2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Advection on Networks with an Application to Decentralized Load Balancing

Airlie Chapman, Eric Schoof and Mehran Mesbahi

***Abstract*—This paper examines an advection-based protocol for the coordination of a networked, multi-agent system. Diffusion forms the basis of the popular consensus dynamics and is closely related to advection. It is with this motivation that we examine a discretized version of advection over a network with the flow field realized through directed graph edges. We endeavor to demonstrate that the subsequent advection protocol forms an attractive set of system dynamics for coordinated control. This paper includes a formulation of the advection dynamics on directed graphs and a presentation of some of its characteristics. We also demonstrate the versatility of the advection dynamics with a decentralized load balancing application.**

***Index Terms*—Advection protocol; Load balancing; Networked control**

I. INTRODUCTION

Advection shares many similarities to diffusion and may be interpreted as diffusion in a flow field. An appeal of the advection framework is that it leads to locally-based interaction dynamics that can produce global network characteristics. Further, the performance characteristics are coupled to the underlying network structure.

As the discretization of the advection dynamics has the effect of inducing a conserved flow through the directed edges in the graph, the state sum is conserved for all time. It is this property as well as the ability to tune the set of equilibrium that makes advection dynamics attractive for control applications. Advection has been used to model the spread of diseases [1], population migration [2], and supply and demand in economic systems [3], and recently applied to formation control and a sensor coverage problem by the authors in [4].

The advection dynamics were independently studied by Berman *et al.* [5] and stochastically simulated for task allocation. In this paper, we highlight some of the features that complement this work and implement a similar load balancing application on a robotics testbed, featured in an attached video.

The organization of the paper is as follows. We begin by defining advection dynamics and characterize its state matrix, dynamics and equilibrium, with a particular focus on the underlying graph structure. We highlight some of the features of advection that make it appropriate for decentralized load balancing application. We conclude with some comments.

The research was supported by AFOSR grant FA9550-09-1-0091 and NSF grant CMMI-0856737. The authors are with the Department of Aeronautics and Astronautics, University of Washington, WA 98105. Emails: {airliec, eschoof, mesbahi}@uw.edu.

II. BACKGROUND AND MODEL

We provide a brief background on constructs and models that will be used in this paper.

Firstly, we define the operator $\mathrm{sgn}(c)$ for $c \in \mathbb{R}$, which is $c/|c|$ for $c \neq 0$ and 0 otherwise. For column vector $v \in \mathbb{R}^p$, v_i denotes the ith element and $|v|$ denotes the number of elements in v. For matrix $M \in \mathbb{R}^{p\times q}$, $[M]_{ij}$ denotes the element in its ith row and jth column. The notations $\|\cdot\|_2$ and $\|\cdot\|_\infty$, denote the 2-norm and the infinity norm, respectively.

The advection equation, also known as the transport equation, involves a scalar concentration u of a material affected by a flow field $\vec{v}$. The flux of the advection process is $F = \vec{v}u$.

In a discrete calculus analogue of the advection equation, we first define an interaction graph (directed and weighted) over nodes based on the flow $\vec{v}$. The flow vector $\vec{v}$ dictates the interactions between nodes by defining directed edges and edge weights. We then adopt a discretized view of the flux $\vec{v}u$ through an edge $i \to j$ as consisting of the flow w_{ji} prescribed by $\vec{v}$ at the edge modified by the concentration x_i prescribed by u at node i. The flow along edge $i \to j$ is consequently $w_{ij}x_i(t)$. The concentration at node i at time t is denoted $x_i(t)$. The flux of node i is then the flow into the node minus the flow out of the node, i.e.,

$$\dot{x}_i(t) = -\sum_{\{\forall j | i \to j\}} w_{ji}x_i(t) + \sum_{\{\forall j | j \to i\}} w_{ij}x_j(t). \quad (1)$$

These dynamics are well suited to a graph theoretic analysis. As such, we proceed by presenting some graph theory background and rewrite the dynamics (1).

A weighted directed graph $\mathcal{G} = (V, E, W)$ is defined by a node set V with cardinality n, an edge set E comprised of pairs of nodes, and a weight set W, where there is a flow from node i to j if $(i, j) \in E$ with edge weight $w_{ji} \in W$. The out-degree matrix $\Delta_{\text{out}}(\mathcal{G}) \in \mathbb{R}^{n\times n}$ is a diagonal matrix with d_i at entry (i, i). The adjacency matrix $\mathcal{A}(\mathcal{G})$ is an $n \times n$ matrix with $[\mathcal{A}(\mathcal{G})]_{ij} = w_{ij}$ when $(j, i) \in E$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. The out-degree Laplacian is defined as $L_{\text{out}}(\mathcal{G}) = \Delta_{\text{out}}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$.

We can now rewrite our dynamics (1) using these graph concepts with the flow $\vec{v}$ generating the graph $\mathcal{G}(\vec{v}) = (V, E(\vec{v}), W(\vec{v}))$. For brevity, we will denote the graph as $\mathcal{G} = (V, E, W)$. The advection dynamics can therefore be written as

$$\dot{x}(t) = -L_{\text{out}}(\mathcal{G})x(t). \quad (2)$$

We proceed to examine system characteristics of the advection dynamics relevant to load balancing.

III. Load Balancing and Advection Properties

Load balancing is a rich area of research with varied applications from mobility-on-demand systems to the reallocation of resources in a business [5], [6], [7], [8]. The advection protocol is well-suited for application to decentralized load balancing, which is demonstrated in the attached video with r ground vehicles corresponding to the load and n ground markers corresponding to the nodes. The global objective is to attain a pre-selected distribution of vehicles at each node.

For our application, the advection dynamics must be quantized and so generally the exact equilibrium can not be reached. Instead, a close integer solution is guaranteed. We refer to the paper by Kashyap *et al.* [9] for the larger set of equilibrium generated by quantization consensus dynamics; a similar results can be attained for the advection dynamics.

The following are properties that support a load balancing application with omitted proofs included in [4], [10].

Features of advection are positive invariance and state sum conservation summarized in the following propositions. These properties ensure that while the protocol is running every portion of the total load will be assigned to some node.

Proposition 1. *The advection dynamics are positively invariant over $x_i \geq 0$ for all $i \in V$, i.e., if $x_i(0) \geq 0$ for all $i \in V$ then $x_i(t) \geq 0$ for all $i \in N$ for all $t > 0$.*

Proposition 2. *The advection dynamics* (2) *are (state) sum conservative, i.e., $\sum_{i=1}^{n} x_i(t) = \sum_{i=1}^{n} x_i(0)$ for all time t.*

The following proposition characterizes the equilibrium for the advection dynamics where $\mathcal{G}$ is strongly connected.[1] The proposition states that if there is a path from each node to every other node, the dynamics will converge to an equilibrium. Further, if load is removed from the system the dynamics will adapt and re-distribute the remaining load, making it robust to load variability.

Proposition 3. *For a strongly connected graph $\mathcal{G}$, the advection dynamics* (2)*, initialized from $x(0) = x_0$, satisfies*

$$\lim_{t\to\infty} x(t) = \frac{1}{\sqrt{n}} \left(\mathbf{1}^T x_0\right) z,$$

where $z = sgn(\bar{z}_1)\bar{z}/\left\|\bar{z}\right\|_2$ and $L_{\text{out}}(\mathcal{G})\bar{z} = 0$. Further, $z_i > 0$ for all $i \in V$ and graphically $\bar{z}_i = \sum_{T\in\mathcal{T}_i}\prod_{(i,j)\in T} w_{ij}$, where $\mathcal{T}_i$ is the set of spanning trees of $\mathcal{G}$ rooted at node i.

A consequence of Proposition 3, summarized in the following corollary, is that if $\mathcal{G}$ is strongly connected then the edges of $\mathcal{G}$ can be re-weighted so that the advection dynamics converges to an arbitrary positive set of equilibrium.

Corollary 1. *For a strongly connected $\mathcal{G} = (V, E, [w_{ij}])$, the equilibrium set is $\frac{1}{z_1}x_1 = \frac{1}{z_2}x_2 \cdots = \frac{1}{z_n}x_n$, where z_i is defined in Proposition 3. A new equilibria set $\alpha_1 x_1 = \alpha_2 x_2 = \cdots = \alpha_n x_n$ where $\alpha_i > 0$, for all $i \in V$, can be achieved by re-weighting the edges of $\mathcal{G}$ forming a new graph $\widetilde{\mathcal{G}} = (V, E, [\widetilde{w}_{ij}])$, with the weights $\widetilde{w}_{ji} = \frac{\alpha_i}{z_i} w_{ji}$ for all $(i,j) \in E$.*

[1] A graph $\mathcal{G}$ is strongly connected if between every pair of distinct vertices there exists a directed path.

The rate of convergence is summarized in the following.

Proposition 4. *For a strongly connected $\mathcal{G}$, the rate of convergence is greater than or equal to the second smallest eigenvalue of $P := DL_{out}(\mathcal{G}) + L_{out}(\mathcal{G})^T D$, denoted $\lambda_2(P)$, where $D = diag(z)$. Here, $z = sgn(\bar{z}_1)\bar{z}/\left\|\bar{z}\right\|_\infty$ and $L_{\text{out}}(\mathcal{G})\bar{z} = 0$.*

Proof: From [10] a governing Lypaunov function is $V(x) = x^T D x$ with $\dot{V}(x) = -\frac{1}{2}x^T P x$ and P is positive semidefinite. For $x^T z = 0$, i.e., x is perpendicular to the set of equilibrium and noting that $\left\|z\right\|_\infty = 1$, $\dot{V}(x) = -\frac{1}{2}x^T P x \leq -\frac{1}{2}\lambda_2(P)x^T x = -\frac{1}{2}\lambda_2(P)\left\|z\right\|_\infty x^T x \leq -\frac{1}{2}\lambda_2(P)x^T D x \leq -\lambda_2(P)V(x)$. Hence using [11] (Theorem 4.10), the result follows. ■

Consequently, as $\lambda_2(P)$ is inversely proportional to the diameter of the graph, denoted diam($\mathcal{G}$),[2] the performance of the protocol depends on structural properties of the graph.

Typically, load balancing for centralized approaches scales poorly with the number of nodes and decentralized approaches often require performance validation through simulation [7], [8]. For our application, each iteration of the protocol requires $|E|$ operations and r communications. The algorithm has a computational complexity of $\mathcal{O}(\text{diam}(\mathcal{G})\,|E|)$ and communication complexity of $\mathcal{O}(\text{diam}(\mathcal{G})r)$, distributed across the n nodes of the graph.

IV. Conclusion

This paper presents an advection-based approach to multi-agent cooperative control. We derive features of the advection protocol and demonstrate its utility through a decentralized load balancing application. One future application of particular interest is the introduction of control nodes, which do not conform to advection, for establishing guided advection networks.

References

[1] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[2] R. A. Horn and C. R. Johnson, *Matrix Analysis*. NY: Cambridge University Press, 1990.

[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, 1979.

[4] A. Chapman and M. Mesbahi, "Advection on graphs," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 1461 – 1466.

[5] S. Berman, A. Halasz, M. A. Hsieh, and V. Kumar, "Optimized Stochastic Policies for Task Allocation in Swarms of Robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927–937, 2009.

[6] M. Pavone, S. Smith, and E. Frazzoli, "Load balancing for mobility-on-demand systems," in *Proceedings of Robotics: Science and Systems*, no. i, 2011.

[7] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *International Journal of Roboitcs Research*, vol. 23, no. 9, pp. 939–954, 2004.

[8] J. Mclurkin and D. Yamins, "Dynamic Task Assignment in Robot Swarms," in *Robotics Science and Systems*, 2005.

[9] A. Kashyap, T. Basar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, Jul. 2007.

[10] A. Chapman and M. Mesbahi, "Stability analysis of nonlinear networks via M-matrix theory: Beyond linear consensus," in *Proc. American Control Conference*, 2012, pp. 6626–6631.

[11] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, NJ, 1996.

[2] The diameter of a graph is the maximum length of all shortest (or minimal) paths between any two nodes.