

Improvement of a Distributed Algorithm for Solving Linear Equations

Xuan Wang, Shaoshuai Mou, *Member, IEEE*, and Dengfeng Sun

Abstract—This paper performs further improvement to a distributed algorithm for solving linear algebraic equations via multi-agent networks recently developed in [1], in which all agents’ states converge exponentially fast to the same solution to a group of linear equations by assuming each agent only knows part of the linear equations and its nearby neighbors’ states. We first prove that the algorithm in [1] with special initialization is able to achieve the solution that is closest to a given point in the Euclidean distance. Second, we eliminate the initialization step required in [1] by a modification to the update equation. Both analytical and numerical results are provided for validation.

Index Terms—Distributed Algorithms, Linear Equations, Multi-agent Network.

I. MOTIVATIONS

MANY research problems in science and engineering can be reformulated as controlling a group of interconnected agents constrained by a network topology, in which each link could represent the leader-follower relation [2]–[7], the information flow in distributed computations [8]–[12], and so on. These network constraints among agents preclude centralized processing and give rise to distributed control with the goal of achieving global objectives through local coordination among nearby neighboring agents [13]. Developing distributed control strategies for solving linear algebraic equations via multi-agent networks are attractive since it allows decomposing a large systems of linear equations into smaller ones that can be cooperatively solved by multi-agent networks [14]–[17]. To be more specific, we are interested in the problem of solving a linear equation $Ax = b$ by employing a network of m agents, where each agent i knows a private equation $A_i x = b_i$ and receives states from its nearby neighbors. Here, $[A \ b]$ is partitioned as follows

$$[A \ b] = \begin{bmatrix} A_1 & b_1 \\ A_2 & b_2 \\ \vdots & \vdots \\ A_m & b_m \end{bmatrix}, \quad A \in \mathbb{R}^{\bar{n} \times n} \quad (1)$$

Suppose each agent i controls a state vector $x_i(t) \in \mathbb{R}^n$, which could be looked as the estimate of agent i to the solution of the overall equation $Ax = b$. A natural idea to enable all agents to achieve a solution of $Ax = b$ is by employing

Manuscript received April 15, 2016; revised September 4, 2016; accepted October 17, 2016. This work was supported by a funding from Northrop Grumman Cooperation.

X. Wang, S. Mou and D. Sun are with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47906 USA (e-mail: wang3156@purdue.edu, mous@purdue.edu, dsun@purdue.edu)

a so-called “agreement principle”, in which each agent i ’s state x_i satisfy is own private equation $A_i x_i(t) = b_i$ while all $x_i(t), i = 1, 2, \dots, m$ reach a consensus [18]–[24]. Under the agreement principle, the authors of [1] have devised the following Distributed Algorithm for solving Linear Equations (DALE):

- 1) **Initialization:** At $t = 0$, each agent i initializes $x_i(0)$ such that $A_i x_i(0) = b_i$;
- 2) **Update:** At $t + 1, t = 0, 1, \dots$, each agent i receives $x_j(t)$ from certain other agents denoted by $\mathcal{N}_i(t)$, and then update its own state to be

$$x_i(t+1) = x_i(t) - P_i \left(x_i(t) - \frac{1}{d_i(t)} \sum_{j \in \mathcal{N}_i(t)} x_j(t) \right) \quad (2)$$

Here, P_i is the orthogonal projection matrix to the kernel of A_i ; $\mathcal{N}_i(t)$ denotes the set of agent i ’s neighbors at time t , from which agent i can receive information. We always assume that each agent is a neighbor of itself, that is, $i \in \mathcal{N}_i(t)$; $d_i(t)$ denotes the cardinality of $\mathcal{N}_i(t)$.

A major result of the DALE is the following theorem:

Theorem 1: [1] Suppose $Ax = b$ has at least one solution and assume that the sequence of neighbor graphs $\mathbb{N}(t), t \geq 1$, is repeatedly jointly strongly connected. Then the DALE drives all $x_i(t)$ converges exponentially fast to the same solution to $Ax = b$ as $t \rightarrow \infty$.

We refer to [1] for introduction of “repeatedly jointly strongly connected”, which has been shown to be not only the sufficient but also necessary requirement of the network connectivity for the above theorem to hold. Throughout this paper we also assume that the underlying neighbor graphs $\mathbb{N}(t)$ is repeatedly jointly strongly connected.

The DALE developed in [1] is applicable to all linear equations that have at least one solution; converges exponentially fast; works for a large class of time-varying directed graphs; operates asynchronously; and does not involve any time-varying step-size. The aim of this paper is to achieve further improvement to the DALE by addressing the following two problems:

- When $Ax = b$ has more than one solutions, the DALE achieves one of them, however, which solution to be achieved by DALE is unclear. In this paper, we will utilize a special initialization step to enable the DALE to achieve a specific solution that is closest to a prescribed point in \mathbb{R}^n in the Euclidean distance. The significance of such solution is that if x represents the state vector needed to

be controlled in industrial process [25]–[27], the obtained solution ensures the system to achieve its goal with the lowest cost.

- The DALE heavily depends on that each state vector $x_i(t)$ is initialized to be a solution to $A_i x = b_i$. When there are round-off errors in such initialization step [28], [29], the DALE might fail to achieve the exact solution to $Ax = b$. Moreover, such initialization step requires each agent to be capable of finding an exact solution to $A_i x = b_i$, which may be out of the computation capability of low-cost agents [30]. In this paper, we will eliminate such initialization step and allows $x_i(0)$ can be chosen arbitrarily from \mathbb{R}^n .

II. ACHIEVE THE SOLUTION CLOSEST TO A SPECIFIED POINT

When $Ax = b$ has multiple solutions, the DALE developed in [1] enables each agent to achieve one of its solutions. But which one is to be achieved is not clear. In this section, we will modify the initialization step of the DALE to achieve a specific solution x_{\min}^q which minimizes $\frac{1}{2}|x - q|^2$ subject to $Ax = b$. That is,

$$x_{\min}^q = \arg \min_{Ax=b} \frac{1}{2}|x - q|^2 \quad (3)$$

where $|\cdot|$ denotes the Euclidean norm. Specially, when $q = 0$, x_{\min}^q becomes the solution to $Ax = b$ with the minimum Euclidean norm. Finding such a x_{\min}^q can be formulated as solving a convex optimization problem by the Dykstras cyclic projection method [31], [32]. It is noted that such cyclic projection method usually requires a centralized scheduling among all agents in the network. In this section we will show that the update (2) achieves x_{\min}^q exponentially fast under the same network connectivity requirement as in [1] by utilizing the following special initialization step:

Initialization (♣): Each agent i initializes its $x_i(0)$ to minimize

$$\frac{1}{2}|x - q|^2 \text{ subject to } A_i x = b_i$$

The above initialization (♣) could be easily completed by solving a linear equation $A_i x = b_i$ and $P_i x = P_i q$ according to the following lemma:

Lemma 1: $x = x_{\min}^q$ if and only if it satisfies $Ax = b$ and $P_A x = P_A q$, where P_A is the orthogonal projection to $\ker A$.

Proof of Lemma 1: By the standard Lagrange multiplier method for convex optimization subject to linear constraints [33], there must exist a λ and x_{\min}^q such that

$$Ax = b \quad (4)$$

$$x - q + A'\lambda = 0 \quad (5)$$

Multiplying P_A to (5), one has $P_A(x - q) + (AP_A)'\lambda = 0$ which and $AP_A = 0$ imply

$$P_A x = P_A q \quad (6)$$

Thus x_{\min}^q must satisfies linear equations (4) and (6), which has the unique solution since $\ker P_A \cap \ker A = 0$ because of the fact $\text{image } P_A = \ker A$.

To summarize, the point in \mathbb{R}^n which is a solution to $Ax = b$ and minimizes $\frac{1}{2}|x - q|^2$ always exists, and must be the unique solution of (4) and (6). Thus Lemma 1 is true. ■

The main result of this section is the following theorem

Theorem 2: With the initialization (♣) and the update (2), all $x_i(t)$ converge exponentially fast to be x_{\min}^q .

Proof of Theorem 2: Note that by the initialization (♣) one still has $A_i x_i(0) = b_i$, which is a special case of the DALE. By Theorem 1, all $x_i(t)$ converges exponentially fast to be the same x^* such that $Ax^* = b$. To prove Theorem 2, we only need to prove that x^* additionally minimizes $\frac{1}{2}|x - q|^2$, or equivalently by Lemma 1, $P_A x^* = P_A q$ holds. Since x^* is the final value that all $x_i(t)$ converges to, it suffices to show

$$P_A x_i(t) = P_A q, \quad i = 1, 2, \dots, n \quad (7)$$

for all $t = 0, 1, 2, \dots$, for which the induction method will be employed in the followings.

We first show

$$P_A x_i(0) = P_A q. \quad (8)$$

From $\text{image } P_A = \ker A$, $\text{image } P_i = \ker A_i$ and $\ker A \subset \ker A_i$, one has $\text{image } P_A \subset \text{image } P_i$. Then

$$\ker P_i \subset \ker P_A$$

which and $\text{image } (I - P_i) = \ker P_i$ imply $P_A(I - P_i) = 0$, that is,

$$P_A P_i = P_A \quad (9)$$

By the initialization (♣), one has $P_i x_i(0) = P_i q$, which and (9) lead to (8).

Now we suppose (7) is true for all i at t , and show it is true at $t + 1$. From (2), (9) and the induction assumption, one has

$$\begin{aligned} P_A x_i(t+1) &= P_A x_i(t) - P_A P_i \left(x_i(t) - \frac{1}{d_i(t)} \sum_{j \in \mathcal{N}_j(t)} x_j(t) \right) \\ &= P_A x_i(t) - \left(P_A x_i(t) - \frac{1}{d_i(t)} \sum_{j \in \mathcal{N}_j(t)} P_A x_j(t) \right) \\ &= \frac{1}{d_i(t)} \sum_{j \in \mathcal{N}_j(t)} P_A q \\ &= P_A q \end{aligned}$$

Thus (7) is true. ■

Remark 1: It is worth pointing out that the initialization (♣) is a special case of the initialization step in the DALE. All the results obtained under the DALE in [1] are directly applicable here.

III. A MODIFIED DALE WITHOUT INITIALIZATION

The DALE in [1] requires an initialization step in which each $x_i(0)$ is initialized to be a solution to $A_i x = b_i$. In order to eliminate the initialization step, we modify the DALE by adding one additional term to (2), which drives all $x_i(t)$ to the manifold $Ax = b$ even when they are not initialized to be so. Similar idea has also been discussed in [34] and [16].

Let $[\bar{A}_i \bar{b}_i]$ denotes a submatrix of $[A_i b_i]$ such that $\ker A_i = \ker \bar{A}_i$ and $\bar{A}_i \bar{A}_i'$ is non-singular, which also implies that $A_i x = b_i$ if and only if $\bar{A}_i x = \bar{b}_i$. Thus the orthogonal projection matrix P_i to the kernel of A_i can be expressed as $P_i = I - \bar{A}_i'(\bar{A}_i \bar{A}_i')^{-1} \bar{A}_i$. The modified DALE will be: At $t = 0$, $x_i(0)$ is any vector in \mathbb{R}^n ; The update for each agent i at $t + 1$ is

$$x_i(t+1) = x_i(t) - P_i \left(x_i(t) - \frac{1}{d_i(t)} \sum_{j \in \mathcal{N}_j(t)} x_j(t) \right) - \bar{A}_i'(\bar{A}_i \bar{A}_i')^{-1}(\bar{A}_i x_i(t) - \bar{b}_i) \quad (10)$$

Remark 2: It is worth mentioning that by multiplying \bar{A}_i to both sides of algorithm (10), one has $\bar{A}_i x_i(t+1) = \bar{A}_i x_i(t) - (\bar{A}_i x_i(t) - \bar{b}_i)$ because of $\bar{A}_i P_i = 0$. Then $\bar{A}_i x_i(t+1) = \bar{b}_i$ for $t = 0, 1, 2, \dots$. This observation implies that $x_i(t)$ for $t \geq 1$ is always a solution to $\bar{A}_i x = \bar{b}_i$ even if $x_i(0)$ is not.

To prove $x_i(t)$ converges to x^* where $Ax^* = b$, we prove the error vector $e_i(t) = x_i(t) - x^*$ converge to 0. From (10), $\bar{A}_i x^* = \bar{b}_i$, and $P_i = I - \bar{A}_i'(\bar{A}_i \bar{A}_i')^{-1} \bar{A}_i$, one has

$$\begin{aligned} e_i(t+1) &= x_i(t) - x^* \\ &\quad - P_i \left(x_i(t) - x^* - \frac{1}{d_i(t)} \sum_{j \in \mathcal{N}_j(t)} (x_j(t) - x^*) \right) \\ &\quad - \bar{A}_i'(\bar{A}_i \bar{A}_i')^{-1}(\bar{A}_i x_i(t) - \bar{A}_i x^*) \\ &= e_i(t) - P_i(e_i(t) - \frac{1}{d_i(t)} \sum_{j \in \mathcal{N}_j(t)} e_j(t)) - (I - P_i)e_i(t) \\ &= \frac{1}{d_i(t)} P_i \sum_{j \in \mathcal{N}_j(t)} e_j(t) \end{aligned}$$

which is the same as the error equation in [1], for which one has the following lemma:

Lemma 2: All $e_i(t)$ converge exponentially fast to be the same value e^* , where $e^* = 0$ when $Ax = b$ has a unique solution.

From Lemma 2 and $e_i(t) = x_i(t) - x^*$, one has all $x_i(t)$ converges to be

$$z^* = e^* + x^*$$

When $Ax = b$ has a unique solution. One has $z^* = x^*$ and all $x_i(t)$ converges to the unique solution x^* . Since e^* may not be zero when $Ax = b$ has multiple solutions, we will need to show that z^* is actually also a solution to $Ax = b$. Note that z^* is the common value to which all $x_i(t)$ converge to. Then z^* satisfies the equations (10). By replacing $x_i(t+1)$ and $x_i(t)$ by z^* , one has

$$\begin{aligned} z^* &= z^* - P_i(z^* - \frac{1}{d_i(t)} \sum_{j \in \mathcal{N}_j(t)} z^*) \\ &\quad - \bar{A}_i'(\bar{A}_i \bar{A}_i')^{-1}(\bar{A}_i z^* - \bar{b}_i), \quad i = 1, 2, \dots, m \quad (11) \end{aligned}$$

which implies

$$\bar{A}_i'(\bar{A}_i \bar{A}_i')^{-1}(\bar{A}_i z^* - \bar{b}_i) = 0, \quad i = 1, 2, \dots, m \quad (12)$$

Multiplying \bar{A}_i to both sides of (12) leads to

$$\bar{A}_i z^* - \bar{b}_i = 0, \quad i = 1, 2, \dots, m \quad (13)$$

Then

$$A_i z^* = b_i, \quad i = 1, 2, \dots, m$$

Thus z^* is a solution to $Ax = b$. To summarize, we have the following theorem:

Theorem 3: The modified DALE (10) with arbitrary $x_i(0) \in \mathbb{R}^n$ drives all $x_i(t)$ to converge exponentially fast to be a common solution to $Ax = b$.

IV. VALIDATION

Numerical simulations will be provided in this section to demonstrate the effectiveness of the main results in Theorem 2 and Theorem 3.

Example 1: Suppose that the following linear equation is solved by the improved DALE with 3 agents:

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & -1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \\ 3 \end{bmatrix} \quad (14)$$

x_{\min}^q and the preferred point q are:

$$x_{\min}^q = \begin{bmatrix} 5.5 \\ 5.5 \\ 3 \end{bmatrix}, \quad q = \begin{bmatrix} 5 \\ 6 \\ 5 \end{bmatrix} \quad (15)$$

Let $V(t) = \sum_{i=1}^3 |x_i(t) - x_{\min}^q|^2$. Then $V(t) = 0$ if and only if all $x_i(t) = x_{\min}^q$. Fig. 1 shows that the DALE in [1] does not achieve x_{\min}^q while the modified DALE with the special initialization step (♣) does.

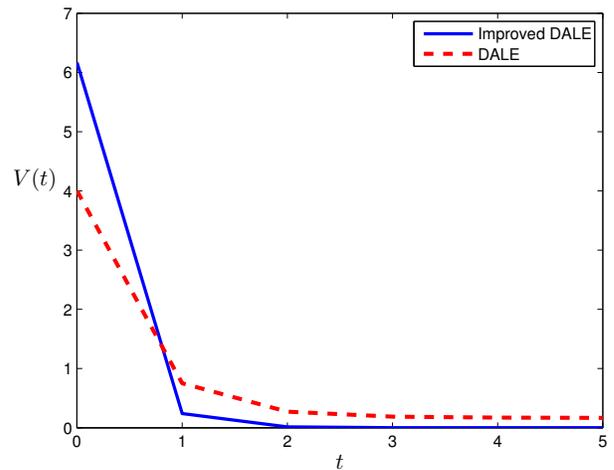


Fig. 1. The improved DALE with special initializations achieves x_{\min}^q in Example 1

We also provide Fig. 2 to illustrate the detailed process of the improved DALE to achieve x_{\min}^q . By $P_i x_i(0) = P_i q$ and

$A_i x_i(0) = b_i$, each $x_i(0)$ is chosen as the closest point to q subject to $A_i x = b_i$. All the $x_i(t)$ then converge to reach a consensus value which is x_{\min}^q as indicated in Fig. 2.

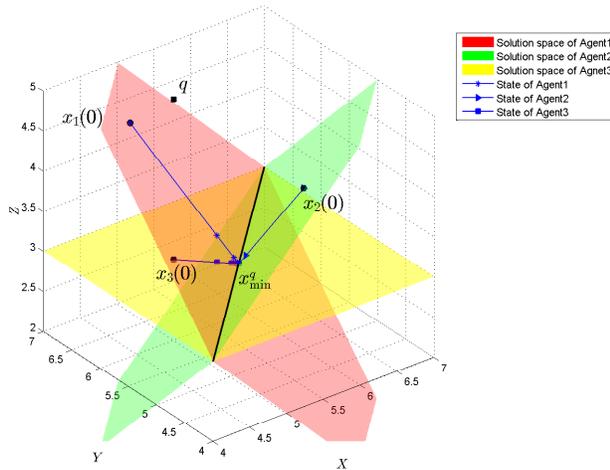


Fig. 2. The process of each agent achieves x_{\min}^q in 3D space

Example 2: Given the following linear equation,

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} 19 & 0 & 12 & 5 & 10 \\ 12 & 16 & 10 & 5 & 5 \\ 5 & 12 & 16 & 14 & 16 \\ 4 & 16 & 16 & 17 & 4 \\ 13 & 4 & 5 & 7 & 4 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 17 \\ 99 \\ 9 \\ 51 \end{bmatrix} \quad (16)$$

with the unique solution $x^* = [0.96 \ 2.77 \ -12.46 \ 7.25 \ 10.10]'$. Let $V(t) = \sum_{i=1}^3 |x_i(t) - x^*|^2$. Then $V(t) = 0$ if and only if all $x_i(t) = x^*$. Fig. 3 shows that the modified DALE (10) without any initialization step drives all agents exponentially fast to x^* .

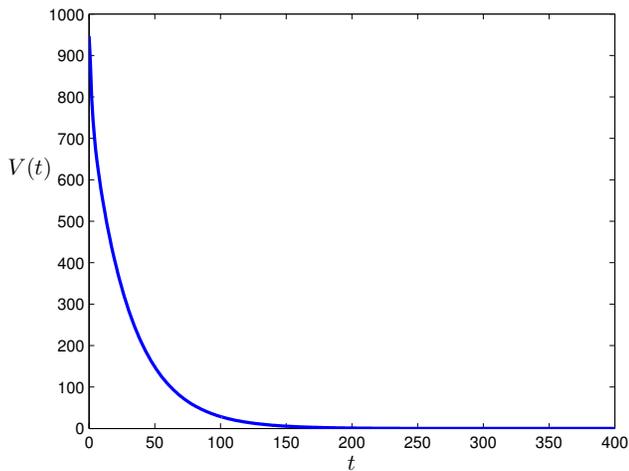


Fig. 3. The speed $x_i(t)$ achieves x^* in Example 2

In addition, Table I about an index $D_i(t) = |A_i x_i(t) - b_i|^2$ demonstrates that for any $t \geq 1$, $x(t)$ is a solution to $A_i x = b_i$, which agree with Remark 2.

TABLE I
SOLUTION ERROR IN EACH AGENT

Number of Agent	$D_i(t) = A_i x_i(t) - b_i ^2$		
	$t = 0$	$t = 1$	$t = 2$
Agent 1	314	6.0×10^{-15}	2.0×10^{-29}
Agent 2	9882	7.3×10^{-14}	3.3×10^{-28}
Agent 3	2601	1.5×10^{-14}	5.1×10^{-29}

V. CONCLUSION

In this paper we have performed further improvements to the distributed algorithm for solving linear equations (DALE) developed in [1]: For equations with more than one solutions, the modified initialization (\clubsuit) guarantees the DALE to find the solution that has the smallest Euclidean distance to a given point in \mathbb{R}^n ; For agents with limited computation capability, the improved DALE (10) enables all agents to achieve exponentially fast a solution to the whole linear equations without any initialization step. Our future work includes further improvement on DALE for achieving finite-time convergence using the ideas appearing in [35], [36], approaching the least square solution even if the overall linear equations do not have solutions as in [37], [38] and being resilient to cyber attacks to agents in the network [39].

REFERENCES

- [1] S. Mou, J. Liu, and A. S. Morse, "A distributed algorithm for solving a linear algebraic equation," *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 2863–2878, Nov. 2015.
- [2] Z. Sun, S. Mou, B. D. O. Anderson, and M. Cao, "Exponential stability for formation control systems with generalized controllers: A unified approach," *Systems & Control Letters*, vol. 93, pp. 50–57, Jul. 2016.
- [3] S. Mou, M. A. Belabbas, A. S. Morse, Z. Sun, and B. D. O. Anderson, "Undirected formations are problematic," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2821–2836, Dec. 2016.
- [4] X. Chen, M. A. Belabbas, and T. Basar, "Controllability of formations over directed time-varying graphs," *IEEE Trans. Control of Network Systems*, pp. 4362–4367, Nov. 2015.
- [5] X. Chen, "Decentralized formation control with a quadratic lyapunov function," *Proceedings of American Control Conference*, pp. 4362–4367, Jul. 2015.
- [6] M. Cao, C. Yu, and B. D. O. Anderson, "Formation control using range-only measurements," *Automatica*, vol. 47, no. 4, pp. 776–781, Apr. 2011.
- [7] S. Mou, M. Cao, and A. S. Morse, "Target-point formation control," *Automatica*, vol. 61, pp. 113–118, Nov. 2015.
- [8] X. Chen, M. A. Belabbas, and T. Basar, "Distributed averaging with linear objective maps," *Automatica*, vol. 70, no. 3, pp. 179–188, Aug. 2016.
- [9] F. He, S. Mou, J. Liu, and A. S. Morse, "Convergence rate on periodic gossiping," *Information Sciences*, pp. 111–125, Oct. 2016.
- [10] C. Li and N. Elia, "Stochastic sensor scheduling via distributed convex optimization," *Automatica*, vol. 58, pp. 173–182, Aug. 2015.
- [11] S. Mou and M. Cao, "Distributed averaging using compensation," *IEEE Commun. Lett.*, no. 8, pp. 1672–1675, Aug. 2013.
- [12] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 622–629, Apr. 2004.
- [13] P. Ramazi, J. Riehl, and M. Cao, "Networks of conforming or nonconforming individuals tend to reach satisfactory decisions," *Proceedings of the National Academy of Sciences*, Oct. 2016.
- [14] S. Mou and A. S. Morse, "A fixed-neighbor, distributed algorithm for solving a linear algebraic equation," *European Control Conference*, pp. 2269–2273, Jul. 2013.

- [15] S. Mou, J. Liu, and A. S. Morse, "A distributed algorithm for solving a linear algebraic equation," *the 51st Annual Allerton Conference on Communication, Control, and Computing*, pp. 267–274, Nov. 2013.
- [16] B. D. O. Anderson, S. Mou, U. R. Helmke, and A. S. Morse, "Decentralized gradient algorithm for solution of a linear equation," *Numerical Algebra, Control and Optimization*, vol. 6, no. 3, pp. 319–328, Sep. 2016.
- [17] J. Wang and N. Elia, "Distributed solution of linear equations over unreliable networks," *Proceedings of American Control Conference*, pp. 6471–6476, Jul. 2016.
- [18] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, pp. 988–1001, Jun. 2003, also in Proc. 41st IEEE CDC, pages 2953 - 2958, 2002.
- [19] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May. 2005.
- [20] W. Ren, "On consensus algorithms for double-integrator dynamics," *IEEE Trans. Autom. Control*, vol. 53, pp. 1503–1509, 2008.
- [21] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus in a dynamically changing environment: a graphical approach," *SIAM Journal on Control and Optimization*, vol. 47, pp. 575–600, Feb. 2008.
- [22] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus in a dynamically changing environment - convergence rates, measurement delays and asynchronous events," *SIAM Journal on Control and Optimization*, pp. 610–623, Feb. 2008.
- [23] J. Qin, W. Zheng, and H. Gao, "Consensus of multiple second-order vehicles with a time-varying reference signal under directed topology," *Automatica*, vol. 47, pp. 1983–1991, Sep. 2011.
- [24] J. Qin, H. Gao, and W. Zheng, "Second-order consensus for multi-agent systems with switching topology and communication delay," *Systems and Control Letters*, vol. 60, pp. 390–397, Jun. 2011.
- [25] C. O. Martinez and V. Puig, "Piece-wise linear functions-based model predictive control of large-scale sewage systems," *IET Control Theory Applications*, vol. 4, pp. 1581–1593, Sep. 2010.
- [26] D. H. Lee, S. A. Zaheer, and J. H. Kim, "Ad hoc network-based task allocation with resource-aware cost generation for multirobot systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6871–6881, Dec. 2014.
- [27] O. C. Mak, Y. C. Wong, and A. Ioinovici, "Step-up dc power supply based on a switched-capacitor circuit," *IEEE Trans. Ind. Electron.*, vol. 42, pp. 90–97, Feb. 1995.
- [28] A. E. Garjanov, "Controlled round-off error oscillations for initial value problem numerical solution," in *Control of Oscillations and Chaos, 1997 1st International Conference*, vol. 2, pp. 333–334, Aug. 1997.
- [29] M. F. Iacchetti, R. Perini, M. S. Carmeli, F. Castelli-Dezza, and N. Bressan, "Numerical integration of odes in real-time systems like state observers: Stability aspects," *IEEE Trans. Ind. Appl.*, vol. 48, no. 1, pp. 132–141, Jan. 2012.
- [30] P. Iovanna, F. Testa, R. Sabella, A. Bianchi, M. Puleri, M. R. Casanova, and A. Germoni, "Packet-optical integration nodes for next generation transport networks," *IEEE/OSA Journal of Opti. Commun. and Netw.*, vol. 4, pp. 821–835, Oct. 2012.
- [31] R. L. Dykstra, "An iterative procedure for obtaining i-projections onto the intersection of convex sets," *The Annals of Probability*, pp. 975–984, 1985.
- [32] H. H. Bauschke and J. M. Borwein, "Dykstra's alternating projection algorithm for two sets," *Journal of Approximation Theory*, vol. 79, no. 3, pp. 418–443, Dec. 1994.
- [33] S. Boyd and L. Vandenberghe, "Convex optimization," *Cambridge University Press*, pp. 141–142, 2004.
- [34] L. Wang, D. Fullmer, and A. S. Morse, "A distributed algorithm with an arbitrary initialization for solving a linear algebraic equation," *Proceedings of American Control Conference*, pp. 1078–1081, Jul. 2016.
- [35] L. Wang and F. Xiao, "Finite-time consensus problems for networks of dynamic agents," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 950–955, Apr. 2010.
- [36] S. Sundaram and C. N. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," *Proceedings of American Control Conference*, pp. 711–716, Jul. 2007.
- [37] G. Shi and B. D. O. Anderson, "Distributed network flows solving linear algebraic equations," *Proceedings of American Control Conference*, pp. 2864–2869, Jul. 2016.
- [38] S. Mou, A. S. Morse, Z. Lin, L. Wang, and D. Fullmer, "A distributed algorithm for efficiently solving linear equations and its applications," *Systems and Control Letters*, vol. 91, pp. 21–27, May. 2016.
- [39] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun.*, no. 4, pp. 766–781, Apr. 2013.



Xuan Wang received his B. S. and M. S. degree in Control Science and Engineering from Harbin Institute of Technology, Harbin, China, in 2014 and 2016, respectively. Between February 2013 and July 2013, he was a exchange student in College of Electrical Engineering & Computer Science, National Taiwan University, Taiwan, China. Currently, he is an Ph.D. student in School of Aeronautics and Astronautics at Purdue University.



Shaoshuai Mou (M'08) received his B. S. and M. S. from Harbin Institute of Technology, China, in 2006 and 2008, respectively. He finished his Ph.D. study under the supervision of Prof. A. Stephen Morse at Yale University in 2014. He then joined MIT as a postdoctoral researcher. Since July 2015 he has started working as a tenure-track assistant professor at Purdue University.



Dengfeng Sun is an Associate Professor in School of Aeronautics and Astronautics at Purdue University, West Lafayette, Indiana. Before joining Purdue, he was an Associate Scientist with University of California Santa Cruz at NASA Ames Research Center. He received a bachelor's degree in precision instruments and mechnology from Tsinghua University in China, a master's degree in industrial and systems engineering from the Ohio State University, and a PhD degree in civil engineering from University of California at Berkeley. Dr. Sun's research interests include control and optimization of large scale systems and their applications in aerospace engineering.