Decentralized Dynamic Optimization Through the Alternating Direction Method of Multipliers

Qing Ling and Alejandro Ribeiro

***Abstract*—This paper develops the application of the alternating direction method of multipliers (ADMM) to optimize a dynamic objective function in a decentralized multi-agent system. At each time slot, agents in the network observe local functions and cooperate to track the optimal time-varying argument of the sum objective. This cooperation is based on maintaining local primal variables that estimate the value of the optimal argument and auxiliary dual variables that encourage proximity with neighboring estimates. Primal and dual variables are updated by an ADMM iteration that can be implemented in a distributed manner whereby local updates require access to local variables and the most recent primal variables from adjacent agents. For objective functions that are strongly convex and have Lipschitz continuous gradients, the distances between the primal and dual iterates to their corresponding time-varying optimal values are shown to converge to a steady state gap. This gap is explicitly characterized in terms of the condition number of the objective function, the condition number of the network that is defined as the ratio between the largest and smallest nonzero Laplacian eigenvalues, and a bound on the drifts of the optimal primal variables and the optimal gradients. Numerical experiments corroborate theoretical findings and show that the results also hold for non-differentiable and non-strongly convex primal objectives.**

***Index Terms*—Alternating direction method of multipliers (ADMM), decentralized multi-agent system, dynamic optimization.**

I. Introduction

We consider a multi-agent system composed of n networked agents whose goal at time k is to solve a decentralized dynamic optimization problem with a separable cost of the form

$$\min \sum_{i=1}^{n} f_i^k(\tilde{x}). \tag{1}$$

The variable $\tilde{x} \in \mathbb{R}^p$ is common to all agents that have as their goal the determination of the vector $\tilde{x}^*(k) := \arg\min \sum_{i=1}^{n} f_i^k(\tilde{x})$ that solves (1). The problem is decentralized because the cost is separated into convex functions $f_i^k : \mathbb{R}^p \to \mathbb{R}$ known to different agents i and dynamic because the functions f_i^k change over time. The purpose of this paper is to develop the application of the alternating directions method of multipliers (ADMM) to the solution of (1).

Problems having the general structure in (1) arise in decentralized multi-agent systems whose tasks are time-varying. Problems of this sort are typical in wireless sensor networks and autonomous teams, with specific examples including estimation of the path of a stochastic process [2], signal detection with adaptive filters [3], tracking moving targets [4], and scheduling trajectories in an autonomous team of robots [5]. In the case of static problems, i.e., when the functions $f_i^k = f_i$ are the same for all times k, there are many iterative algorithms that enable decentralized solution of (1) which can be classified as operating in either the primal [6]–[10] or dual domain [11]–[17]. Primal domain methods determine new iterates by averaging local solutions with those of neighbors and descending along negative subgradient directions [6]–[10]. Dual ascent methods rely on the observation that subgradients of the dual function depend on local and neighboring variables only and can thereby be computed without global cooperation [11]. Convergence of primal descent and dual ascent algorithms is typically slow with dual ascent methods exhibiting somewhat faster convergence. The convergence rate of dual ascent can be further sped up by using distributed Newton methods [12] or the ADMM algorithm [13]–[17]. The ADMM modifies dual ascent by introducing a quadratic regularization term that reduces the variability of subsequent iterates. Reducing this variability improves numerical stability and results in a convergence rate that, while not dramatically different from that of dual ascent, is noticeably better in problems with ill-conditioned dual functions [18]–[21]. Of particular note, the ADMM has been proved to converge linearly to both the primal and dual optimal solutions, when all local objective functions are strongly convex and have Lipschitz continuous gradients [22].

Since a dynamic optimization problem can be considered as a sequence of static optimizations any of the methods in [6]–[17] can be utilized in their solution. This has indeed been tried in, e.g., [23], [24], where separate time scales are assumed so that the descent iterations are allowed to converge in between different instances of (1). This is not entirely faithful to the time-varying nature of (1) motivating the introduction of algorithms that consider the same time scale for the evolution of the functions f_i^k and the iterations of the distributed optimization algorithm [2], [25]–[27]. In their respective contexts these papers establish that if the change in the functions f_i^k is suffi-

Manuscript received June 28, 2013; revised October 03, 2013; accepted November 29, 2013. Date of publication December 16, 2013; date of current version February 10, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Joakim Jalden. The work of Q. Ling was supported by NSFC 61004137. The work of A. Ribeiro was supported by NSF CAREER CCF-0952867, NSF CCF-1017454, and AFOSR MURI FA9550-10-1-0567. This paper was presented in part at the 14th IEEE Workshop on Signal Processing Advances for Wireless Communications, Darmstadt, Germany, June16–19, 2013.

Q. Ling is with the Department of Automation, University of Science and Technology of China, Hefei, Anhui 230026, China and also with The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, Shannxi 710071, China.

A. Ribeiro is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA.

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.2013.2295055

ciently slow minor modifications of static algorithms work reasonably well on keeping track of the time-varying optimal argument $\tilde{x}^*(k)$. The goal of this paper is to show that the same is true when applying the ADMM to a dynamic optimization problem.

We begin the paper by introducing a formal problem definition and the dynamic ADMM algorithm which is based on the introduction of local variables subject to consensus constraints and alternating minimization of an augmented time-varying Lagrangian (Section II). We further manipulate iterations and introduce an initialization condition so that a simple decentralized algorithm is obtained (Proposition 1). We then proceed to analyze convergence properties of the dynamic ADMM algorithm assuming that the local objective functions are strongly convex and have Lipschitz continuous gradients (Section III). These conditions are sufficient to ensure that the distances between the primal and dual iterates to their corresponding time-varying optimal values contract between successive steps (Theorem 2) from where it follows that they reach a steady state optimality gap (Theorem 3). This gap is characterized in terms of problem specific constants and is shown to be proportional to the condition number of the primal function, the condition number of the network that is defined as the ratio between the largest and smallest nonzero Laplacian eigenvalues, and a bound on the drifts of the optimal primal variables and the optimal gradients. Numerical results for a tracking application are presented (Section IV) for strongly convex objectives (Section IV-A) as well as for non-strongly convex, non-differentiable primal objective functions (Sections IV-B, IV-C, and IV-D). These numerical analyses corroborate theoretical findings and show that the dynamic ADMM algorithm also works for non-differentiable and non-strongly convex primal objectives functions. Concluding remarks are presented to close the paper (Section V).

Notation: For column vectors $v_1, \ldots, v_n$ we use the notation $v := [v_1; \ldots; v_n]$ to represent the stacked column vector v. For a block matrix M we use $(M)_{i,j}$ to denote the (i,j)th block. Given matrices $M_1, \ldots, M_n$ we use $\mathrm{diag}(M_1, \ldots, M_n)$ to denote the block diagonal matrix whose ith diagonal block is M_i. A sequence $s(k)$ is said to converge Q-linearly to s^* if $\|s^{k+1} - s^*\|/\|s^k - s^*\| \leq \theta$ for all times k where $\theta \in (0,1)$ is a constant. A sequence $s(k)$ is said to converge R-linearly to s^* if $\|s^k - s^*\| \leq \tau\theta^k$ for all times k where $\theta \in (0,1)$ and $\tau > 0$ are constants.

II. Problem Formulation and Algorithm Design

Consider a network composed of a set of n agents $\mathcal{V} = \{1, \ldots, n\}$ and a set of m arcs $\mathcal{A} = \{1, \ldots, m\}$, where each arc $e \sim (i,j)$ is associated with an ordered pair (i,j) indicating that i can communicate to j. We assume the network is connected and the communication is bidirectional so that if $e \sim (i,j)$ there exists another arc $e' \sim (j,i)$. The set of agents adjacent to i is termed its neighborhood and denoted as $\mathcal{N}_i$. The cardinality of this set is the degree d_i of agent i. We define the block arc source matrix $A_s \in \mathbb{R}^{mp \times np}$ where the block $(A_s)_{e,i} = I_p \in \mathbb{R}^{p \times p}$ is an identity matrix if the arc $e \sim (i,j)$ originates at node i and is null otherwise. Likewise, define the block arc destination matrix $A_d \in \mathbb{R}^{mp \times np}$ where the block $(A_d)_{e,j} = I_p \in \mathbb{R}^{p \times p}$ if the arc $e \sim (i,j)$ terminates at node j and is null otherwise. Here the arcs $e \sim (i,j)$ are sorted first in an ascending order of i and second in an ascending order of j and numbered from 1 to m. Observe that the extended oriented incidence matrix can be written as $E_o = A_s - A_d$ and the unoriented incidence matrix as $E_u = A_s + A_d$. The extended oriented (signed) Laplacian is then given by $L_o = (1/2)E_o^T E_o$, the unoriented (unsigned) Laplacian by $L_u = (1/2)E_u^T E_u$ and the degree matrix containing nodes' degrees d_i in the diagonal is $D = (1/2)(L_o + L_u)$. Denote as Γ_L the largest singular value of L_u and as γ_L the smallest nonzero singular value of L_o. The singular value ratio Γ_L/γ_L is a measure of network connectedness that we refer to as the condition number of the graph.

To solve (1) in a decentralized manner we introduce variables $x_i \in \mathbb{R}^p$ representing local copies of the variable $\tilde{x}$, auxiliary variables $z_{ij} \in \mathbb{R}^p$ associated with each arc $(i,j) \in \mathcal{A}$, and reformulate (1) as

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} f_i^k(x_i), \\ \text{s.t.} \quad & x_i = z_{ij},\ x_j = z_{ij},\ \text{for all } (i,j) \in \mathcal{A}. \end{aligned} \tag{2}$$

The constraints $x_i = z_{ij}$ and $x_j = z_{ij}$ imply that for all pairs of agents $(i,j) \in \mathcal{A}$ forming an arc, the feasible set of (2) is such that $x_i = x_j$. We interpret the auxiliary variables z_{ij} as being attached to the arc (i,j) with the purpose of enforcing the equality of the variables x_i and x_j attached to its source agent i and destination agent j. For a connected network these local neighborhood constraints further imply that feasible variables must satisfy $x_i = x_j$ for all, not necessarily neighboring, pairs of agents i and j. As a consequence, the optimal local variables in (2) must coincide with the solution of (1); i.e., $x_i^*(k) = \tilde{x}^*(k)$ for all nodes i.

To simplify discussion define the vector $x = [x_1; \ldots; x_n] \in \mathbb{R}^{np}$ concatenating all variables x_i, the vector $z = [z_1; \ldots; z_m] \in \mathbb{R}^{mp}$ concatenating all variables $z_e = z_{ij}$, and define the aggregate function $f^k : \mathbb{R}^{np} \to \mathbb{R}$ as $f^k(x) := \sum_{i=1}^n f_i^k(x_i)$. Using these definitions and the definitions of the arc source matrix A_s and the arc destination matrix A_d we can rewrite (2) in a matrix form as

$$\min f^k(x), \qquad \text{s.t. } A_s x - z = 0,\ A_d x - z = 0. \tag{3}$$

Further define the matrix $A = [A_s; A_d] \in \mathbb{R}^{2mp \times np}$ stacking the arc source and arc destination matrices A_s and A_d and the matrix $B = [-I_{mp}; -I_{mp}]$ stacking the opposite of two identity matrices so that (3) reduces to

$$\min\ f^k(x), \qquad \text{s.t. } Ax + Bz = 0. \tag{4}$$

To introduce the dynamic ADMM for the problem in (2)—and its equivalent forms in (3) and (4)—consider Lagrange multipliers $\alpha_e = \alpha_{ij}$ associated with the constraints $x_i = z_{ij}$ and Lagrange multipliers $\beta_e = \beta_{ij}$ associated with the constraints $x_j = z_{ij}$. Group the multipliers α_e in the vector $\alpha = [\alpha_1; \ldots; \alpha_m] \in \mathbb{R}^{mp}$ and the multipliers β_e in the vector $\beta = [\beta_1; \ldots; \beta_m] \in \mathbb{R}^{mp}$ which are thus associated with the constraints $A_s x - z = 0$ and $A_d x - z = 0$, respectively. Further define $\lambda = [\alpha; \beta] \in \mathbb{R}^{2mp}$ associated with the constraint $Ax + Bz = 0$, a positive constant $c > 0$, and define the augmented Lagrangian function at time k as

$$L_k(x, z, \lambda) = f^k(x) + \lambda^T(Ax + Bz) + \frac{c}{2}\|Ax + Bz\|^2,$$

which differs from the regular Lagrangian function by the addition of the quadratic regularization term $(c/2)\|Ax + Bz\|^2$.

The dynamic ADMM proceeds iteratively through alternating minimizations of the Lagrangian $L_k(x, z, \lambda)$ with respect to the primal variables x and z followed by an ascent step on the dual variable λ. To be specific, consider arbitrary time k and given past iterates $z(k-1)$ and $\lambda(k-1)$. The primal iterate $x(k)$ is defined as $x(k) := \arg\min_x L_k(x, z(k-1), \lambda(k-1))$ and given as the solution of the first order optimality condition

$$\nabla f^k(x(k)) + A^T\lambda(k-1) + cA^T[Ax(k) + Bz(k-1)] = 0. \tag{5}$$

Using the value of $x(k)$ from (5) along with the previous dual iterate $\lambda(k-1)$ the primal iterate $z(k)$ is defined as $z(k) := \arg\min_z L_k(x(k), z, \lambda(k-1))$ and explicitly given by the solution of the first order optimality condition

$$B^T\lambda(k-1) + cB^T[Ax(k) + Bz(k)] = 0. \tag{6}$$

The dual iterate $\lambda(k-1)$ is then updated by the constraint violation $Ax(k) + Bz(k)$ corresponding to primal iterates $x(k)$ and $z(k)$ in order to compute

$$\lambda(k) = \lambda(k-1) + c[Ax(k) + Bz(k)]. \tag{7}$$

Observe that the step size c in (5)–(7) is the same constant used in the augmented Lagrangian function.

The computations necessary to implement (5)–(7) can be distributed through the network. However, it is also possible to rearrange (5)–(7) so that with proper initialization the updates of the auxiliary variables $z(k)$ are not necessary and the Lagrange multipliers $\alpha \in \mathbb{R}^{mp}$ and $\beta \in \mathbb{R}^{mp}$ can be replaced by a lower dimension vector $\phi = [\phi_1; \ldots; \phi_n] \in \mathbb{R}^{np}$. We do this in the following proposition before showing that these rearranged updates can be implemented in a decentralized manner. The simplification technique is akin to those used in decentralized implementations of the ADMM for static optimization problems; see e.g., [14], [18, Ch. 3].

Proposition 1: Consider iterates $x(k)$, $z(k)$, and $\lambda(k) = [\alpha(k); \beta(k)]$ generated by recursive application of (5)–(7). Recall the definition of $A = [A_s; A_d]$, the oriented incidence matrix $E_o = A_s - A_d$, the unoriented incidence matrix $E_u = A_s + A_d$, the oriented Laplacian $L_o = (1/2)E_o^T E_o$, the unoriented Laplacian $L_u = (1/2)E_u^T E_u$, and the degree matrix $D = (1/2)(L_o + L_u)$. Require the initial multipliers $\lambda(0) = [\alpha(0); \beta(0)]$ to satisfy $\alpha(0) = -\beta(0)$, the initial auxiliary variables $z(0)$ to be such that $E_u x(0) = 2z(0)$ and further define variables $\phi(k) := E_o^T\alpha(k) \in \mathbb{R}^{np}$. Then, for all times $k > 0$ iterates $x(k)$ can be alternatively generated by the recursion

$$\begin{aligned} \nabla f^k(x(k)) + 2cDx(k) &= cL_u x(k-1) - \phi(k-1), \\ \phi(k) &= \phi(k-1) + cL_o x(k). \end{aligned} \tag{8}$$

Proof: See Appendix A. ∎

The iterations in (8) can be implemented in a decentralized manner. To see that this is true consider the component of the update for $x(k)$ corresponding to the variable $x_i(k)$. Using the definitions of the degree matrix D and the unoriented Laplacian L_u we can write this component of the first equality in (8) as

$$\nabla f_i^k(x_i(k)) + 2cd_i x_i(k) = c\sum_{j\in\mathcal{N}_i}[x_i(k-1) + x_j(k-1)] - \phi_i(k-1). \tag{9}$$

Likewise, using the definitions of the oriented Laplacian L_o the update for $\phi_i(k)$ can be written as

$$\phi_i(k) = \phi_i(k-1) + c\sum_{j\in\mathcal{N}_i}[x_i(k) - x_j(k)]. \tag{10}$$

At the initialization stage, we choose $\phi(0)$ in the column space of L_o (e.g., $\phi(0) = 0$). This is equivalent to choosing $\lambda(0) = [\alpha(0); \beta(0)]$ such that both $\alpha(0)$ and $\beta(0)$ are in the column space of E_o. Such initialization is necessary for the analysis in Section III.

The decentralized dynamic ADMM algorithm run by agent i is summarized in Algorithm 1. At the initial time $k = 0$ we initialize local variables to $x_i(0) = 0$ and $\phi_i(0) = 0$. Agent i also initializes its local copies of neighboring variables to $x_j(0) = 0$ for all $j \in \mathcal{N}_i$, which is consistent with the initialization at agent j. For all subsequent times agent i goes through successive steps implementing the primal and dual iterations in (9) and (10) as shown in steps 3 and 5 of Algorithm 1, respectively. Implementation of Step 3 requires observation of the local function f_i^k as shown in Step 2 and availability of neighboring variables $x_j(k-1)$ from the previous iteration. Implementation of Step 5 requires availability of current neighboring variables $x_j(k)$, which become available through the exchange implemented in Step 4. This variable exchange also makes variables available for the update in Step 3 corresponding to the following time index.

Algorithm 1: Decentralized Dynamic ADMM at agent i

Require: Initialize local variables to $x_i(0) = 0$, $\phi_i(0) = 0$.

Require: Initialize neighboring variables $x_j(0) = 0$ for all $j \in \mathcal{N}_i$.

1: **for** times $k = 1, 2, \ldots$ **do**

2: Observe local function f_i^k.

3: Compute local estimate $x_i(k)$ of optimal variable $\tilde{x}^*(k)$ from [cf. (9)]

$$\nabla f_i^k(x_i(k)) + 2cd_i x_i(k) = c\sum_{j\in\mathcal{N}_i}[x_i(k-1) + x_j(k-1)] - \phi_i(k-1).$$

4: Transmit $x_i(k)$ to and receive $x_j(k)$ from neighbors $j \in \mathcal{N}_i$.

5: Update local variable $\phi_i(k)$ as [cf. (10)]

$$\phi_i(k) = \phi_i(k-1) + c\sum_{j\in\mathcal{N}_i}[x_i(k) - x_j(k)].$$

6: **end for**

III. Convergence Analysis

This section analyzes convergence properties of the decentralized dynamic ADMM algorithm (9), (10) by studying the distance $\|x(k) - x^*(k)\|$ between primal iterates $x(k)$ and the optimal primal variables $x^*(k) = [x_1^*(k); \cdots; x_n^*(k)] = [\tilde{x}^*(k); \cdots; \tilde{x}^*(k)]$. Throughout this section we make the following assumptions on the local objective functions f_i^k.

Assumption 1 (Strong Convexity): Local objective functions are differentiable and strongly convex. I.e., for all agents i, times k, and all pairs of points $\tilde{x}_a$ and $\tilde{x}_b$ it holds $[\tilde{x}_a - \tilde{x}_b]^T[\nabla f_i^k(\tilde{x}_a) - \nabla f_i^k(\tilde{x}_b)] \geq m_{f_i^k}\|\tilde{x}_a - \tilde{x}_b\|^2$, where $m_{f_i^k} \geq m_f$ is the strong convexity constant and $m_f > 0$ is a constant.

Assumption 2 (Lipschitz Gradients): Local objective functions have Lipschitz continuous gradients. I.e., for all agents i, times k, and all pairs of points $\tilde{x}_a$ and $\tilde{x}_b$ it holds $\|\nabla f_i^k(\tilde{x}_a) - \nabla f_i^k(\tilde{x}_b)\| \leq M_{f_i^k}\|\tilde{x}_a - \tilde{x}_b\|$, where $M_{f_i^k} \leq M_f$ is the Lipschitz constant and $M_f < +\infty$ is a constant.

Assumptions 1 and 2 imply that the sum functions $f^k(x) := \sum_{i=1}^n f_i^k(x_i)$ are also strongly convex and with Lipschitz gradients. Indeed, since m_f is the minimum of all strong convexity constants it follows from Assumption 1 that for all times k and pairs of points x_a and x_b it holds

$$[x_a - x_b]^T\left[\nabla f^k(x_a) - \nabla f^k(x_b)\right] \geq m_f\|x_a - x_b\|^2. \quad (11)$$

Likewise, since M_f is the maximum of all Lipschitz constants it follows from Assumption 2 for all times k and pairs of points x_a and x_b it holds

$$\left\|\nabla f^k(x_a) - \nabla f^k(x_b)\right\| \leq M_f\|x_a - x_b\|. \quad (12)$$

Assumptions 1 and 2 and their respective global versions in (11) and (12) are customary assumptions in the analysis of descent algorithms.

Observe that in (4) the optimal primal variables $x^*(k)$ and $z^*(k)$ are unique because the primal function $f^k(x)$ is strongly convex, but there are more than one optimal multipliers. There exists, however, a unique optimal multiplier $\lambda^*(k) = [\alpha^*(k); \beta^*(k)]$ where $\alpha^*(k) = -\beta^*(k)$ lies in the column space of E_o. We will show existence and uniqueness of such an $\alpha^*(k)$ in the proof of Lemma 1. We define the vector $u(k) = [z(k); \alpha(k)]$ which combines primal iterate $z(k)$ and dual iterate $\alpha(k)$ as well as the vector $u^*(k) = [z^*(k); \alpha^*(k)]$ concatenating the unique primal optimal value $z^*(k)$ and the unique optimal dual variable $\alpha^*(k)$ lying in the column space of E_o.

We will bound the distance to optimality $\|x(k) - x^*(k)\|$ by the distance to optimality $\|u(k) - u^*(k)\|_G$ associated with the vector $u(k)$ measured in the Euclidean norm with respect to the block diagonal matrix $G := \text{diag}(cI_{mp}, (1/c)I_{mp})$. To study the evolution of the latter distance we introduce two lemmas respectively concerned with the distance reduction associated with the ADMM iteration and the distance increase associated with the drift of the optimal argument $\tilde{x}^*(k)$. The first lemma, relating the distance $\|u(k) - u^*(k)\|_G$ between iterate and optimal value at time k to the distance $\|u(k-1) - u^*(k)\|_G$ between the optimal value associated with time k and the iterate at time $k-1$, is introduced next.

Lemma 1: Consider the dynamic ADMM algorithm defined by (5)–(7). The Lagrange multiplier $\lambda(k) = [\alpha(k); \beta(k)]$ is initialized by $\alpha(0) = -\beta(0)$ where $\alpha(0)$ lies in the column space of E_o and the primal variables are initialized by $E_u x(0) = 2z(0)$. Consider the optimal multiplier $\lambda^*(k) = [\alpha^*(k); \beta^*(k)]$ of (4) where $\alpha^*(k)$ lies in the column space of E_o. Recall the definitions of the vectors $u(k) = [z(k); \alpha(k)]$ that stacks the primal and dual iterates at time k and $u^*(k) = [z^*(k); \alpha^*(k)]$ that stacks the current primal variable and optimal dual variable lying in the column space of E_o at time k. Further define the matrix $G = \text{diag}(cI_{mp}, (1/c)I_{mp})$ and let $\mu > 1$ be an arbitrary constant to which we associate the contraction parameter

$$\delta = \min\left\{\frac{(\mu-1)\gamma_L}{\mu\Gamma_L}, \frac{2cm_f\gamma_L}{c^2\Gamma_L\gamma_L + \mu M_f^2}\right\}, \quad (13)$$

where Γ_L is the largest singular value of the unoriented Laplacian L_u, γ_L is the smallest nonzero singular value of the oriented Laplacian L_o, M_f is the Lipschitz continuity constant of ∇f^k, m_f is the strong convexity constant of f^k, and c is the ADMM stepsize. Then, the norm with respect to G of the difference between $u(k)$ and $u^*(k)$ decreases by a factor of at least $1/\sqrt{1+\delta}$ relative to the difference between $u(k-1)$ and $u^*(k)$

$$\|u(k) - u^*(k)\|_G \leq \frac{\|u(k-1) - u^*(k)\|_G}{\sqrt{1+\delta}}. \quad (14)$$

Proof: See Appendix B. ■

Since the ADMM iteration at time k descends on the dual function associated with the primal function f^k, the result in (14) of Lemma 1 is just a descent bound on the contraction of the distance to optimality between times $k-1$ and k. Observe that, consistent with this observation, the iterates $u(k)$ and $u(k-1)$ are for different times, whereas the optimal vector $u^*(k)$ is the one that corresponds to time k in both sides of the inequality. The reduction in distance from $\|u(k-1) - u^*(k)\|_G$ to $\|u(k) - u^*(k)\|_G$ is determined by the contraction parameter δ, which is the same constant that appears in the analysis of the static decentralized ADMM in [22]. Since we know that $\delta > 0$, it follows that $u(k)$ is closer to $u^*(k)$ than $u(k-1)$. Fixing $\|u(k-1) - u^*(k)\|_G$, larger δ means smaller distance from $u(k)$ to $u^*(k)$ and stronger contraction.

As written in (13) the constant δ yields little insight on the algorithm's performance relative to problem parameters. To make this dependence clearer let us select the step size c and arbitrary constant μ that yield the largest contraction parameter δ for given strong convexity constant m_f, Lipschitz continuity constant M_f, as well as network connectedness constants Γ_L and γ_L. For any $\mu > 1$, the step size $c = M_f\sqrt{\mu/(\Gamma_L\gamma_L)}$ maximizes the second term of the two minimization arguments in (13) and yields

$$\delta = \min\left\{\frac{(\mu-1)\gamma_L}{\mu\Gamma_L}, \frac{m_f}{M_f}\sqrt{\frac{\gamma_L}{\mu\Gamma_L}}\right\}. \quad (15)$$

The first argument of the minimization operator in (15) is monotonically increasing in μ while the second term is monotonically decreasing. To maximize their minimum we choose the constant $\mu > 1$ that makes them equal, implying that we must have

$$\sqrt{\frac{1}{\mu}} = \sqrt{\frac{1}{4}\frac{m_f^2}{M_f^2}\frac{\Gamma_L}{\gamma_L} + 1} - \frac{1}{2}\frac{m_f}{M_f}\sqrt{\frac{\Gamma_L}{\gamma_L}}. \quad (16)$$

Substituting the expression in (16) for the parameter μ in (15) yields the optimal contraction parameter

$$\delta = \frac{m_f}{M_f}\left[\sqrt{\frac{1}{4}\frac{m_f^2}{M_f^2}+\frac{\gamma_L}{\Gamma_L}}-\frac{1}{2}\frac{m_f}{M_f}\right]. \tag{17}$$

The best contraction parameter δ is a function of the upper bound of the condition number M_f/m_f of the primal functions f^k and the condition number of the graph Γ_L/γ_L. Observe that we always have $\delta < 1$ and that we can have small values of δ when $M_f/m_f \gg 1$ or when $\Gamma_L/\gamma_L \gg 1$, i.e., when either the primal functions or the graph are ill conditioned. When the conditioning numbers are such that $\Gamma_L/\gamma_L \gg M_f^2/m_f^2$ the condition number of the graph dominates and we have $\delta \approx \gamma_L/\Gamma_L$ implying that the contraction is determined by the condition number of the graph. When $M_f^2/m_f^2 \gg \Gamma_L/\gamma_L$ the condition number of the primal functions dominates and we have $\delta \approx (m_f/M_f)\sqrt{\gamma_L/\Gamma_L}$. In this latter case the contraction is constrained by both, the condition number of the primal functions and the condition number of the graph.

To complete the analysis of the evolution of $\|u(k)-u^*(k)\|_G$ we relate the norm $\|u(k-1)-u^*(k)\|_G$ with the distance $\|u(k-1)-u^*(k-1)\|_G$. This result is given in the following lemma that shows how the drifts of the optimal primal variables $x^*(k)$ and the optimal gradients $\nabla f^k(x^*(k))$ translate into a drift of the optimal solutions $u^*(k)$.

Lemma 2: Consider the dynamic ADMM algorithm defined by (5)–(7). The Lagrange multiplier $\lambda(k) = [\alpha(k);\beta(k)]$ is initialized by $\alpha(0) = -\beta(0)$ where $\alpha(0)$ lies in the column space of E_o and the primal variables are initialized by $E_u x(0) = 2z(0)$. Let $x^*(k)$ be the optimal solutions of (4) at time k. Define $u(k)$, $u^*(k)$, and G as in Lemma 1. Further define the optimal drift

$$\begin{aligned} g(k) =& \frac{\sqrt{cm}}{\sqrt{n}}\|x^*(k)-x^*(k-1)\| \\ &+ \frac{1}{\sqrt{2c\gamma_L}}\|\nabla f^k(x^*(k)) - \nabla f^{k-1}(x^*(k-1))\|, \end{aligned} \tag{18}$$

where γ_L is the smallest nonzero eigenvalue of the oriented Laplacian L_o, n is the number of nodes, m is the number of arcs, and c is the ADMM stepsize. Then, the Euclidean distance with respect to G between $u(k-1)$ and the optimal argument $u^*(k)$ is upper bounded by the sum of $g(k)$ and the Euclidean distance with respect to G between $u(k-1)$ and the optimal argument $u^*(k-1)$

$$\|u(k-1)-u^*(k)\|_G \le \|u(k-1)-u^*(k-1)\|_G + g(k). \tag{19}$$

Proof: See Appendix C. ■

The gap $g(k)$ determines the drift from $u^*(k-1)$ to $u^*(k)$ on the basis of $u(k-1)$. We expect this gap to be small enough. That is, the drift between the two successive optimal solutions $x^*(k-1)$ and $x^*(k)$ as well as the drift between the two successive optimal gradients $\nabla f^k(x^*(k))$ and $\nabla f^{k-1}(x^*(k-1))$ are small enough; in another word, the change in the functions f^k is sufficiently slow.

Note that in (18) and (19), $\tilde{x}^*(k-1)$ and $u^*(k-1)$ are undefined when $k = 1$. To address this issue, we can define a virtual initial optimization problem $\min \sum_{i=1}^n f_i^0(\tilde{x})$ such that $\tilde{x}^*(0) = 0$ and $u^*(0) = [z^*(0);\alpha^*(0)] = 0$. Combining Lemma 1 and Lemma 2, we get the following theoretical bound which describes the relationship between $\|u(k)-u^*(k)\|_G$ and $\|u(k-1)-u^*(k-1)\|_G$.

Theorem 1: Consider the dynamic ADMM algorithm defined by (5)–(7). The Lagrange multiplier $\lambda(k) = [\alpha(k);\beta(k)]$ is initialized by $\alpha(0) = -\beta(0)$ where $\alpha(0)$ lies in the column space of E_o and the primal variables are initialized by $E_u x(0) = 2z(0)$. Define $u(k)$, $u^*(k)$, and G as in Lemma 1, the positive number δ as in (13) and the time-varying gap $g(k)$ as in (18). Then the distance between $u(k)$ and $u^*(k)$ and the distance between $u(k-1)$ and $u^*(k-1)$, both measured by the norm with respect to G, satisfy

$$\|u(k)-u^*(k)\|_G \le \frac{\|u(k-1)-u^*(k-1)\|_G}{\sqrt{1+\delta}} + \frac{g(k)}{\sqrt{1+\delta}}. \tag{20}$$

Proof: Combining $\sqrt{1+\delta}\|u(k)-u^*(k)\|_G \le \|u(k-1)-u^*(k)\|_G$ in (19) and $\|u(k-1)-u^*(k)\|_G \le \|u(k-1)-u^*(k-1)\|_G + g(k)$ in (14) we obtain (20). ■

Theorem 1 establishes linear convergence of the dynamic ADMM to the neighborhood of optimality. The convergence is discussed upon a vector $u(k)$, which is the combination of the auxiliary primal variable $z(k)$ and the dual variable $\alpha(k)$. The iterates $\|u(k)-u^*(k)\|_G$ are Q-linearly convergent to a neighborhood of 0 with a constant $\sqrt{1+\delta}$. The neighborhood of optimality is characterized by the scaled optimal argument drift $g(k)$ in (18). For static optimization problems, $g(k) = 0$ and Theorem 1 degenerates to Q-linear convergence of $\|u(k)-u^*(k)\|_G$ to 0 with a constant $\sqrt{1+\delta}$, which coincides with the analysis of the static ADMM in [22].

The following theorem relates the distances $\|u(k-1)-u^*(k-1)\|_G$ and $\|x(k)-x^*(k)\|$ so that the convergence result in Theorem 1 can be translated into a more meaningful statement regarding the suboptimality of primal iterates $x(k)$.

Theorem 2: Consider the dynamic ADMM algorithm defined by (5)–(7). The Lagrange multiplier $\lambda(k) = [\alpha(k);\beta(k)]$ is initialized by $\alpha(0) = -\beta(0)$ where $\alpha(0)$ lies in the column space of E_o and the primal variables are initialized by $E_u x(0) = 2z(0)$. Define m_f as the strong convexity constant of f^k in (11), $u(k)$, $u^*(k)$, and G as in Lemma 1, and the time-varying gap $g(k)$ as in (18). The distance between $x(k)$ and $x^*(k)$ measured by the Euclidean norm and the distance between $u(k-1)$ and $u^*(k-1)$ measured by the norm with respect to G satisfies

$$\|x(k)-x^*(k)\| \le \frac{\|u(k-1)-u^*(k-1)\|_G}{\sqrt{m_f}} + \frac{g(k)}{\sqrt{m_f}}. \tag{21}$$

Proof: (53) implies that

$$m_f\|x(k)-x^*(k)\|^2 \le \|u(k-1)-u^*(k)\|_G^2,$$

or equivalently

$$\sqrt{m_f}\|x(k)-x^*(k)\| \le \|u(k-1)-u^*(k)\|_G. \tag{22}$$

Combining (22) with $\|u(k-1)-u^*(k)\|_G \le \|u(k-1)-u^*(k-1)\|_G + g(k)$ in (19) leads to (21). ■

Theorem 2 shows that the primal iterates $x(k)$ are R-linearly convergent to a neighborhood of $x^*(k)$ since the iterates $u(k)$ are Q-linearly convergent to a neighborhood of $u^*(k)$. Both neighborhoods of optimality are characterized by the scaled optimal argument drift $g(k)$ in (18). For static optimization problems $g(k) = 0$ and the convergence is exactly linear to the optimality.

The result in the following theorem is stated in the form of a steady state suboptimality gap which follows from recursive application of the bound in Theorem 2.

Theorem 3: Consider the dynamic ADMM algorithm defined by (5)–(7). The Lagrange multiplier $\lambda(k) = [\alpha(k); \beta(k)]$ is initialized by $\alpha(0) = -\beta(0)$ where $\alpha(0)$ lies in the column space of E_o and the primal variables are initialized by $E_u x(0) = 2z(0)$. Define m_f as the strong convexity constant of f^k in (11) and the corresponding positive numbers δ as in (13). If the time-varying gap $g(k)$ defined in (18) is smaller than g_{max} for all times k, the distance between $x(k)$ and $x^*(k)$ measured by the Euclidean norm satisfies

$$\limsup_{k\to+\infty} \|x(k) - x^*(k)\| \leq \frac{\sqrt{1+\delta}}{\sqrt{m_f}[\sqrt{1+\delta}-1]} g_{max}. \quad (23)$$

Proof: See Appendix D. ■

Remark: The analysis of the proposed decentralized dynamic ADMM requires that the local objective functions are strongly convex. For centralized static optimization, [20] proposes a multi-block ADMM and establishes its *locally* linear convergence in the absence of the strongly convex assumption. Transplanting the multi-block ADMM in [20] to the decentralized dynamic case yields an algorithm in which each arc maintains its Lagrange multiplier update and the agents optimize their local solutions in a Gauss-Seidel manner. To implement the Gauss-Seidel iterates the agents need to predefine an order to update, which is nontrivial for a large-scale multi-agent system. In comparison, the proposed algorithm adopts Jacobi iterates that do not rely on any predefined order. Further, in this paper the analysis is along the line of [19] and [22] where the ADMM has *globally* linear convergence under the assumption of strong convexity.

IV. Numerical Experiments

This section provides numerical experiments to demonstrate the effectiveness of the proposed dynamic ADMM and validate the theoretical analysis. Though the theoretical analysis assumes that the decentralized optimization problem (1) is unconstrained, and the local objective functions are differentiable and strongly convex, the proposed dynamic ADMM is applicable to the constrained, non-differentiable and non-strongly convex cases with a minor modification. Suppose that in (1) the cost functions f_i^k are not necessarily differentiable and the optimization variable $\tilde{x}$ is subject to constraint $\tilde{x} \in \mathcal{X}^k$ where $\mathcal{X}^k$ is a convex set. For this case the update (9) is modified to

$$x_i(k) = \arg\min_{x_i} f_i^k(x_i) + cd_i \|x_i - p_i(k)\|^2, \text{ s.t. } x_i \in \mathcal{X}^k, \quad (24)$$

where $p_i(k) = (1/2d_i)\sum_{j\in\mathcal{N}_i}[x_i(k-1) + x_j(k-1)] - (1/2cd_i)\phi_i(k-1)$ is a proximal point.

We consider a bidirectionally connected network composed of $n = 100$ agents where $m = 1810$ arcs (out of all 9900 possible arcs) are randomly chosen to be connected. At time k agent i measures a true signal $\tilde{x}_0(k)$ through a linear observation function $y_i(k) = H_i(k)\tilde{x}_0(k) + e_i(k)$ where $e_i(k)$ is random noise; $\tilde{x}_0(k) \in \mathbb{R}^2$. Throughout the simulation we let $\tilde{x}_0(k)$ evolves along a nearly circular trajectory that is randomly polluted and $\|\tilde{x}_0(k) - \tilde{x}_0(k-1)\| \leq \rho$ and $\rho = 0.1$ for all k. We consider four cases:

Case 1: The decentralized optimization problem (1) is unconstrained, and the local objective functions are differentiable and strongly convex. Entries of the matrices $H_i(k) \in \mathbb{R}^{2\times 2}$ follow normal distribution $\mathcal{N}(0,1)$ and $H_i(k)^T H_i(k)$ are nonsingular. Entries of the noises $e_i(k) \in \mathbb{R}^2$ follow normal distribution $\mathcal{N}(0, 0.01)$. The objective function at time k is $f^k(\tilde{x}) = \sum_{i=1}^n \frac{1}{2}\|H_i(k)\tilde{x} - y_i(k)\|^2$ and the local objective function of agent i is $f_i^k(\tilde{x}) = \frac{1}{2}\|H_i(k)\tilde{x} - y_i(k)\|^2$. In this case, the update of $x_i(k)$ in (9) reduces to solving a linear equation

$$\left[H_i^T(k)H_i(k) + 2cd_i I_p\right] x_i(k) = H_i^T(k) y_i(k) + 2cd_i p_i(k).$$

Case 2: The decentralized optimization problem (1) is unconstrained, and the local objective functions are differentiable but not strongly convex. Entries of the matrices $H_i(k) \in \mathbb{R}^{1\times 2}$ follow normal distribution $\mathcal{N}(0,1)$ and $H_i(k)^T H_i(k)$ are singular. Entries of the noises $e_i(k) \in \mathbb{R}$ follow normal distribution $\mathcal{N}(0, 0.01)$. The objective function at time k is $f^k(\tilde{x}) = \sum_{i=1}^n \frac{1}{2}\|H_i(k)\tilde{x} - y_i(k)\|^2$ and the local objective function of agent i is $f_i^k(\tilde{x}) = \frac{1}{2}\|H_i(k)\tilde{x} - y_i(k)\|^2$. Note that in this case the update of $x_i(k)$ in (9) is the same as that in Case 1.

Case 3: The decentralized optimization problem (1) is unconstrained, and the local objective functions are neither differentiable nor strongly convex. Entries of the matrices $H_i(k) \in \mathbb{R}^{2\times 2}$ follow normal distribution $\mathcal{N}(0,1)$. Among entries of the noises $e_i(k) \in \mathbb{R}$, 90% are 0 and the remaining 10% follow normal distribution $\mathcal{N}(0,1)$. The objective function at time k is $f^k(\tilde{x}) = \sum_{i=1}^n \|H_i(k)\tilde{x} - y_i(k)\|_1$ and the local objective function of agent i is $f_i^k(\tilde{x}) = \|H_i(k)\tilde{x} - y_i(k)\|_1$. In this case, the update of $x_i(k)$ in (9) minimizes an objective function that is the sum of an ℓ_1-norm term and a least squares term

$$x_i(k) = \arg\min_{x_i} \|H_i(k)x_i - y_i(k)\|_1 + cd_i \|x_i - p_i(k)\|^2.$$

Case 4: The decentralized optimization problem (1) is constrained, and the local objective functions are differentiable and strongly convex. Entries of the matrices $H_i(k) \in \mathbb{R}^{2\times 2}$ follow normal distribution $\mathcal{N}(0,1)$ and $H_i(k)^T H_i(k)$ are nonsingular. Entries of the noises $e_i(k) \in \mathbb{R}^2$ follow normal distribution $\mathcal{N}(0, 0.01)$. We know in advance that the Euclidean distance between $\tilde{x}_0(k)$ and $\tilde{x}_0(k-1)$ is smaller than a threshold ρ. The objective function at time k is $f^k(\tilde{x}) = \sum_{i=1}^n \frac{1}{2}\|H_i(k)\tilde{x} - y_i(k)\|^2$ subject to $\|\tilde{x} - \tilde{x}(k-1)\| \leq \rho$. The local objective function of agent i is $f_i^k(\tilde{x}) = \frac{1}{2}\|H_i(k)\tilde{x} - y_i(k)\|^2$ subject to $\|\tilde{x} - x_i(k-1)\| \leq \rho$. Note that the local and global constraints are different since at time k the agents are unable to obtain a common $\tilde{x}(k-1)$. In this case, the update of $x_i(k)$ in (9) solves a constrained optimization problem

$$x_i(k) = \arg\min_{x_i} \|H_i(k)x_i - y_i(k)\|^2 + cd_i \|x_i - p_i(k)\|^2,$$
$$\text{s.t. } \|x_i - x_i(k-1)\| \leq \rho.$$

In the numerical experiments we compare the proposed dynamic ADMM with independent optimization of agents. By independent optimization we mean that each agent optimizes by itself without collaboration with others. Suppose that agent i's local objective function is $f_i^k(\tilde{x}) = \frac{1}{2}\|H_i(k)\tilde{x} - y_i(k)\|^2$ at time k, its estimate on $\tilde{x}(k)$ is $x_i(k) = [H_i(k)^T H_i(k)]^{-1} H_i(k)^T y_i(k)$. Obviously this approach is not applicable when $H_i(k)^T H_i(k)$ is singular. Hence we only consider independent optimization in Case 1.

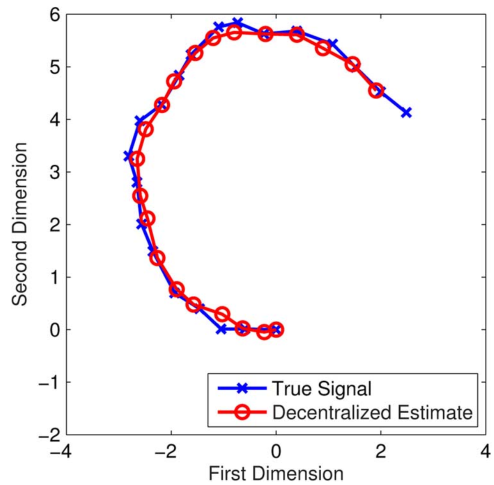

Fig. 1. True signal and decentralized estimate of agent 1 for Case 1.

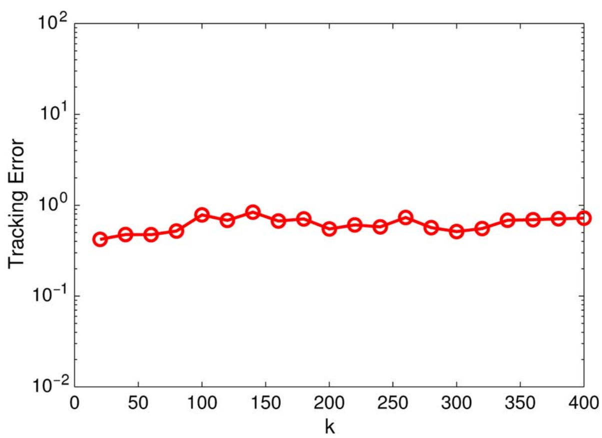

Fig. 2. Tracking error of the dynamic ADMM for Case 1.

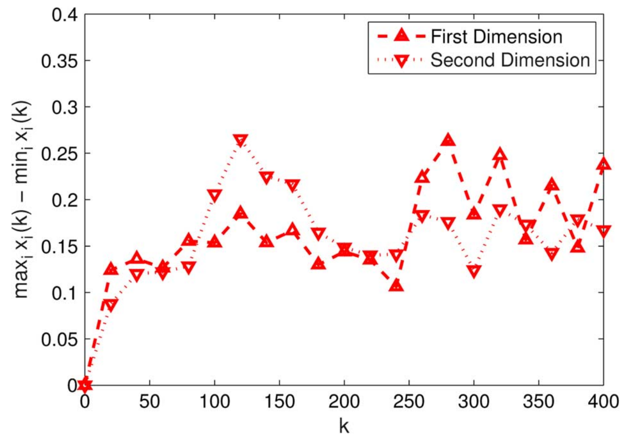

Fig. 3. Maximum distance between decentralized estimates with respect to the two dimensions for Case 1.

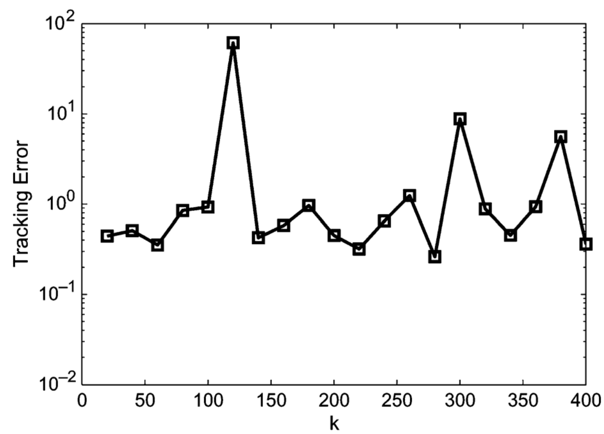

Fig. 4. Tracking error of independent optimization for Case 1.

We evaluate performance of the algorithms by the tracking error which is defined as $(1/n)\sum_{i=1}^{n}\|x_i(k)-\tilde{x}_0(k)\|$.

Throughout the numerical experiments, we set the ADMM parameter c as 1. Though tuning c may lead to better tracking of the signal, we simply fix it since performance of the dynamic ADMM is not sensitive to the value of c as long as it is set as a reasonable value.

A. Case 1

Fig. 1–Fig. 3 show simulation results of the dynamic ADMM for Case 1. Fig. 1 compares the true signal, which is close to the centralized solution, and the decentralized estimate of agent 1. Due to the delay of network information diffusion agents are unable to recognize dynamics of the aggregated objective function. As a result of this essential limitation of dynamic optimization, the decentralized estimate cannot accurately track the true signal. The difference between them is bounded throughout the optimization process as discussed in the theoretical analysis [cf. Theorem 3] and validated by Fig. 2. Fig. 3 shows the maximum distance between decentralized estimates of all the agents with respect to the two dimensions. Though each agent optimizes by itself, the agents keep tight consensus. The key is the optimization of the dual variables which guarantees that the consensus constraints are not violated too much.

As a comparison, Fig. 4 shows tracking error of independent optimization for Case 1. At some time slots, the tracking errors of independent optimization are two-magnitude larger than those of the dynamic ADMM. The main reason is that the dynamic ADMM incorporates information of all agents and achieves collaboration gain on the tracking accuracy, while independent optimization often fails when the observation noise is very large or the local objective functions are ill-conditioned at some agents.

B. Case 2

Fig. 5 and Fig. 6 show simulation results of the dynamic ADMM for Case 2. Compared to Case 1 where the local objective functions are strongly convex, we can observe degradation of the tracking performance when the local objective functions are not strongly convex. Fig. 5 depicts an obvious delay of the decentralized estimate of agent 1 with respect to the trajectory of the true signal; such delay is much larger than that in Case 1 [cf. Fig. 1]. Accordingly, Fig. 6 shows that the tracking error is larger than that in Case 1 [cf. Fig. 2]. We explain its reason below. Non-strong convexity of a local objective function leads to ambiguity of the corresponding local solution. In the dynamic ADMM, introduction of the proximal term helps address this ambiguity issue [cf. (24)]; however, its tracking performance is still limited by the intrinsic non-strong convexity of local objective functions.

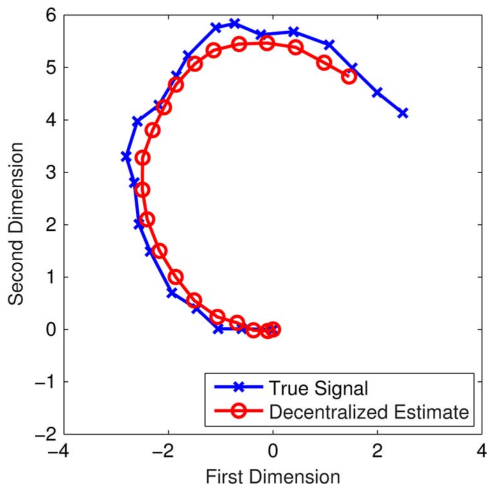

Fig. 5. True signal and decentralized estimate of agent 1 for Case 2.

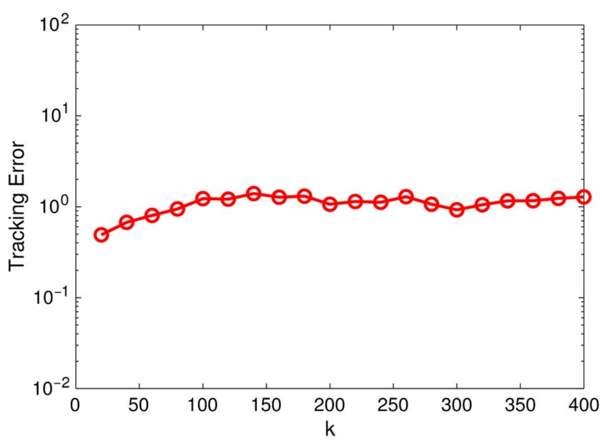

Fig. 6. Tracking error of the dynamic ADMM for Case 2.

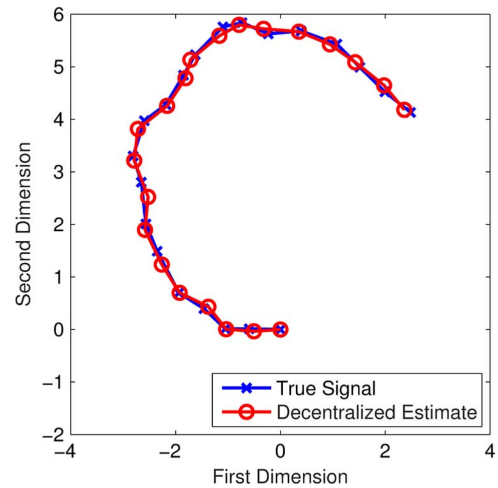

Fig. 7. True signal and decentralized estimate of agent 1 for Case 3.

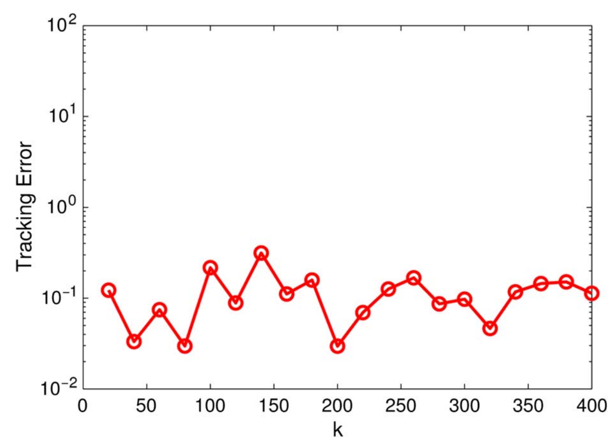

Fig. 8. Tracking error of the dynamic ADMM for Case 3.

Nevertheless, simulation results of Case 2 demonstrates the advantage of the dynamic ADMM over the independent optimization approach and the dynamic Lagrangian method in [2] that are unable to handle the non-strong convexity case. It is the quadratic regularization term appended in the augmented Lagrangian function that introduces a proximal term to each agent and makes each agent solve a strongly convex subproblem. Therefore, the dynamic ADMM has favorable numerical stability similar to the static ADMM.

C. Case 3

Fig. 7 and Fig. 8 show simulation results of the dynamic ADMM for Case 3 in which the local objective functions are neither differentiable nor strongly convex but the non-strong convexity in this case does not bring ambiguity to the local solutions. The non-differentiability is well handled in (24). From Fig. 7 we can observe that the decentralized estimate of agent 1 tracks the true signal well. The tracking error in Fig. 8 is smaller than that of Case 1 since the optimization model exploits sparsity prior of the true signals [cf. Fig. 2].

D. Case 4

Fig. 9 and Fig. 10 show simulation results of the dynamic ADMM for Case 4. This constrained case is nontrivial since in the constraints the inexact solutions $x_i(k-1)$, which are the estimates of $\tilde{x}_0(k-1)$, bring extra uncertainty to the subsequent problems that solves $x_i(k)$. Interestingly, the simulation results of Case 4 are similar with those of Case 1. Fig. 9 depicts that the decentralized estimate of agent 1 closely tracks the true signal. The tracking error is also similar with that in Case 1, as shown in Fig. 10 [cf. Fig. 2].

V. Conclusion

This paper introduces the ADMM to solve a decentralized dynamic optimization problem. Traditionally the ADMM is a powerful tool to solve centralized and/or static optimization problems; we show that a minor modification enables it to adapt to the decentralized dynamic cases. We prove that under certain conditions, the differences between the ADMM iterates and the optimal solutions, in both the primal and the dual domains, can be characterized by the drifts between the successive primal optimal solutions.

Appendix A
Proof of Proposition 1

Proof: Substituting the multiplier update $\lambda(k-1) = \lambda(k) - c[Ax(k) + Bz(k)]$ in (7) into the update for the primal variables $x(k)$ in (5) leads to

$$\nabla f^k(x(k)) + A^T\lambda(k) + cA^TB[z(k-1) - z(k)] = 0. \quad (25)$$

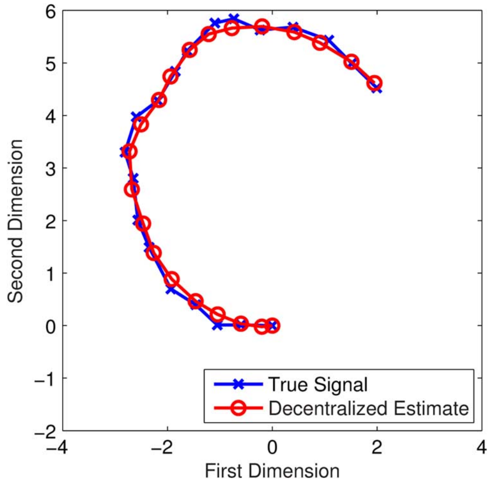

Fig. 9. True signal and decentralized estimate of agent 1 for Case 4.

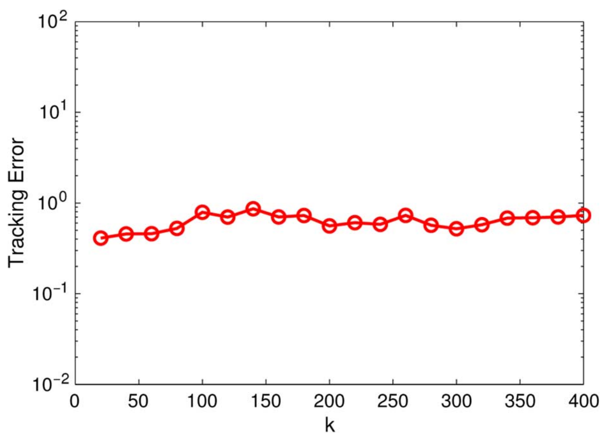

Fig. 10. Tracking error of the dynamic ADMM for Case 4.

Similarly, substituting the multiplier update $\lambda(k-1) = \lambda(k) - c[Ax(k) + Bz(k)]$ in (7) into the expression for the auxiliary variable $z(k)$ in (6) leads to

$$B^T\lambda(k) = 0. \tag{26}$$

Recalling the definitions of $B = [-I_{mp}; -I_{mp}]$ and $\lambda(k) = [\alpha(k); \beta(k)]$ it follows from (26) that $\alpha(k) = -\beta(k)$ for all $k > 0$ other than the initial value. But for $k = 0$ we have $\alpha(0) = -\beta(0)$ by hypothesis from where it follows that $\alpha(k) = -\beta(k)$ for all $k \geq 0$. Using this fact, the definition of $A = [A_s; A_d]$, and the observation that the oriented incidence matrix is $E_o = A_s - A_d$ we can conclude that

$$A^T\lambda(k) = A_s^T\alpha(k) - A_d^T\alpha(k) = E_o^T\alpha(k). \tag{27}$$

Further observe that from the definitions of $A = [A_s; A_d]$, $B = [-I_{mp}; -I_{mp}]$ and the unoriented edge incidence matrix $E_u = A_s + A_d$ it follows that $A^TB = [A_s^T, A_d^T][-I_{mp}; -I_{mp}] = -A_s^T - A_d^T = -E_u^T$. Substituting this expression and the result in (27) into (25) yields

$$\nabla f^k(x(k)) + E_o^T\alpha(k) - cE_u^T[z(k-1) - z(k)] = 0. \tag{28}$$

Consider now (7) and recall that $\lambda(k) = [\alpha(k); \beta(k)]$ to separate the equality along the $\alpha(k)$ and $\beta(k)$ directions

$$\begin{aligned}\alpha(k) &= \alpha(k-1) + c[A_s x(k) - z(k)], \\ \beta(k) &= \beta(k-1) + c[A_d x(k) - z(k)].\end{aligned} \tag{29}$$

Since we know from (26) and the initialization hypothesis that $\alpha(k) = -\beta(k)$ for all $k \geq 0$ we can sum up the two equalities in (29) to obtain $c[A_s x(k) - z(k)] + c[A_d x(k) - z(k)] = 0$. Reorder terms to write

$$\frac{1}{2}E_u x(k) = \frac{1}{2}(A_s + A_d)x(k) = z(k), \tag{30}$$

where we also use the definition of the unoriented edge incidence matrix $E_u = A_s + A_d$ to write the first equality. Further recall that (30) is true for $k = 0$ by hypothesis to conclude that (30) is true for all times $k \geq 0$.

Using (30) to eliminate $z(k)$ from the update for $\alpha(k)$ in (29) yields

$$\begin{aligned}\alpha(k) &= \alpha(k-1) + c\left[A_s x(k) - \frac{1}{2}(A_s + A_d)x(k)\right] \\ &= \alpha(k-1) + \frac{c}{2}E_o x(k).\end{aligned} \tag{31}$$

Here we use the definition of the oriented edge incidence matrix $E_o = A_s - A_d$. Multiplying both sides of (31) by E_o^T and using the definitions of the oriented Laplacian matrix $L_o = (1/2)E_o^T E_o$ and the vector $\phi(k) = E_o^T\alpha(k)$, we obtain the update for $\phi(k)$ in (8). Likewise, use (30) to eliminate $z(k)$ and $z(k-1)$ from (28) so as to write

$$\nabla f^k(x(k)) + E_o^T\alpha(k) - \frac{c}{2}E_u^T E_u[x(k-1) - x(k)] = 0. \tag{32}$$

From the definition of the unoriented Laplacian we can replace $(1/2)E_u^T E_u = L_u$ in (32) and further substitute $\phi(k)$ for its expression in (8). Since $E_o^T\alpha(k) = \phi(k) = \phi(k-1) + L_o x(k)$, we can write

$$\nabla f^k(x(k)) + \phi(k-1) + cL_o x(k) - cL_u[x(k-1) - x(k)] = 0. \tag{33}$$

The update for $x(k)$ in (8) follows from (33) by regrouping terms and observing that the degree matrix is $D = (1/2)(L_o + L_u)$. ■

Appendix B
Proof of Lemma 1

Proof: Proof of Lemma 1 is along the line of proving a similar lemma that appears in analyzing the decentralized static ADMM [22]. Start by observing that the initial condition $\alpha(0) = -\beta(0)$ guarantees that $\alpha(k) = -\beta(k)$ for all subsequent iterations $k \geq 0$. In such case the dynamic ADMM iterations in (5)–(7) can be proven to be equivalent to [cf. (28), (30), and (31)]

$$\nabla f^k(x(k)) + E_o^T\alpha(k) - cE_u^T[z(k-1) - z(k)] = 0, \tag{34}$$

$$\frac{1}{2}E_u x(k) = z(k), \tag{35}$$

$$\frac{c}{2}E_o x(k) = \alpha(k) - \alpha(k-1). \tag{36}$$

Since $\alpha(0)$ lies in the column space of E_o, from (36) we know that $\alpha(k)$ also lie in the column space of E_o for all subsequent iterations $k \geq 0$.

Recall now the definitions of the optimal primal variables $x^*(k)$ and $z^*(k)$ and the definition of the optimal multiplier $\lambda^*(k) = [\alpha^*(k); \beta^*(k)]$. Observe that the optimal primal variables $x^*(k)$ and $z^*(k)$ are unique because the primal functions $f^k(x)$ are strongly convex, but that there are more than one optimal multipliers $\lambda^*(k)$. However, there exists a unique optimal multiplier $\lambda^*(k) = [\alpha^*(k); \beta^*(k)]$ where $\alpha^*(k) = -\beta^*(k)$ lies in the column space of E_o. To see so write down the KKT conditions for the optimization problem in (4) so as to obtain the equalities

$$\nabla f^k(x^*(k)) + A^T\lambda^*(k) = 0, \tag{37}$$
$$B^T\lambda^*(k) = 0, \tag{38}$$
$$Ax^*(k) + Bz^*(k) = 0. \tag{39}$$

The definition of the matrix $B = [-I_{mp}; -I_{mp}]$ and the KKT condition in (38) imply that the optimal multiplier $\lambda^*(k) = [\alpha^*(k); \beta^*(k)]$ must satisfy $\alpha^*(k) = -\beta^*(k)$. Using this fact and the matrix definitions $A = [A_s; A_d]$ and $E_o = A_s - A_d$ we rewrite (37) as

$$\nabla f^k(x^*(k)) + E_o^T\alpha^*(k) = 0, \tag{40}$$

which implies that there exists $\alpha^*(k)$ lying in the column space of E_o. We prove uniqueness by contradiction. Consider two vectors $E_o r_1$ and $E_o r_2$ both lying in the column space of E_o where $r_1, r_2 \in \mathbb{R}^{np}$ and $E_o r_1 \neq E_o r_2$. If they both satisfy (40), i.e.

$$\begin{aligned}\nabla f^k(x^*(k)) + E_o^T E_o r_1 &= 0,\\ \nabla f^k(x^*(k)) + E_o^T E_o r_2 &= 0,\end{aligned} \tag{41}$$

then subtracting the two equalities in (41) yields

$$E_o^T E_o[r_1 - r_2] = 0. \tag{42}$$

Since $\|E_o^T E_o[r_1 - r_2]\| \geq \gamma_E\|E_o[r_1 - r_2]\|$ where γ_E is the smallest nonzero singular value of E_o, (42) implies that $\|E_o[r_1 - r_2]\| = 0$, which contradicts with $E_o r_1 \neq E_o r_2$. Since this is absurd we must have $E_o r_1 = E_o r_2$ implying that there is a unique optimal multiplier $\lambda^*(k) = [\alpha^*(k); \beta^*(k)]$ where $\alpha^*(k) = -\beta^*(k)$ lies in the column space of E_o.

We proceed now to manipulate (39) in order to obtain a set of equations similar to (35)-(36). Using again the definitions of the matrices $B = [-I_{mp}; -I_{mp}]$ and $A = [A_s; A_d]$ we can separate (39) in the two equations

$$\begin{aligned}A_s x^*(k) - z^*(k) &= 0,\\ A_d x^*(k) - z^*(k) &= 0.\end{aligned} \tag{43}$$

Summing up the two equalities in (43) and using the definition of the unoriented edge incidence matrix $E_u = A_s + A_d$ yields

$$\frac{1}{2}E_u x^*(k) = z^*(k). \tag{44}$$

Likewise, subtracting the two equalities in (43) and using the definition of the oriented edge incidence matrix $E_o = A_s - A_d$ yields

$$E_o x^*(k) = 0. \tag{45}$$

Subtracting (40) from (28) and reordering terms yields

$$\begin{aligned}\nabla f^k(x(k)) - \nabla f^k(x^*(k)) = cE_u^T[z(k-1) - z(k)]\\ -E_o^T[\alpha(k) - \alpha^*(k)].\end{aligned} \tag{46}$$

Subtracting (44) from (30) yields

$$\frac{1}{2}E_u[x(k) - x^*(k)] = z(k) - z^*(k). \tag{47}$$

Multiplying (45) by $c/2$ and subtracting the result from (31) yields

$$\frac{c}{2}E_o[x(k) - x^*(k)] = \alpha(k) - \alpha(k-1). \tag{48}$$

To complete the proof we use (46)–(48), the strong convexity inequality (11), and the Lipschitz continuity inequality (12). Next we split the proof of (14) into two steps.

Fact 1: The first step proves that

$$\begin{aligned}m_f\|x(k) - x^*(k)\|^2 \leq \|u(k-1) - u^*(k)\|_G^2\\ - \|u(k) - u^*(k)\|_G^2 - \|u(k-1) - u(k)\|_G^2.\end{aligned} \tag{49}$$

Proof: To prove that (49) holds true start by noting that according to Assumption 1 the primal objective functions $f^k(x)$ are strongly convex for all k with constant m_f. Replacing x_a by $x(k)$ and x_b by $x^*(k)$ in the strong convexity inequality (11) yields

$$\begin{aligned}m_f\|x(k) - x^*(k)\|^2\\ \leq [x(k) - x^*(k)]^T\left[\nabla f^k(x(k)) - \nabla f^k(x^*(k))\right].\end{aligned} \tag{50}$$

Substituting the gradient difference (46) into (50) leads to

$$\begin{aligned}m_f\|x(k) - x^*(k)\|^2 \leq [x(k) - x^*(k)]^T\left[cE_u^T(z(k-1)\right.\\ \left.-z(k)) - E_o^T(\alpha(k) - \alpha^*(k))\right].\end{aligned} \tag{51}$$

Expanding the right-hand side of (51) and rearranging terms in each of the resulting summands we can rewrite (51) as

$$\begin{aligned}m_f\|x(k)-x^*(k)\|^2 &\leq [x(k)-x^*(k)]^T\left[cE_u^T(z(k-1)-z(k))\right]\\ &\quad -[x(k)-x^*(k)]^T\left[E_o^T(\alpha(k)-\alpha^*(k))\right]\\ &= c\,[E_u(x(k)-x^*(k))]^T[z(k-1)-z(k)]\\ &\quad -[E_o(x(k)-x^*(k))]^T[\alpha(k)-\alpha^*(k)].\end{aligned} \tag{52}$$

Substituting (47) and (48) into (52) and rearranging terms, we have

$$\begin{aligned}m_f\|x(k) - x^*(k)\|^2 \leq 2c[z(k-1) - z(k)]^T[z(k) - z^*(k)]\\ +\frac{2}{c}[\alpha(k-1) - \alpha(k)]^T[\alpha(k) - \alpha^*(k)].\end{aligned} \tag{53}$$

Using the definitions of u and G we can rewrite (53) as

$$\begin{aligned}m_f\|x(k)-x^*(k)\|^2 &= 2[u(k-1)-u(k)]^T G[u(k)-u^*(k)]\\ &= \|u(k-1)-u^*(k)\|_G^2 - \|u(k)-u^*(k)\|_G^2\\ &\quad - \|u(k-1)-u(k)\|_G^2,\end{aligned} \tag{54}$$

which completes the proof of (49). ■

Fact 2: The second step proves that

$$\|u(k-1)-u(k)\|_G^2 + m_f\,\|x(k)-x^*(k)\|^2 \geq \delta\,\|u(k)-u^*(k)\|_G^2. \quad (55)$$

Proof: Expanding terms and using the definition of the Euclidean norm with respect to G it follows that the inequality in (56) is equivalent to

$$\frac{c}{\delta}\,\|z(k-1)-z(k)\|^2+\frac{1}{c\delta}\,\|\alpha(k-1)-\alpha(k)\|^2+\frac{m_f}{\delta}\,\|x(k)-x^*(k)\|^2 \geq c\,\|z(k)-z^*(k)\|^2+\frac{1}{c}\,\|\alpha(k)-\alpha^*(k)\|^2. \quad (56)$$

Computing the squared norm of both sides of (47) and using the definition of the unoriented Laplacian $L_u=(1/2)E_u^TE_u$ it follows that

$$\|z(k)-z^*(k)\|^2=\frac{1}{4}\,\|E_u\,[x(k)-x^*(k)]\|^2 = \frac{1}{2}\,[x(k)-x^*(k)]^T\,L_u\,[x(k)-x^*(k)]. \quad (57)$$

Further using the definition of Γ_L as the largest singular value of the unoriented Laplacian we can rewrite (57) as

$$\|z(k)-z^*(k)\|^2 \leq \frac{1}{2}\Gamma_L\,\|x(k)-x^*(k)\|^2. \quad (58)$$

As per Assumption 2 the primal objective has Lipschitz continuous gradients with Lipschitz constant upper bounded by M_f [cf. (12)]. Thus, replacing x_a by $x(k)$ and x_b by $x^*(k)$ in the Lipschitz continuity inequality (12) yields

$$M_f^2\,\|x(k)-x^*(k)\|^2 \geq \left\|\nabla f^k\,(x(k))-\nabla f^k\,(x^*(k))\right\|^2. \quad (59)$$

To further bound the optimality gap $\|x(k)-x^*(k)\|^2$ we use the gradient difference in (46). Observe that for any constant $\mu>0$ it holds $(\mu-1)\|a-b\|^2 \geq (1-1/\mu)\|b\|^2-\|a\|^2$. Identifying $a=cE_u^T[z(k)-z(k-1)]$ and $b=E_o^T[\alpha(k)-\alpha^*(k)]$ we can then conclude from (46) that for any $\mu>1$

$$(\mu-1)\left\|\nabla f^k\,(x(k))-\nabla f^k\,(x^*(k))\right\|^2 \geq \frac{\mu-1}{\mu}\left\|E_o^T\,[\alpha(k)-\alpha^*(k)]\right\|^2-\left\|cE_u^T\,[z(k-1)-z(k)]\right\|^2. \quad (60)$$

Observe that the first term in the right-hand side of (60) can be bounded in terms of the smallest nonzero eigenvalue γ_L of the oriented Laplacian $L_o=(1/2)E_o^TE_o$. Indeed, simply write the squared norm as the inner product $\|E_o^T[\alpha(k)-\alpha^*(k)]\|^2=[\alpha(k)-\alpha^*(k)]^TE_oE_o^T[\alpha(k)-\alpha^*(k)]$ where $E_oE_o^T$ have the same nonzero eigenvalues as $E_o^TE_o$ and observe that $\alpha(k)$ and $\alpha^*(k)$ both lie in the column space of E_o since we defined $\alpha^*(k)$ as the optimal multiplier lying in this column space and initialized $\alpha(0)$ to be in the span of E_o which guarantees that $\alpha(k)$ has this property for all k. It then follows from the definitions of L_o and γ_L that

$$\left\|E_o^T\,[\alpha(k)-\alpha^*(k)]\right\|^2 \geq 2\gamma_L\,\|\alpha(k)-\alpha^*(k)\|^2. \quad (61)$$

Likewise, the second term in the right-hand side of (60) can be bounded in terms of the largest eigenvalue Γ_L of the unoriented Laplacian $L_u=(1/2)E_u^TE_u$. Indeed, simply write the squared norm as the inner product $\|cE_u^T[z(k-1)-z(k)]\|^2=c^2[z(k-1)-z(k)]^TE_u^TE_u[z(k-1)-z(k)]$ and use the definitions of L_u and Γ_L to conclude that

$$\left\|cE_u^T\,[z(k-1)-z(k)]\right\|^2 \leq 2c^2\Gamma_L\,\|z(k-1)-z(k)\|^2. \quad (62)$$

Substituting the bound in (61) and (62) into (60) and the result into (59) establishes that for all constants $\mu>1$

$$(\mu-1)M_f^2\,\|x(k)-x^*(k)\|^2 \geq \frac{2\gamma_L(\mu-1)}{\mu}\,\|\alpha(k)-\alpha^*(k)\|^2 -2c^2\Gamma_L\,\|z(k-1)-z(k)\|^2. \quad (63)$$

Multiplying (58) by c and (63) by $\mu/[2c(\mu-1)\gamma_L]$ followed by summation of the resulting inequalities yields

$$\frac{\mu c\Gamma_L}{(\mu-1)\gamma_L}\,\|z(k-1)-z(k)\|^2+\left(\frac{c\Gamma_L}{2}+\frac{\mu M_f^2}{2c\gamma_L}\right)\|x(k)-x^*(k)\|^2 \geq c\,\|z(k)-z^*(k)\|^2+\frac{1}{c}\,\|\alpha(k)-\alpha^*(k)\|^2. \quad (64)$$

As per the definition of the contraction parameter δ in (13) we have $1/\delta \geq \mu c\Gamma_L/(\mu-1)\gamma_L$ and $1/\delta \geq c\Gamma_L/2+\mu M_f^2/2c\gamma_L$. Thus, it follows from (64) that

$$\frac{c}{\delta}\,\|z(k-1)-z(k)\|^2+\frac{m_f}{\delta}\,\|x(k)-x^*(k)\|^2 \geq c\,\|z(k)-z^*(k)\|^2+\frac{1}{c}\,\|\alpha(k)-\alpha^*(k)\|^2, \quad (65)$$

and consequently (567), which proves (55). ■

Combining (49) and (55) yields (14). ■

Appendix C
Proof of Lemma 2

Using the triangle inequality $\|u(k-1)-u^*(k)\|_G-\|u(k-1)-u^*(k-1)\|_G \leq \|u^*(k)-u^*(k-1)\|_G$ and the definition of $G:=\mathrm{diag}(cI_{mp},(1/c)I_{mp})$, we have

$$\begin{aligned}&\|u(k-1)-u^*(k)\|_G-\|u(k-1)-u^*(k-1)\|_G \quad (66)\\ &\leq \sqrt{c\|z^*(k)-z^*(k-1)\|^2+\frac{1}{c}\|\alpha^*(k)-\alpha^*(k-1)\|^2}\\ &\leq \sqrt{c}\|z^*(k)-z^*(k-1)\|+\frac{1}{\sqrt{c}}\|\alpha^*(k)-\alpha^*(k-1)\|.\end{aligned}$$

The right-hand side of (66) involves the drifts of z^* and α^*. We analyze the two drifts below. Recall that $\tilde{x}^*(k)$ is the unique optimal solution of (1) at time k. Due to the consensus constraints in (2), we have that $z^*(k)=e_m\otimes\tilde{x}^*(k)$ and $z^*(k-1)=e_m\otimes\tilde{x}^*(k-1)$ where $e_m\in\mathbb{R}^m$ denotes an all-one vector and $\otimes$ denotes the Kronecker product. Similarly, we have that $x^*(k)=e_n\otimes\tilde{x}^*(k)$ and $x^*(k-1)=e_n\otimes\tilde{x}^*(k-1)$. Therefore, it holds

$$\|z^*(k)-z^*(k-1)\|=\frac{\sqrt{m}}{\sqrt{n}}\|x^*(k)-x^*(k-1)\|. \quad (67)$$

From KKT conditions of (4) we know that (cf. (40))

$$\nabla f^k(x^*(k))+E_o^T\alpha^*(k)=0. \quad (68)$$

Therefore $\nabla f^k(x^*(k)) - \nabla f^{k-1}(x^*(k-1)) = E_o^T[\alpha^*(k) - \alpha^*(k-1)]$ for all $k \geq 0$ and consequently

$$\begin{aligned}&\|E_o^T[\alpha^*(k) - \alpha^*(k-1)]\| \\ &\quad = \|\nabla f^k(x^*(k)) - \nabla f^{k-1}(x^*(k-1))\|. \quad (69)\end{aligned}$$

Note again that we only consider $\alpha^*(k)$ and $\alpha^*(k-1)$ which lie in the column space of E_o. Expanding the squared norm $\|E_o^T[\alpha^*(k) - \alpha^*(k-1)]\|^2 = [\alpha^*(k) - \alpha^*(k-1)]^T E_o E_o^T [\alpha^*(k) - \alpha^*(k-1)]$ and observing that γ_L is the smallest nonzero eigenvalue of $L_o = (1/2)E_o^T E_o$ as well as the smallest nonzero eigenvalue of $(1/2)E_o E_o^T$, the vector $\alpha^*(k) - \alpha^*(k-1)$ lying in the column space of E_o guarantees that $\|\alpha^*(k) - \alpha^*(k-1)\| \leq (1/\sqrt{2\gamma_L})\|E_o^T[\alpha^*(k) - \alpha^*(k-1)]\|$ and hence

$$\begin{aligned}&Vert\alpha^*(k) - \alpha^*(k-1)\| \\ &\quad \leq \frac{1}{\sqrt{2\gamma_L}}\|\nabla f^k(x^*(k)) - \nabla f^{k-1}(x^*(k-1))\|. \quad (70)\end{aligned}$$

Substituting (67) and (70) to (66), the gap between $\|u(k-1) - u^*(k)\|_G$ and $\|u(k-1) - u^*(k-1)\|_G$ is bounded by $g(k)$ defined in , which completes the proof.

Appendix D
Proof of Theorem 3

Proof: We characterize the limit property of $\|x(k) - x^*(k)\|$ based on (20) in Theorem 1 and (21) in Theorem 2. From (20), we know that

$$\begin{aligned}&\|u(0) - u^*(0)\|_G + g(1) \geq \sqrt{1+\delta}\,\|u(1) - u^*(1)\|_G\,, \\ &\quad \ldots, \|u(k-2) - u^*(k-2)\|_G + g(k-1) \\ &\quad \geq \sqrt{1+\delta}\,\|u(k-1) - u^*(k-1)\|_G\,. \quad (71)\end{aligned}$$

Multiplying the row in (71) with $g(s)$ by $\sqrt{1+\delta}^{(s-1)}$ and summing all the rows up, we have

$$\begin{aligned}&\|u(0) - u^*(0)\|_G + \sum_{s=1}^{k-1}\sqrt{1+\delta}^{(s-1)}g(s) \\ &\quad \geq \sqrt{1+\delta}^{(k-1)}\,\|u(k-1) - u^*(k-1)\|_G\,. \quad (72)\end{aligned}$$

Adding $\sqrt{1+\delta}^{(k-1)}g(k)$ to both sides, (72) becomes

$$\begin{aligned}&\|u(0) - u^*(0)\|_G + \sum_{s=1}^{k}\sqrt{1+\delta}^{(s-1)}g(s) \\ &\quad \geq \sqrt{1+\delta}^{(k-1)}\,[\|u(k-1) - u^*(k-1)\|_G + g(k)]\,. \quad (73)\end{aligned}$$

Combining (73) with $\|u(k-1) - u^*(k-1)\|_G + g(k) \geq \sqrt{m_f}\|x(k) - x^*(k)\|$ in (21) leads to

$$\begin{aligned}&\|u(0) - u^*(0)\|_G + \sum_{s=1}^{k}\sqrt{1+\delta}^{(s-1)}g(s) \\ &\quad \geq \sqrt{1+\delta}^{(k-1)}\sqrt{m_f}\,\|x(k) - x^*(k)\|\,, \quad (74)\end{aligned}$$

or equivalently

$$\begin{aligned}&\frac{1}{\sqrt{1+\delta}^{(k-1)}}\,\|u(0) - u^*(0)\|_G + \sum_{s=1}^{k}\frac{1}{\sqrt{1+\delta}^{(k-s)}}g(s) \\ &\quad \geq \sqrt{m_f}\,\|x(k) - x^*(k)\|\,. \quad (75)\end{aligned}$$

Since $g(s) \leq g_{max}$ for $s = 1, \ldots, k$ and $\delta > 0$, using the summation formula of a geometric series

$$\begin{aligned}\sum_{s=1}^{k}\frac{1}{\sqrt{1+\delta}^{(k-s)}}g(s) &\leq \sum_{s=1}^{k}\frac{1}{\sqrt{1+\delta}^{(k-s)}}g_{max} \\ &= \frac{1-\sqrt{1+\delta}^{(-k)}}{1-\sqrt{1+\delta}^{(-1)}}g_{max}. \quad (76)\end{aligned}$$

Putting together (74) and (75), we know

$$\begin{aligned}&\frac{1}{\sqrt{1+\delta}^{(k-1)}}\,\|u(0) - u^*(0)\|_G + \frac{1-\sqrt{1+\delta}^{(-k)}}{1-\sqrt{1+\delta}^{(-1)}}g_{max} \\ &\quad \geq \sqrt{m_f}\,\|x(k) - x^*(k)\|\,. \quad (77)\end{aligned}$$

Taking $k \to +\infty$, the first term in the left-hand side of (77) vanishes while the second term reaches a limit value $g_{max}/(1 - \sqrt{1+\delta}^{(-1)})$. Therefore we obtain (23) that proves Theorem 3. ■

References

[1] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," in *Proc. SPAWC*, 2013, pp. 170–174.

[2] F. Jakubiec and A. Ribeiro, "D-MAP: Distributed maximum *a posteriori* probability estimation of dynamic systems," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 450–466, 2013.

[3] R. Cavalcante and S. Stanczak, "A distributed subgradient method for dynamic convex optimization problems under noisy information exchange," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 243–256, 2013.

[4] K. Zhou and S. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 678–695, 2010.

[5] S. Tu and A. Sayed, "Mobile adaptive networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 649–664, 2011.

[6] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, 2009.

[7] S. Ram, A. Nedic, and V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optimiz. Theory Appl.*, vol. 147, pp. 516–545, 2010.

[8] D. Jakovetic, J. Xavier, and J. Moura, "Fast distributed gradient methods," [Online]. Available: http://arxiv.org/pdf/1112.2972v3.pdf

[9] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, 2012.

[10] K. Tsianos and M. Rabbat, "Distributed dual averaging for convex optimization under communication delays," in *Proc. ACC*, 2012, pp. 1067–1072.

[11] M. Rabbat, R. Nowak, and J. Bucklew, "Generalized consensus computation in networked systems with erasure links," in *Proc. SPAWC*, 2005, pp. 1088–1092.

[12] M. Zargham, A. Ribeiro, A. Jadbabaie, and A. Ozdaglar, "Accelerated dual descent for network optimization," [Online]. Available: http://www.seas.upenn.edu/~aribeiro/preprints/2013_zargham_etal.pdf

[13] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in *ad hoc* WSNs with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, 2008.

[14] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, 2010.

[15] Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3816–3827, 2010.

[16] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5523–5537, Nov. 2011.

[17] J. Mota, J. Xavier, P. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1942–1956, Apr. 2012.

[18] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. New York, NY, USA: Athena Scientific, 1997.

[19] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," [Online]. Available: http://www.optimization-online.org/DB_FILE/2012/08/3578.pdf
[20] M. Hong and Z. Luo, "On the linear convergence of the alternating direction method of multipliers," [Online]. Available: http://arxiv.org/pdf/1208.3922v3.pdf
[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, 2010.
[22] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "Linearly convergent decentralized consensus optimization with the alternating direction method of multipliers," in *Proc. ICASSP*, 2013, pp. 4613–4617.
[23] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, pp. 215–233, 2007.
[24] S. Kar and J. Moura, "Gossip and distributed Kalman filtering: Weak consensus under weak detectability," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1766–1784, 2011.
[25] P. Braca, S. Marano, V. Matta, and P. Willet, "Asymptotic optimality of running consensus in testing binary hypothesis," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 814–825, 2010.
[26] D. Bajovic, D. Jakovetic, J. Xavier, B. Sinopoli, and J. Moura, "Distributed detection via Gaussian running consensus: Large deviations asymptotic analysis," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4381–4396, 2011.
[27] F. Cattivelli and A. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, 2010.

Qing Ling received the B.Sc. degree in automation and the Ph.D. degree in control theory and control engineering from University of Science and Technology of China, Hefei, in 2001 and 2006, respectively.

From 2006 to 2009, he was a Post-Doctoral Research Fellow at the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI, USA. Since 2009, he has been an Associate Professor at the Department of Automation, University of Science and Technology of China, Hefei. His current research focuses on decentralized optimization of networked multiagent systems.

Alejandro Ribeiro received the B.Sc. degree in electrical engineering from the Universidad de la Republica Oriental del Uruguay, Montevideo, in 1998 and the M.Sc. and Ph.D. degrees in electrical engineering from the Department of Electrical and Computer Engineering, the University of Minnesota, Minneapolis, MN, USA, in 2005 and 2007, respectively.

From 1998 to 2003, he was a member of the technical staff at Bellsouth Montevideo. After his M.Sc. and Ph.D. studies, in 2008, he joined the University of Pennsylvania, Philadelphia, PA. USA, where he is currently an Assistant Professor at the Department of Electrical and Systems Engineering. His research interests are in the applications of statistical signal processing to the study of networks and networked phenomena. His current research focuses on wireless networks, network optimization, learning in networks, networked control, robot teams, and structured representations of networked data structures.

Dr. Ribeiro received the 2012 S. Reid Warren, Jr. Award presented by the University of Pennsylvania's undergraduate student body for outstanding teaching and the NSF CAREER Award in 2010. He is also a Fulbright scholar and the recipient of student paper awards at the 2013 American Control Conference (as adviser), as well as the 2005 and 2006 International Conferences on Acoustics, Speech and Signal Processing.