Mobile Adaptive Networks with Self-Organization Abilities

Sheng-Yuan Tu and Ali H. Sayed
Department of Electrical Engineering
University of California, Los Angeles, CA 90095
{shinetu, sayed}@ee.ucla.edu

***Abstract*—In this paper we investigate the self-organization and cognitive abilities of adaptive networks when the individual agents are allowed to move in pursuit of an objective. The network as a whole acts as an adaptive entity with localized processing and is able to respond to stimuli in real-time. We apply the ensuing model to the foraging behavior of fish schools in search of food sources and reproduce their ability to move in remarkable coherence.**

***Index Terms*—Adaptive networks, mobility, self-organization, diffusion adaptation.**

I. Introduction

Self-organization is observed in several physical and biological phenomena. Examples include fish joining together in schools, birds flying in V-formation [9], and bees swarming towards a new hive [1]. In these cases, a global pattern of behavior emerges from localized interactions among the individual components of the system.

One interesting behavior of animal groups is collective motion [1][2], where animals move together in amazing coherence and synchrony. There have been extensive prior studies in the literature on the collective motion of animals. Previous analyses [6][7][10] have been successful in emulating the harmonious motion of animal groups by assuming that individual agents move along the average direction of their neighbors and use repulsion and attraction mechanisms to maintain a safe distance from the neighbors. While these models help emulate the coordinated motion behavior of animals, they nevertheless do not address the combined problem of how agents can move in synchrony while at the same time attempting to solve an estimation problem of interest, such as tracking a target or moving towards a food source. To do so, it is important to study how information processing diffuses through the moving agents and how mobility affects learning and tracking abilities.

In earlier works [4][5], we studied the problem of how to design adaptive networks that are able to solve estimation problems in a fully distributed manner and in real-time. In these networks, each node has access to local information (its local measurements) and can observe the actions taken by its immediate neighbors. Through a diffusion process, the nodes share information locally and cooperate in a manner that enables the entire network to solve an estimation problem in real-time and without the need for centralized processing or fusion centers. In comparison with other distributed approaches that rely on consensus-based techniques, adaptive networks avoid the need to iterate over data and do not require all nodes to converge to the same equilibrium. Instead, both time- and spatial-diversity are exploited to endow the network with learning and tracking abilities. In this paper, we add another dimension of complexity and incorporate node mobility into the design of adaptive networks. Our objective is to develop what we refer to as mobile adaptive networks. These are networks that possess adaptation abilities and can exhibit collective patterns of motion.

This work was supported in part by NSF grants ECS-0601266, ECS-0723441, and CCF-0942936

II. Coherent Motion Model

Consider a set of N nodes distributed over some spatial region in $\mathbb{R}^3$. Let $x_{k,i}$ denote the location vector of a node k at time i relative to some global coordinate system. The location of the center of gravity of the network at this same time instant is denoted by x_i^g and is defined as

$$x_i^g = \sum_{k=1}^{N} x_{k,i}/N \tag{1}$$

That is, x_i^g is the average location of all nodes in the network. In a mobile network, every node k will update its location vector over time according to the rule:

$$x_{k,i+1} = x_{k,i} + \triangle t \cdot v_{k,i+1} \tag{2}$$

where $\triangle t$ represents the time step and $v_{k,i+1}$ is the velocity vector of the node. Several factors influence the determination of the velocity by node k such as the desire to move toward the target, the desire to move in coordination with the other nodes, and the desire to avoid collisions.

To begin with, each node would like to move towards the unknown location of a target, say, w°. We translate this objective to mean that the center of gravity of the network should approach w° as time progresses, i.e.,

$$x_i^g \to w^\circ \quad \text{as } i \to \infty \tag{3}$$

One way to assist with this objective is to have each node adjust its moving direction towards the direction of the target, w°. Ideally, this can happen by having each node select its velocity vector to point along the direction $w^\circ - x_k$, i.e.,

$$v_k = \alpha \cdot \frac{w^\circ - x_k}{\|w^\circ - x_k\|} \tag{4}$$

for some positive scaling factor α that controls the speed of the node. In (4), we are dropping the time index for simplicity and writing $\{x_k, v_k\}$ instead of $\{x_{k,i}, v_{k,i+1}\}$. Also, nodes do not know location w° and, therefore, later we shall replace w° in (4) by a local estimate for it at node k and time i, denoted by $w_{k,i}$ (see (11)).

Additionally, the nodes in the network do not only want to move in the direction of w°, but they want to do so in an organized manner. By self-organization we mean that the nodes avoid collisions by maintaining a certain distance r from neighbors during the motion behavior. Specifically, every node k would like to satisfy the relation below in reference to its neighbors:

$$r - \varepsilon \leq \|x_k - x_l\| \leq r + \varepsilon \quad \text{for all } l \in \mathcal{N}_k \setminus \{k\} \tag{5}$$

where ε is a small positive number. To achieve cohesion and to avoid collision, we introduce the following cost function:

$$J_k(v_k) = \sum_{l \in \mathcal{N}_k \setminus \{k\}} [\|(x_k + \triangle t \cdot v_k) - (x_l + \triangle t \cdot v_l)\| - r]^2 \tag{6}$$

In the above, the term $x_k + \triangle t \cdot v_k$ refers to the updated location of node k and the terms $\{x_l + \triangle t \cdot v_l\}$ refer to the updated locations of the neighbors of node k. The minimization of (6) over v_k is meant to ensure that the distance between the updated locations stays close to r. As the discussion will reveal, the above cost function also helps the nodes align their velocities. To determine the optimal v_k, we differentiate (6) with respect to v_k and get

$$\frac{dJ_k(v_k)}{dv_k} = 2 \sum_{l \in \mathcal{N}_k \setminus \{k\}} \Bigg(\triangle t \cdot v_k - \triangle t \cdot v_l - (x_l - x_k) + r \frac{x_l - x_k + \triangle t \cdot (v_l - v_k)}{\|x_l - x_k + \triangle t \cdot (v_l - v_k)\|} \Bigg) \tag{7}$$

Note that $x_l - x_k$ is the relative displacement vector and denotes the location of node l relative to node k. To solve for v_k from (7) we investigate the last term in (7). Figure 1 illustrates the current locations of nodes k and l and their updated locations. The term, $\triangle t \cdot (v_l - v_k)$ is a measure of how misaligned the displacements of nodes l and k are after the update. It is reasonable to assume that this misalignment is small relative to the displacement distance $\|x_l - x_k\|$ because, in general, the velocity of node k will be close to its neighbors' velocities. Thus, we may introduce the approximation:

$$\frac{x_l - x_k + \triangle t(v_l - v_k)}{\|x_l - x_k + \triangle t(v_l - v_k)\|} \approx \frac{x_l - x_k}{\|x_l - x_k\|} \tag{8}$$

The result is a normalized direction vector along the direction connecting nodes l and k. Using (8) and setting the derivative in (7) to 0, the velocity vector by node k to achieve self-organization is found to be:

$$v_k = \frac{1}{|\mathcal{N}_k| - 1} \sum_{l \in \mathcal{N}_k \setminus \{k\}} \left[v_l + \left(1 - \frac{r}{\|x_l - x_k\|}\right) \frac{x_l - x_k}{\triangle t} \right] \tag{9}$$

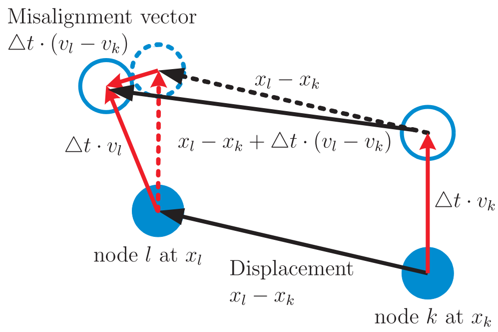


Fig. 1. Location of node l relative to node k.

The operator $|\cdot|$ on a set denotes the number of elements in the set. Expression (9) consists of two terms. The first term is the average velocity of the neighbors of node k, excluding itself. Doing so results in a pattern of collective motion. The second term in (9) is an average of the displacement vectors $\{x_l - x_k\}$; the magnitudes of these vectors are scaled by subtracting from them unit vectors of size r. This term suggests that nodes should adjust their velocity direction to be consistent with the average displacement vector in the neighborhood while maintaining a distance r from their neighbors. A structure similar to (9) was suggested before to induce repulsion and attraction behavior among nodes in a network (see [10]-[11]). Here, we arrived at (9) by starting from the optimization problem (6) and by resorting to the geometric approximation in Fig. 1. Actually, the final structure that we shall adopt for updating the velocity vector appears in (11) below and is different from (9) in two respects. First, expression (11) incorporates the term $w_{k,i} - x_{k,i}$, which relates to the ultimate objective of the network, namely, moving towards the unknown target w°. Second, expression (11) incorporates a term $v^g_{k,i}$, which refers to a local estimate for the velocity of the center of gravity of the network, v^g, defined as

$$v^g \triangleq \frac{1}{N} \sum_{l=1}^{N} v_l \tag{10}$$

Based on (4), (9), and (10), we assume in this work that nodes adjust their velocity vectors according to three criteria as follows:

$$\boxed{v_{k,i+1} = \alpha \frac{w_{k,i} - x_{k,i}}{\|w_{k,i} - x_{k,i}\|} + \beta v^g_{k,i} + \gamma \delta_{k,i}} \tag{11}$$

where $\{\alpha, \beta, \gamma\}$ are non-negative weighting factors and

$$\delta_{k,i} = \frac{1}{|\mathcal{N}_k| - 1} \sum_{l \in \mathcal{N}_k \setminus \{k\}} \left(1 - \frac{r}{\|x_{l,i} - x_{k,i}\|}\right) (x_{l,i} - x_{k,i}) \tag{12}$$

Expression (11) uses local estimates $\{w^{k,i}, v^g_{k,i}\}$ for the global quantities $\{w^\circ, v^g\}$. We now develop diffusion mechanisms that evaluate these local estimates in a fully distributed manner and in real-time.

III. DISTRIBUTED ESTIMATION

A. Measurement Model

As Fig. 2 shows, the distance between the target w° and a node k at any time i is given by the inner product

$$d_k^\circ(i) = u_{k,i}(w^\circ - x_{k,i}) \tag{13}$$

where $u_{k,i}$ denotes the direction of the target including the azimuth angle, $\theta_k(i)$, and the elevation angle, $\varphi_k(i)$, i.e.,

$$u_{k,i} = \begin{bmatrix}\cos\theta_k(i)\cos\varphi_k(i) & \sin\theta_k(i)\cos\varphi_k(i) & \sin\varphi_k(i)\end{bmatrix} \tag{14}$$

Each node is assumed to observe a noisy measurement of the distance to the target, say,

$$d_k(i) = u_{k,i}(w^\circ - x_{k,i}) + n_k(i) \tag{15}$$

where $n_k(i)$ denotes additive noise. Rearranging the above equation, we obtain a linear regression model of the form:

$$\begin{aligned}\hat{d}_k(i) &\triangleq d_k(i) + u_{k,i}x_{k,i} \\ &= u_{k,i}w^\circ + n_k(i)\end{aligned} \tag{16}$$

B. Estimating w°

At every time instant i, every node k is assumed to have access to the local measurements $\{\hat{d}_k(i), u_{k,i}\}$. Using these local data, as well as data shared with their neighbors, the nodes would like to estimate in a distributed manner the global parameter w° that minimizes the following cost function over these variables:

$$J_w^{glob}(w) = \sum_{k=1}^{N} E|\hat{\boldsymbol{d}}_k(i) - \boldsymbol{u}_{k,i}w|^2 \tag{17}$$

where E denotes the expectation operator. Individual nodes cannot optimize (17) because they do not have access to the data across all nodes. We therefore apply the Adapt-then-Combine (ATC) diffusion algorithm [5]. Introduce two sets of non-negative real coefficients $\{c_{l,k}^w\}$ and $\{a_{l,k}^w\}$ satisfying:

$$\begin{aligned}&\sum_{l=1}^{N} c_{l,k}^w = \sum_{k=1}^{N} c_{l,k}^w = \sum_{l=1}^{N} a_{l,k}^w = 1 \\ &c_{l,k}^w = a_{l,k}^w = 0 \text{ if } l \notin \mathcal{N}_k\end{aligned} \tag{18}$$

The superscript w is used to indicate that the set of coefficients is for the estimation problem involving w°. The coefficients refer to the weights across the links in the network. The ATC algorithm consists of two steps. The first step involves local adaptation, where node k incorporates its local information $\{\hat{d}_l(i), u_{l,i}\}$ for $l \in \mathcal{N}_k$ into the processing task. And the second step is a combination step where the estimates from the neighborhood are combined through the coefficients $\{a_{l,k}^w\}$. The algorithm is described as follows:

$$\boxed{\begin{aligned}\psi_{k,i} &= w_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} c_{l,k}^w u_{l,i}^T[\hat{d}_l(i) - u_{l,i}w_{k,i-1}] \\ w_{k,i} &= \sum_{l\in\mathcal{N}_k} a_{l,k}^w \psi_{l,i}\end{aligned}} \tag{19}$$

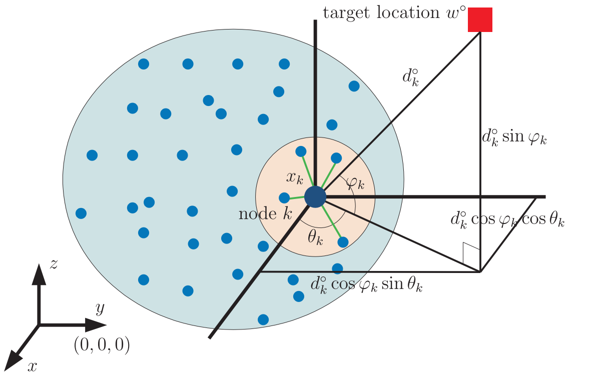


Fig. 2. System model in $\mathbb{R}^3$. The node at location x_k is at a distance d_k° from the target at location w^o. The figure shows the azimuth and elevation angles, θ_k and ψ_k, respectively.

where μ_k is a positive step size used by node k. The resulting estimate of node k at time i is denoted by $w_{k,i}$. According to (19), the nodes in the neighborhood of node k share their intermediate estimates $\{\psi_{l,i}\}$ and measurements $\{\hat{d}_l(i), u_{l,i}\}$.

C. Estimating v^g

The velocity of the center of gravity, v^g, should be also estimated in a distributed way. By definition, v^g is the average velocity of all nodes in the network, as in (10). However, since the velocities of the nodes are changing and so is v^g, we need to keep track of v^g over time. Introduce the global cost function:

$$J_v^{glob}(v^g) = \sum_{k=1}^{N} E\|v_{k,i} - v^g\|^2 \tag{20}$$

Following the derivation in (17), we can arrive at the following diffusion algorithm for estimating v^g:

$$\boxed{\begin{aligned}\phi_{k,i} &= v_{k,i-1}^g + \nu_k \sum_{l\in\mathcal{N}_k} c_{l,k}^v (v_{l,i} - v_{k,i-1}^g) \\ v_{k,i}^g &= \sum_{l\in\mathcal{N}_k} a_{l,k}^v \phi_{l,i}\end{aligned}} \tag{21}$$

where ν_k is a positive step size used by node k and $\{c_{l,k}^v\}$ and $\{a_{l,k}^v\}$ are two sets of non-negative real coefficients satisfying the same properties as (18).

We therefore end up with two diffusion mechanisms. Equations (19) and (21) refer to the diffusion mechanisms for estimating the unknown location, w°, and for tracking the velocity of the center of gravity, v^g, respectively.

IV. MOBILE ADAPTIVE NETWORK

Different variants of the algorithm are possible, for example, by selecting different weighting coefficients. The following variant is one possibility where we set $c_{l,k}^w = c_{l,k}^v = \delta_{l,k}$ in terms of the Kronecker delta function. In this case, the nodes exchange only their estimates and thus the amount of communication is reduced. The following is a summary of the resulting algorithm

ATC Diffusion Algorithm

Start with $\{w_{k,-1} = 0, v^g_{k,-1} = v_{k,-1}\}$. Every node k performs the following steps for $i \geq 0$:

1) The node has access to the local data: $\{d_{k,i}, u_{k,i}, v_{k,i}, x_{k,i}\}$.
2) Perform two local adaptation steps, one for the weight vector, w°, and the other for the velocity of the center of gravity, v^g:

$$\psi_{k,i} = w_{k,i-1} + \mu_k u^T_{k,i}[d_k(i) - u_{k,i}(w_{k,i-1} - x_{k,i})] \quad (22)$$

$$\phi_{k,i} = (1 - \nu_k)v^g_{k,i-1} + \nu_k v_{k,i} \quad (23)$$

3) Perform two local combination steps using data from neighbors: one combines weight estimates for w° and the other combines velocity estimates for v^g:

$$w_{k,i} = \sum_{l \in \mathcal{N}_{k,i}} a^w_{l,k}\psi_{l,i} \quad (24)$$

$$v^g_{k,i} = \sum_{l \in \mathcal{N}_{k,i}} a^v_{l,k}\phi_{l,i} \quad (25)$$

4) Update the node velocity and its location:

$$v_{k,i+1} = \alpha\frac{w_{k,i} - x_{k,i}}{\|w_{k,i} - x_{k,i}\|} + \beta v^g_{k,i} + \gamma\delta_{k,i} \quad (26)$$

$$x_{k,i+1} = x_{k,i} + \triangle t \cdot v_{k,i+1} \quad (27)$$

Note that the neighborhood of node k is now denoted by $\mathcal{N}_{k,i}$ to indicate that the network topology may change due to movement.

V. Simulation Results

In this section, we simulate the motion of mobile networks with 100 nodes. We first specify the neighbors of a node. Let R represent the maximum distance within which two nodes can communicate successfully. All nodes within a radius R of one node are candidate neighbors. However, to reduce computational and communication overhead, the number of neighbors will be constrained, say to N_B.

The simulation parameters are set as follows. The length unit is the body length of a node. The step sizes of updates are $\mu_k = \nu_k = 0.5$ for all k. The combination coefficients are set as $a^w_{l,k} = a^v_{l,k} = 1/|\mathcal{N}_{k,i}|$ if $l \in \mathcal{N}_{k,i}$. For velocity control, the coefficients are $\alpha = 0.5$, $\beta = 1 - \alpha = 0.5$ and $\gamma = 1$. Moreover, the time duration is $\triangle t = 0.5$ sec. In addition, we set $R = 5$, $N_B = 12$, and the optimal distance between two neighbors is $r = 3$.

Initially, nodes are uniformly distributed inside a cube with length 12 and their velocities are set at random directions and unit magnitude. In the following, we compare performance of the ATC diffusion algorithm and a non-cooperative algorithm. For no cooperation, we set $a^w_{l,k} = a^v_{l,k} = \delta_{lk}$ and use

$$v^g_{k,i} = \frac{1}{|\mathcal{N}_{k,i}|}\sum_{l \in \mathcal{N}_{k,i}} v_{l,i}. \quad (28)$$

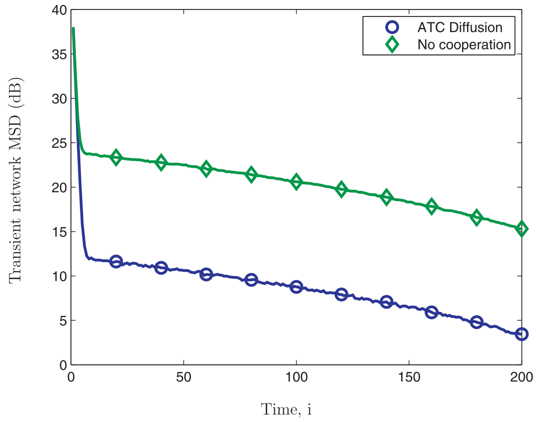


Fig. 3. Transient network MSD of the target location.

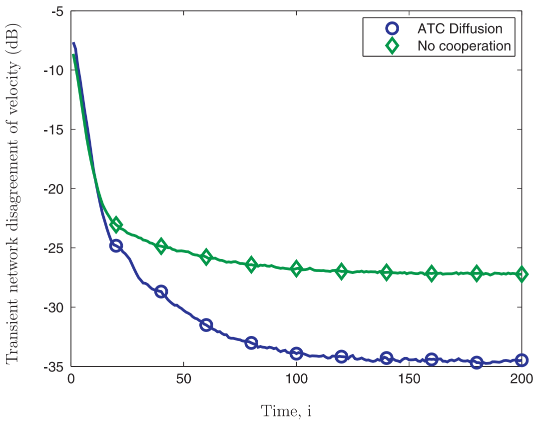


Fig. 4. Transient network disagreement of velocities.

In addition, the measurement noise, $n_k(i)$, in (15) is assumed to be zero-mean Gaussian noise. Intuitively, the noise variance, $\sigma^2_{n,k}(i)$, should vary with the distance between the target and the node since the measurements are noisier at farther distance. We adopt the relation

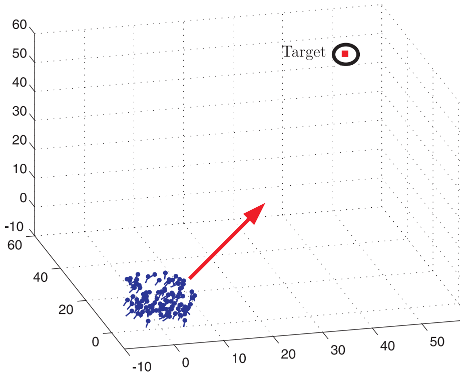
$$\sigma^2_{n,k}(i) = \kappa\|w^\circ - x_{k,i}\|^2 \quad (29)$$

We choose $\kappa = 0.1$ in the sequel. The model is reasonable since we usually assume the signal power to decrease in proportional to the square of the propagation distance.

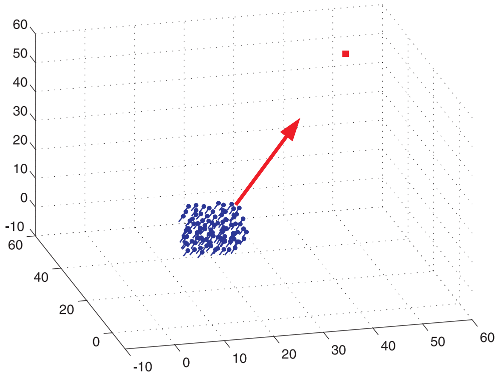
Fig. 3 shows the network transient mean-square deviation (MSD) for estimating w°. The results are averaged over 100 independent experiments. As has been shown in [5], ATC diffusion has better performance than no cooperation. The MSD decreases with time since the network approaches to the target and the noise variance decreases accordingly. Fig. 4 shows the network disagreement, D, of velocity, which is defined as

$$D_i \triangleq \frac{1}{N}\sum_{k=1}^{N}\|v_{k,i} - v^g_i\|^2 \quad (30)$$

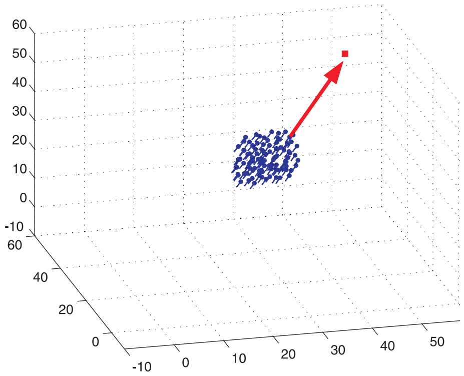
This metric measures the coherence of collective motion of



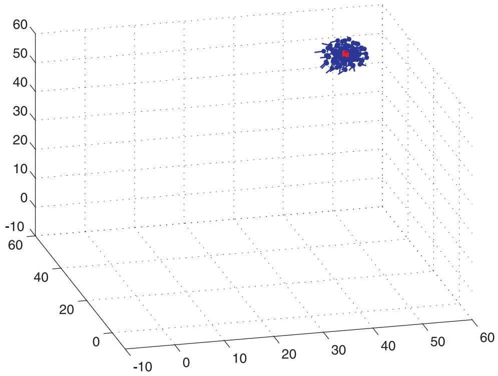
(a) t=0.5 sec



(b) t=25 sec



(c) t=50 sec



(d) t=100 sec

Fig. 5. Maneuvers of mobile network in $\mathbb{R}^3$ over time: (a) $t = 0.5$ sec, (b) $t = 25$ sec, (c) $t = 50$ sec, and (b) $t = 100$ sec.

the network. We observe that the diffusion strategy improves transient and steady-state performance, and helps the network form coherent movement.

We illustrate the maneuver of mobile networks in $\mathbb{R}^3$ over time in Fig. 5. The mark, "■", denotes the target of the network.

VI. Conclusions

In this paper, we developed a diffusion algorithm for mobile adaptive networks such that they can move coherently towards a target with unknown location. Simulation results showed that the network successfully gets to the target and forms an ellipsoid shape. With the aid of the diffusion strategy, the network achieves lower mean-square of velocity disagreement. This demonstrates advantage of the diffusion algorithm in generating such coherent movement.

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] I. D. Couzin, "Collective cognition in animal groups," *Trends in Cognitive Sciences*, vol. 13, pp. 36-43, Jan. 2009.

[3] A. H. Sayed, *Adaptive Filters*, NJ. Wiley, 2008.

[4] 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, Jul. 2008.

[5] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Processing*, vol. 58, pp. 1035-1048, March 2010.

[6] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups for mobile autonomous agents using nearest neighbor rules," *IEEE Trans. on Automatic Control*, vol. 48, pp.988-1001, Jun. 2003.

[7] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems under dynamically changing interaction topologies," *IEEE Trans. on Automatic Control*, vol. 50, pp. 655-661, May 2005.

[8] S. Y. Tu and A. H. Sayed, "Foraging behavior of fish schools via diffustion adaptation," *Proc. International Workshop on Cognitive Information Processing*, Elba Island, Italy, Jun. 2010.

[9] F. Cattivelli and A. H. Sayed, "Self-organization in bird flight formations using diffusion adaptation," *Proc. International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pp. 49-52, Aruba, Dutch Antilles, Dec. 2009.

[10] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. on Automatic Control*, vol. 51, pp. 401-420, Mar. 2006.

[11] S. Janson, M. Middendorf, and M. Beekman, "Honeybee swarms: How do scouts guide a swarm of uninformed bees?" *Anim. Behav*, vol. 70, pp. 349-358, 2005.