}\right) + g\left(\frac{x - \Delta x}{\sigma_x}\right) + \\ \frac{x + \Delta x}{\sigma_x} \cdot g'\left(\frac{x + \Delta x}{\sigma_x}\right) + g\left(\frac{x + \Delta x}{\sigma_x}\right) \Bigg] \quad (8) \end{aligned}$$

$$\begin{aligned} \frac{\partial f_0(x, y, x_{\text{opt}}, y_{\text{opt}})}{\partial y_{\text{opt}}} = -\frac{\alpha V}{\sigma_y} g'\left(\frac{y - y_{\text{opt}}}{\sigma_y}\right) \cdot \Bigg[\\ (x - \Delta x) g\left(\frac{x - \Delta x}{\sigma_x}\right) - (x + \Delta x) g\left(\frac{x + \Delta x}{\sigma_x}\right) \Bigg] \quad (9) \end{aligned}$$

Thus, for bird k, located at position $(x_{k,i}, y_{k,i})$, we define

$$\begin{aligned} w^o &= [x_{\text{opt}} \quad y_{\text{opt}}]^T \\ u_{k,i} &= \left[\left.\frac{\partial f}{\partial x_{\text{opt}}}\right|_{x_{\text{opt}}=\hat{x}_{\text{opt}}} \quad \left.\frac{\partial f}{\partial y_{\text{opt}}}\right|_{y_{\text{opt}}=\hat{y}_{\text{opt}}} \right] \qquad (10) \\ d_k(i) &= f(x_{k,i}, y_{k,i}, x_{\text{opt}}, y_{\text{opt}}) - f(x_{k,i}, y_{k,i}, \hat{x}_{\text{opt}}, \hat{y}_{\text{opt}}) + \\ &\quad \left.\frac{\partial f}{\partial x_{\text{opt}}}\right|_{x_{\text{opt}}=\hat{x}_{\text{opt}}} \hat{x}_{\text{opt}} + \left.\frac{\partial f}{\partial y_{\text{opt}}}\right|_{y_{\text{opt}}=\hat{y}_{\text{opt}}} \hat{y}_{\text{opt}} \qquad (11) \end{aligned}$$

It is easy to verify that with these definitions, the approximation (7) has the same form as the observation model (1). All that remains is to define the variables $\hat{x}_{\text{opt}}$ and $\hat{y}_{\text{opt}}$ for bird k at time i. 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.,

$$\hat{x}_{\text{opt}} = e_1^T w_{k,i-1} \qquad \hat{y}_{\text{opt}} = e_2^T w_{k,i-1} \qquad (12)$$

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

C. Motion model

We assume that every bird positions itself with respect to a reference bird, which is given by its closest leading bird. For every bird k, we define a set of reference coordinates given by those of the closest leading bird, i.e.,

$$(x_{k,i}^{\text{ref}}, y_{k,i}^{\text{ref}}) = \begin{cases} (x_{l,i}, y_{l,i}) & \text{if } l \text{ is the closest leading bird of } k \\ (x_{0,i}, y_{0,i}) & \text{if } k \text{ has no leading bird} \end{cases}$$

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 $x_{k,i+1} = x_{k,i}^{\text{ref}} + e_1^T w_{k,i}$ 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 is given by

$$\begin{aligned} x_{k,i+1} &= \gamma x_{k,i} + (1-\gamma)(x_{k,i}^{\text{ref}} + e_1^T w_{k,i}) + \nu_{k,i} \\ y_{k,i+1} &= \gamma y_{k,i} + (1-\gamma)(y_{k,i}^{\text{ref}} + e_2^T w_{k,i}) + \zeta_{k,i} - V \cdot \Delta T \end{aligned} \qquad (13)$$

where $0 \leq \gamma \leq 1$, and $\nu_{k,i}$ and $\zeta_{k,i}$ are zero-mean random processes, independent in time and space, with variances σ_ν^2 and σ_ζ^2, respectively. These noise processes add randomness to the bird movements, and account for unmodeled natural factors such as wind, bird exhaustion, etc. In order to account for the movement of the entire flock at velocity V in the direction of negative y, we also subtract $V \cdot \Delta T$ from the vertical movement equation in (13), where ΔT is the discrete time-step, and set $x_{0,i+1} = x_{0,i}$ and $y_{0,i+1} = y_{0,i} - V \cdot \Delta T$.

D. 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 $f(x_{k,i}, y_{k,i}, x_{\text{opt}}, y_{opt})$ at its current location. It also has access to the upwash with respect to its estimated best relative position, $(\hat{x}_{\text{opt}}, \hat{y}_{\text{opt}})$ (this would correspond to what they would expect the upwash to be at their current location), and the partial derivatives of this upwash. 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 (2) 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 (13), and the entire process is repeated in the next time instant. The complete algorithm is summarized below.

Self-organizing algorithm using diffusion LMS

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

1) For every bird k, set a reference by finding the leading bird l closest to bird k. Then set $x_{k,i}^{\text{ref}} = x_{l,i}$ and $y_{k,i}^{\text{ref}} = y_{l,i}$. If a bird has no leading bird, set $x_{k,i}^{\text{ref}} = x_{0,i}$ and $y_{k,i}^{\text{ref}} = y_{0,i}$.
2) For every bird k, obtain $d_k(i)$ and $u_{k,i}$ using (11) and (10), where $\hat{x}_{\text{opt}}$ and $\hat{y}_{\text{opt}}$ are obtained from (12).
3) Use the diffusion LMS algorithm (2) to obtain $w_{k,i}$ for every bird k. Notice that communication with neighboring birds is required in this stage.
4) For every bird k, compute the new position using (13).
5) Set $x_{0,i+1} = x_{0,i}$ and $y_{0,i+1} = y_{0,i} - V \cdot \Delta T$.

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 $-b$ and b and the second element of $w_{k,-1}$ uniformly distributed between 0 and $2y_{\text{opt}}$. Though 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.

III. 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 of $b = 1.8$, $\alpha = 1$, $\Delta y = y_{\text{opt}} = 0.9$, $\Delta x = \pi b/8$, $x_{\text{opt}} = \Delta x + b/2$ and $\sigma_x = \sigma_y = 0.4$. For the motion, we use $V = 5$ m/s, $\Delta T = 0.05$ s, $\gamma = 0.9$, and the motion noise processes are zero-mean Gaussian with deviations $\sigma_\nu = \sigma_\zeta = 0.05$. The noise variance observed by every node is $\sigma_{v,k} = 10^{-4}$, and the step-size of the diffusion LMS algorithm is $\mu_k = 10$ for all k. The diffusion matrix for diffusion LMS uses uniform weights (all neighbors have the same weight, normalized to add up to one), and the neighbors of a bird are defined as the two closest birds, plus itself. The initial positions are drawn uniformly between $-bN/4$ and $bN/4$ in the x direction, and between 0 and $bN/2$ in the y direction. The initial estimates are drawn uniformly, with the first entry being between $-b$ and b and the second entry being between 0 and b.

Figure 3 shows the resulting bird formations at different time instants throughout the simulation. It can be observed that after about 500 iterations, the bird flock has converged to a V-shape. Notice that the algorithm is able to resolve the case where birds get "trapped" inside the V, as can be observed at time instant $t = 4s$. Figure 4 shows the resulting upwash generated by the 19 birds. The red dots in the plot indicate the positions of the birds. Notice how every bird flies in such a way that its generated upwash overlaps with the upwash from its leading bird.

Finally, we 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 $A = I$ at all times. Thus, birds are only influenced by neighboring birds through the

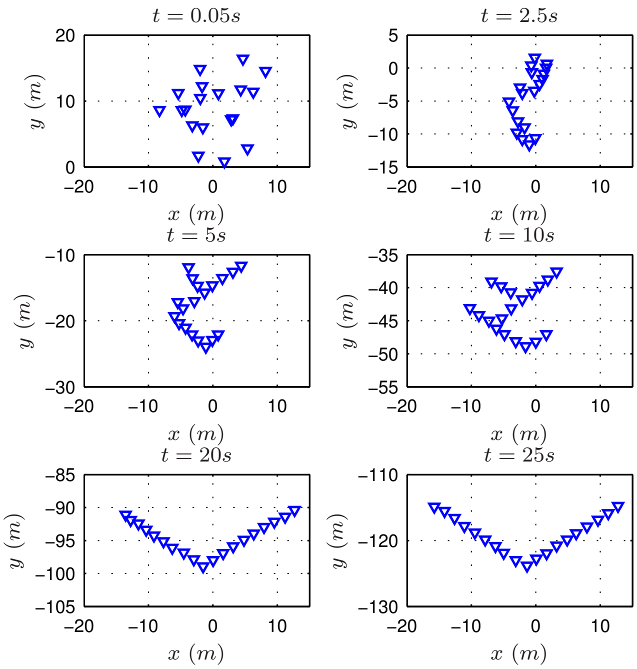

Fig. 3. Bird positions at different time instants.

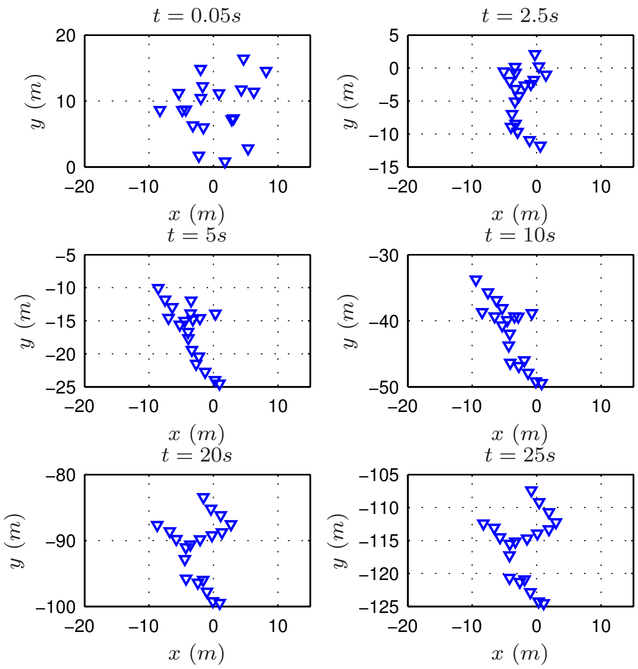

Fig. 5. Bird positions at different time instants, when there is no communication between birds.

upwash, but do not communicate their estimates in any way. Figure 5 shows the resulting formations for different time instants. Note that now the birds do not organize into a V shape, and struggle even to organize into any similar shape. This would suggest that communication is critical to achieve V formations.

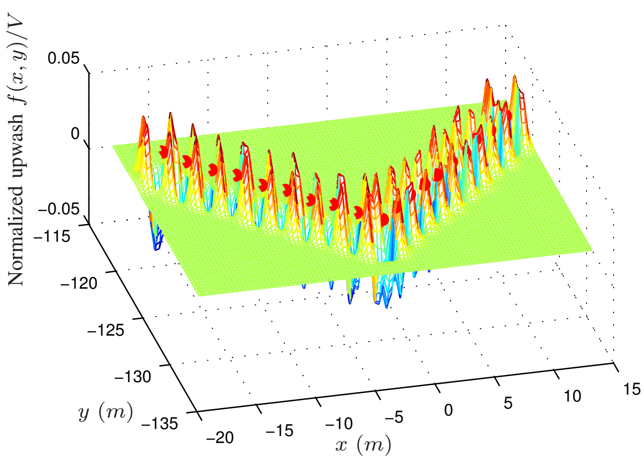

Fig. 4. Upwash generated by birds in steady-state.

IV. Discussion and Conclusions

We presented an algorithm for self-organization in bird formations which uses the diffusion LMS algorithm 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 diffusion step, which requires communication between neighboring birds, is critical to achieve the V-shape.

References

[1] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-organization in Biological Systems*. Princeton University Press, 2003.
[2] M. Dorigo and T. Stutzle, *Ant Colony Optimization*. MIT Press, 2004.
[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Conf. on Neural Networks*, Piscataway, NJ, 1995, pp. 1942–1948.
[4] D. Merkle and M. Middendorf, "Swarm intelligence and signal processing," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 152–158, November 2008.
[5] R. Mathar and J. Mattfeldt, "Pulse-coupled decentral synchronization," *SIAM J. App. Mathematics*, vol. 56, no. 4, pp. 1094–1106, August 1996.
[6] Y. W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE J Selected Areas in Communications*, vol. 23, no. 5, pp. 1085–1099, May 2005.
[7] D. Hummel, "Aerodynamic aspects of formation flight in birds," *J. Theor. Biol.*, vol. 104, pp. 321–347, 1983.
[8] A. H. Sayed, *Fundamentals of Adaptive Filtering*. NJ: Wiley, 2003.
[9] ——, *Adaptive Filters*. NJ: Wiley, 2008.
[10] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. on Fund. of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 8, pp. 1504–1510, August 2007.
[11] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.
[12] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS algorithms with information exchange," in *Proc. Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, October 2008, pp. 251–255.
[13] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.
[14] M. Andersson and J. Wallander, "Kin selection and reciprocity in flight formation?" *Behavioral Ecology*, vol. 15, no. 1, pp. 158–162, 2004.
[15] P. Seiler, P. Aniruddha, and K. Hedrick, "Analysis of bird formations," in *Proc. IEEE Conf. Decision and Control*, Las Vegas, NV, 2002, pp. 3289–3292.