A Universal Decomposition for Distributed Optimization Algorithms

Bryan Van Scoy* Laurent Lessard†

Abstract

In the distributed optimization problem for a multi-agent system, each agent knows a local function and must find a minimizer of the sum of all agents' local functions by performing a combination of local gradient evaluations and communicating information with neighboring agents. We prove that every distributed optimization algorithm can be factored into a centralized optimization method and a second-order consensus estimator, effectively separating the "optimization" and "consensus" tasks. We illustrate this fact by providing the decomposition for many recently proposed distributed optimization algorithms. Conversely, we prove that any optimization method that converges in the centralized setting can be combined with any second-order consensus estimator to form a distributed optimization algorithm that converges in the multi-agent setting. Finally, we describe how our decomposition may lead to a more systematic algorithm design methodology.

1 Introduction

We consider the distributed optimization problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} & x_1 = x_2 = \ldots = x_n, \end{array}$$

where f_i is the local objective function and x_i the local decision variable on agent $i \in \{1,\ldots,n\}$. The problem is to minimize the sum of the local objective functions subject to agreement among the agents on the solution, where each agent $i \in \{1,\ldots,n\}$ can evaluate its local gradient ∇f_i and can communicate with (and only with) neighboring agents[1].

Many distributed algorithms have been proposed in the literature, and several recent works have attempted to uncover an underlying algorithmic structure. The work [1] developed a framework that unified the EXTRA [2] and DIGing [3] algorithms. The work [4] found a canonical form for distributed algorithms that encompasses cases where each agent has two state variables. This canonical form, however, was limited to non-accelerated algorithms. To handle acceleration, Han [5]

*B. Van Scoy is with the Department of Electrical and Computer Engineering, Miami University, OH 45056, USA. Email bvanscoy@miamioh.edu

†L. Lessard is with the Department of Mechanical and Industrial Engineering, Northeastern University, MA 02115, USA. Email l.lessard@northeastern.edu

[1] Some authors refer to this as the *consensus optimization problem* and to such algorithms as *decentralized*.

showed that an optimization method could be combined with two first-order consensus estimators to form a valid distributed optimization algorithm. This structure captures many algorithms, but not all algorithms decompose into this form. This is particularly important when the structure is used for design, as a limited structure may lead to suboptimal performance. For example, the structure in [5] is unable to represent the SVL algorithm [6], which is an optimized implementation of inexact ADMM (see Section 3.2.2).

Our main result overcomes the aforementioned limitations and shows that a broad class of distributed optimization algorithm can be decomposed into a centralized optimization method and a second-order consensus estimator as shown in Figure 1. Specifically, our decomposition applies to algorithms that are linear time-invariant (LTI) systems in feedback with the gradient of the objective function and the graph Laplacian. Conversely, we show that any centralized optimization method can be combined with any second-order consensus estimator to form a distributed optimization algorithm (under mild technical conditions).

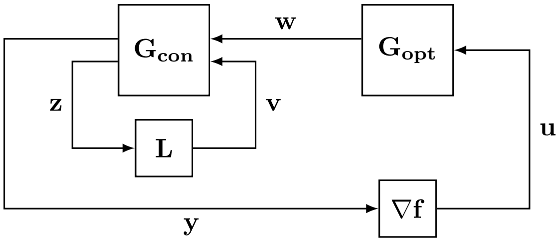


Fig. 1: Universal decomposition of a distributed optimization algorithm into an optimization method $\mathbf{G_{opt}}$ and second-order consensus estimator $\mathbf{G_{con}}$, where $\nabla\mathbf{f}$ is the gradient of the local objective functions and $\mathbf{L}$ is the graph Laplacian.

Our decomposition has several benefits. First, it provides a non-conservative parameterization of distributed optimization algorithms in terms of their components, which can then be systematically analyzed using tools from robust control; see [6–8] for the details of such analyses. Our decomposition also assists algorithm designers by simplifying the taxonomy of distributed optimization algorithms. Simply put, one need not look any further than the already vast literature on gradient-based optimization methods [9–11] and consensus estimators [12–14].

Notation. Subscripts denote an agent's index, and bold symbols to refer to quantities aggregated over all agents, such as $\mathbf{x} = (x_1, \dots, x_n)$. Superscripts denote time indices, as $\{y^0, y^1, \dots\}$. Symbols $\mathbf{1}$ and $\mathbf{0}$ denote the n-dimensional vector of all ones and all zeros, respectively. The symbol $\otimes$ denotes the Kronecker product. For an LTI operator G, the corresponding transfer function is $\widehat{G}(z)$. A transfer function is *stable* if all of its poles are in the open unit disk.

Assumption. To simplify notation, we assume the local objective functions are one-dimensional, $f_i : \mathbb{R} \to \mathbb{R}$. Our results generalize to the multidimensional case $f_i : \mathbb{R}^d \to \mathbb{R}$ under appropriate restrictions on the algorithm form.

2 Preliminaries

Before describing algorithms for distributed optimization, we first describe optimization and consensus separately. We make extensive use of the Final Value Theorem (FVT), which we state here for completeness (e.g., [15, pp. 2-12, 2-15]).

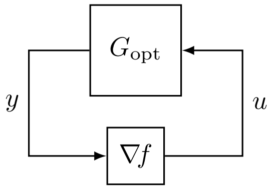
Proposition 1 (Final Value Theorem)**.** *Suppose y^t has the unilateral z-transform $\hat{y}(z)$. The following are equivalent.*

- *The limit of y^t as $t \to \infty$ exists and is finite.*
- *$(z-1)\,\hat{y}(z)$ is stable.*

If the above hold, then $\lim_{t\to\infty} y^t = \lim_{z\to 1}(z-1)\,\hat{y}(z)$.

2.1 Optimization

A gradient-based optimization method is an iterative procedure used to find an extremum of some function f by sequentially querying ∇f. We can view such a method as a discrete-time dynamical system G_{opt} in feedback with ∇f [7].



$$y = G_{\mathrm{opt}}\, u$$
$$u = \nabla f(y)$$

For example, standard gradient descent uses the update $x^{t+1} = x^t - \alpha\,\nabla f(x^t)$, for which G_{opt} can be represented using the discrete-time transfer function $\widehat{G}_{\mathrm{opt}}(z) = \frac{-\alpha}{z-1}$. Methods such as gradient descent are *strictly causal* and have strictly proper transfer functions.

If a method is causal but not strictly causal, then the feedback loop has a circular dependency. An example of such an algorithm is the *proximal point method*, which uses the update $x^{t+1} \in \arg\min_x \bigl(f(x) + \frac{1}{2\alpha}\|x - x_t\|^2\bigr)$. The circular dependency is apparent when we write the associated first-order optimality condition: $x^{t+1} = x^t - \alpha\,\nabla f(x^{t+1})$. This method has a proper transfer function: $\widehat{G}_{\mathrm{opt}}(z) = \frac{-\alpha z}{z-1}$.

In this letter, we define an *optimization method* as a system G_{opt} that has the correct fixed point when placed in feedback with ∇f and also exhibits convergent behavior when using a baseline set of *easy* test functions f.

Definition 1. *Consider the feedback interconnection of a system G_{opt} with the gradient ∇f, where $f(y) := \frac{\varepsilon}{2}\|y - y^\star\|^2$. The system G_{opt} is an* optimization method *if for all $\varepsilon > 0$ sufficiently small and for all $y^\star$, we have $y^t \to y^\star$ as $t \to \infty$, and y^t converges to a constant when $\varepsilon = 0$.*

If G_{opt} is causal, SISO, and LTI as with gradient descent and many other methods, we can characterize optimization methods via properties of the transfer function $\widehat{G}_{\mathrm{opt}}(z)$.

Lemma 1. *A causal SISO LTI system G_{opt} is an optimization method if and only if the following hold:*

(i) The zeros of $1 - \varepsilon\, \widehat{G}_{\text{opt}}(z)$ are inside the unit circle for all $\varepsilon > 0$ sufficiently small.

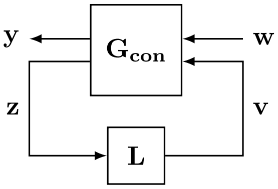
(ii) $\widehat{G}_{\text{opt}}(z)$ has a pole at $z = 1$ and $(z-1)\, \widehat{G}_{\text{opt}}(z)$ is stable.

(iii) $\widehat{G}_{\text{opt}}(z)$ is proper.

Proof. Substituting the given f and eliminating u, the closed-loop dynamics are $y = \frac{-\varepsilon G_{\text{opt}}}{1-\varepsilon G_{\text{opt}}} y^\star$. Applying the FVT, y^t converging is equivalent to stability of the map and the zeros of $1 - \varepsilon\, \widehat{G}_{\text{opt}}(z)$ being inside the unit circle. The limit $y^t \to y^\star$ is equivalent to $\widehat{G}_{\text{opt}}(z)$ having a pole at $z = 1$. For $\varepsilon = 0$, convergence to a constant is equivalent to stability of $(z-1)\, \widehat{G}_{\text{opt}}(z)$. Causality of the system G_{opt} is equivalent to properness of the transfer function $\widehat{G}_{\text{opt}}(z)$. ∎

2.2 Consensus

Consider a network of n agents. Agent $i \in \{1, \ldots, n\}$ observes a time-varying signal w_i. A consensus estimator [12, 16] is an iterative procedure where each agent communicates with its neighbors in order to form an estimate y_i of the average $\frac{1}{n} \sum_{i=1}^{n} w_i$. Such estimators take the following form.



$$\begin{bmatrix} y_i \\ z_i \end{bmatrix} = G_{\text{con}} \begin{bmatrix} w_i \\ v_i \end{bmatrix}$$

$$v_i = \sum_{j=1}^{n} a_{ij} \left(z_i - z_j \right)$$

The $n \times n$ matrix $A := [a_{ij}]$ is the *adjacency matrix* that describes the interaction among agents. The scalar a_{ij} is the weight that agent i places on information from agent j, with a weight of zero if no information flows from agent j to i. Agent j is a *neighbor* of agent i if the weight a_{ij} is nonzero, and computing v_i requires agent i to receive the local variables z_j from each of its neighbors j. The *Laplacian* is the matrix $L := \operatorname{diag}(A\mathbf{1}) - A$. This matrix always satisfies $L\mathbf{1} = \mathbf{0}$, so it has an eigenvalue of zero with corresponding eigenvector $\mathbf{1}$. When the communication network is *connected*, meaning that there is a path between any two agents, there is exactly one zero eigenvalue [12]. When the weights are constructed such that $L^\mathsf{T}\mathbf{1} = \mathbf{0}$, the Laplacian is *balanced* [16]. The block diagram illustrates the global behavior of the system aggregated over all agents, where the aggregated system and Laplacian are

$$\mathbf{G_{con}} := \begin{bmatrix} I_n \otimes G_{\text{con}}^{11} & I_n \otimes G_{\text{con}}^{12} \\ I_n \otimes G_{\text{con}}^{21} & I_n \otimes G_{\text{con}}^{22} \end{bmatrix} \quad \text{and} \quad \mathbf{L} := L \otimes I_m,$$

where m is the dimension of the local vectors v_i and z_i. For example, one particular (first-order) consensus estimator is given by the iterations $\mathbf{x}^{t+1} = \mathbf{x}^t + L\left(\mathbf{w}^t - \mathbf{x}^t\right)$ and $\mathbf{y}^t = \mathbf{w}^t - \mathbf{x}^t$, for which G_{con} can be represented using the discrete-time transfer function

$$\widehat{G}_{\text{con}}(z) = \begin{bmatrix} 1 & \frac{-1}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix}. \tag{1}$$

We are interested in tracking signals with constant mean but potentially *higher-order* deviations from the mean. We define a first-order estimator to have zero steady-state error for constant deviations from the mean, a second-order estimator for ramp deviations, and so on. Similar to the motivation for our definition of optimization methods, we define a *consensus estimator* as a system G_{con} that can successfully track the average when using a baseline set of *easy* Laplacians L.

Definition 2. *A system G_{con} is a* consensus estimator *of order ℓ if, for any connected communication network and associated balanced and diagonalizable Laplacian L with spectral radius sufficiently small and for any signals w_i^t that are polynomials in t of degree $\ell-1$ with constant mean $w^\star := \frac{1}{n}\sum_{j=1}^n w_j^t$, the estimate y_i^t on each agent i converges to the average $w^\star$ as $t \to \infty$.*

If G_{con} is causal and LTI, we can characterize consensus estimators via properties of the transfer function $\widehat{G}_{\text{con}}(z)$. It is also typical to assume G_{con}^{22} is strictly causal to avoid circular dependencies in the network transmissions.

Lemma 2. *Suppose G_{con} is a causal LTI system and G_{con}^{22} is strictly causal. For all complex $\lambda \in \mathbb{C}$, define the map $G_\lambda := G_{\text{con}}^{11} + \lambda\, G_{\text{con}}^{12}\big(I - \lambda\, G_{\text{con}}^{22}\big)^{-1} G_{\text{con}}^{21}$. Then, G_{con} is a consensus estimator of order ℓ if and only if the following hold:*

(i) $\widehat{G}_\lambda(z)$ is stable for all $\lambda \in \mathbb{C}$ satisfying $|\lambda| < \delta$ for some $\delta > 0$ sufficiently small.

(ii) $\widehat{G}_0(1) = 1$.

(iii) $\widehat{G}_\lambda(z)$ has ℓ zeros at $z = 1$ for all $\lambda \neq 0$.

(iv) $\widehat{G}_{\text{con}}(z)$ is proper and $\widehat{G}_{\text{con}}^{22}(z)$ is strictly proper.

Proof. We can write the error $e_i^t := y_i^t - w^\star$ succinctly as $\mathbf{e} = \big(\mathbf{G_L} - \frac{1}{n}\mathbf{1}\mathbf{1}^\mathsf{T}\big)\,\mathbf{w}$, where $\mathbf{G_L}$ is the closed-loop map from $\mathbf{w}$ to $\mathbf{y}$. Let (λ, v^T) be a left eigen-pair of L. Using that $L\mathbf{1} = \mathbf{0}$, we have that $0 = v^\mathsf{T} L \mathbf{1} = \lambda\,(v^\mathsf{T}\mathbf{1})$. Thus, $\mathbf{1}^\mathsf{T} v = 0$ for all v corresponding to nonzero λ. Furthermore, $L^\mathsf{T}\mathbf{1} = \mathbf{0}$ since the Laplacian is balanced (by assumption). The inner product of an eigenvector with the error is then

$$v^\mathsf{T}\mathbf{e} = \begin{cases} (G_0 - 1)(\mathbf{1}^\mathsf{T}\mathbf{w}) & \text{if } v = \mathbf{1} \text{ and } \lambda = 0 \\ G_\lambda(v^\mathsf{T}\mathbf{w}) & \text{otherwise.} \end{cases}$$

Since the Laplacian is diagonalizable (by assumption), convergence of the error $\mathbf{e}$ is equivalent to convergence of $v^\mathsf{T}\mathbf{e}$ for each eigenvector v. Applying the FVT in the case where w_i^t are polynomials in t of degree $\ell - 1$ with constant average $w^\star$, the limit $e_i^t \to 0$ is equivalent to stability of $\widehat{G}_\lambda$ for all eigenvalues λ of the Laplacian L and

$$\lim_{z\to 1}\ \widehat{G}_0(z) - 1 = 0 \quad \text{and} \quad \lim_{z \to 1}\ \frac{\widehat{G}_\lambda(z)}{(z-1)^{\ell-1}} = 0,$$

which correspond to the first three conditions. Causality of G_{con} and strict causality of G_{con}^{22} are equivalent to properness and strict properness of $\widehat{G}_{\text{con}}(z)$ and $\widehat{G}_{\text{con}}^{22}(z)$, respectively. ∎

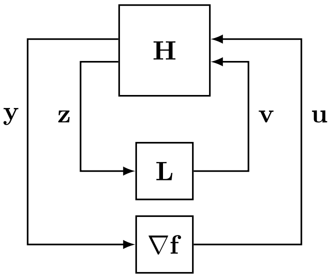
Remark 1. *The transfer function of a consensus estimator is not unique. Let $\widehat{F}$ be any $m \times m$ transfer matrix with full normal rank, where m is the dimension of v_i and z_i. Then the closed-loop map $\mathbf{G_L} : \mathbf{w} \mapsto \mathbf{y}$ is invariant under*

$$\widehat{G}_{\text{con}} \mapsto \begin{bmatrix} 1 & 0 \\ 0 & \widehat{F} \end{bmatrix} \widehat{G}_{\text{con}} \begin{bmatrix} 1 & 0 \\ 0 & \widehat{F}^{-1} \end{bmatrix},$$

although not all choices of $\widehat{F}$ preserve causality of G_{con}.

2.3 Distributed optimization

The distributed optimization setting is conceptually a combination of the optimization and consensus settings. There are n agents that can communicate over a network, agent i has access to the gradient of its local function ∇f_i, and the goal is for all agents to achieve consensus on an extremum of the sum of all functions $f_1 + \cdots + f_n$. Distributed optimization algorithms take the following general form [6, 8]; see Section 3.2 for specific examples from the literature.



$$\begin{bmatrix} y_i \\ z_i \end{bmatrix} = H \begin{bmatrix} u_i \\ v_i \end{bmatrix},$$
$$u_i = \nabla f_i(y_i),$$
$$v_i = \sum_{j=1}^{n} a_{ij} \left(z_i - z_j \right)$$

Similar to the consensus and optimization settings, we have

$$\mathbf{H} := \begin{bmatrix} I_n \otimes H^{11} & I_n \otimes H^{12} \\ I_n \otimes H^{21} & I_n \otimes H^{22} \end{bmatrix}, \qquad \begin{aligned} \mathbf{L} &:= L \otimes I_m, \\ \nabla \mathbf{f} &:= \operatorname{diag}(\nabla f_1, \ldots, \nabla f_n). \end{aligned}$$

We define a distributed optimization algorithm as follows.

Definition 3. *A system H is a* distributed optimization algorithm *if for any connected communication network and associated balanced Laplacian L with spectral radius sufficiently small, and for all $\varepsilon > 0$ sufficiently small and for all $y_i^\star$, the feedback interconnection of $\mathbf{H}$ with $\mathbf{L}$ and $\nabla \mathbf{f}$ satisfies $y_i^t \to y^\star := \frac{1}{n} \sum_{j=1}^n y_j^\star$ as $t \to \infty$ for all i, where $f_i(y) := \frac{\varepsilon}{2} \|y - y_i^\star\|^2$. We also require that all y_i^t converge to a common constant limit when $f_i \equiv 0$ for all i.*

If H is causal and LTI, we can characterize consensus estimators via properties of the transfer function $\widehat{H}(z)$. We will also assume causality of certain maps to ensure that the algorithm is implementable. In particular, H should be causal, and there should be no circular dependencies in the network transmissions or gradient evaluations. This means that the partial closed-loop map $H^{22} + \varepsilon H^{21} (I - \varepsilon H^{11})^{-1} H^{12}$ should be strictly causal, which is equivalent to both H^{22} and $H^{21} H^{12}$ being strictly causal.

Lemma 3. *Suppose H is a causal LTI system, and H^{22} and $H^{21} H^{12}$ are strictly causal. For all $\lambda \in \mathbb{C}$, define the map*

$$H_\lambda := H^{11} + \lambda H^{12} \big(I - \lambda H^{22} \big)^{-1} H^{21}.$$

The system H is a distributed optimization algorithm if and only if the following hold:

(i) The zeros of $1 - \varepsilon \widehat{H}_\lambda(z)$ are inside the unit circle for all $\varepsilon > 0$ sufficiently small and for all $\lambda \in \mathbb{C}$ satisfying $|\lambda| < \delta$ for some $\delta > 0$ sufficiently small.

(ii) $\widehat{H}_0(z)$ has a pole at $z = 1$ and $(z-1)\, \widehat{H}_0(z)$ is stable.

(iii) $\widehat{H}_\lambda(z)$ is stable and has a zero at $z = 1$ for all $\lambda \neq 0$.

(iv) $\widehat{H}(z)$ is proper and both $\widehat{H}^{22}(z)$ and $\widehat{H}^{21}(z)\, \widehat{H}^{12}(z)$ are strictly proper.

Proof. Let $\mathbf{H_L}$ be the partial closed-loop map from $\mathbf{u}$ to $\mathbf{y}$ after we eliminate $\mathbf{z}$ and $\mathbf{v}$. Substituting the given f_i and eliminating $\mathbf{u}$, we obtain the closed-loop dynamics $\mathbf{y} = -\varepsilon \mathbf{H_L}(I - \varepsilon \mathbf{H_L})^{-1}\mathbf{y}^\star$. The condition $y_i^t \to \frac{1}{n}\sum_{j=1}^n y_j^\star$ can be written succinctly as $\mathbf{y}^t \to \frac{1}{n}\mathbf{1}\mathbf{1}^\mathsf{T}\mathbf{y}^\star$. Diagonalizing the closed-loop dynamics as in the proof of Lemma 2 and applying the FVT, we find that y_i^t converging is equivalent to the map $-\varepsilon \widehat{H}_\lambda(z)(I - \varepsilon \widehat{H}_\lambda(z))^{-1}$ being stable, which is equivalent to (i). Again from the FVT, $\mathbf{1}^\mathsf{T}\mathbf{y}^t \to \mathbf{1}^\mathsf{T}\mathbf{y}^\star$ means $H_0(z)$ has a pole at $z = 1$, and convergence to a constant in the case $f_i \equiv 0$ means $(z-1)\,\widehat{H}_0(z)$ is stable, so we have (ii). As in the proof of Lemma 2, we have $v^\mathsf{T}\mathbf{1} = 0$ and $v^\mathsf{T}\mathbf{y}^t \to 0$ for all v corresponding to $\lambda \neq 0$, so $\widehat{H}_\lambda(z)$ has a zero at $z = 1$, and for the case $f_i \equiv 0$, we have that $\widehat{H}_\lambda(z)$ is stable, which is equivalent to (iii). Item (iv) is equivalent to the causality assumptions. ∎

3 Universal decomposition

We now state our main result, which states that every distributed optimization algorithm can be decomposed into consensus and optimization components as in Figure 1.

Theorem 1. *Let H be a distributed optimization algorithm satisfying the conditions of Lemma 3. There exists an optimization method G_opt and a second-order consensus estimator G_con such that*

$$H = G_\text{con}\begin{bmatrix} G_\text{opt} & 0 \\ 0 & I_m \end{bmatrix}. \tag{2}$$

If H^{11} is strictly causal, then G_opt can be chosen to be strictly causal as well.

Proof. From conditions (i), (ii), and (iv) of Lemma 3, $\widehat{H}_0(z)$ has a pole at $z = 1$ and is proper, $(z-1)\,\widehat{H}_0(z)$ is stable, and the zeros of $1 - \varepsilon\,\widehat{H}_0(z)$ are inside the unit circle for all $\varepsilon > 0$ sufficiently small. Then from Lemma 1, $H_0 = H^{11}$ is an optimization method. If $\widehat{H}^{11}(z)$ is non-minimum phase (has zeros on or outside the unit circle), then $\widehat{G}_\text{opt}(z) := z^p \prod \left(\frac{1-\bar{z}_0 z}{z - z_0}\right)\widehat{H}^{11}(z)$, where the product is over all such zeros z_0, will also satisfy the conditions of Lemma 1, provided p is at most the relative degree of $\widehat{H}^{11}(z)$. This follows because $\widehat{G}_\text{opt}(z)$ is still proper, still has a pole at $z = 1$, and because each factor multiplying $\widehat{H}^{11}(z)$ is an all-pass filter with nonnegative phase (phase lead), which therefore can only increase stability margins and preserves the stability requirement.

Set $\widehat{G}_\text{opt}(z) = z^p\,\widehat{\Phi}(z)\,\widehat{H}^{11}(z)$, where $\widehat{\Phi}(z) := \prod \frac{1-\bar{z}_0 z}{z-z_0}$ is the product of all-pass factors that cancel the non-minimum phase zeros of $\widehat{H}^{11}(z)$. Then, invert the transformation (2) and apply Remark 1 using $\widehat{F}(z) = z^{-q}I$ to obtain

$$\widehat{G}_\text{con}(z) = \begin{bmatrix} z^{-p}\,\widehat{\Phi}(z)^{-1} & z^q\,\widehat{H}^{12}(z) \\ z^{-p-q}\,\widehat{H}^{21}(z)\,\widehat{H}^{11}(z)^{-1}\,\widehat{\Phi}(z)^{-1} & \widehat{H}^{22}(z) \end{bmatrix}.$$

Since $\widehat{H}^{11}(z)$ is proper and $\widehat{H}^{21}(z)\,\widehat{H}^{12}(z)$ is strictly proper, we can always ensure $\widehat{G}_\text{con}(z)$ will be proper by letting p and q be the relative degrees of $\widehat{H}^{11}(z)$ and $\widehat{H}^{12}(z)$, respectively. This choice leads to a $\widehat{G}_\text{opt}(z)$ that has relative degree zero. However, when $\widehat{H}^{11}(z)$ is strictly proper, we can reduce p by 1, which ensures that $\widehat{G}_\text{opt}(z)$ is strictly proper as well.

To verify that G_con is a consensus estimator of order two, we can compute G_λ as defined in Lemma 2 and see that $\widehat{G}_\lambda(z) = \left(z^p\widehat{H}^{11}(z)\,\widehat{\Phi}(z)\right)^{-1}\widehat{H}_\lambda(z)$, where H_λ is defined in Lemma 3. We can now verify the properties in Lemma 2. When $\lambda = 0$, the transfer function is $\widehat{G}_0(z) = z^{-p}\,\widehat{\Phi}(z)^{-1}$, which is stable and satisfies $\widehat{G}_0(1) = 1$ since $\widehat{\Phi}$ is all-pass. When $\lambda \neq 0$, the fact that $\widehat{H}_\lambda(z)$ has a zero at $z = 1$ and $\widehat{H}_0(z) = \widehat{H}^{11}(z)$ has a pole at $z = 1$ implies that $\widehat{G}_\lambda(z)$ has two zeros at $z = 1$. To verify stability when $\lambda \neq 0$, stability of $\widehat{G}_\lambda(z)$ follows from stability of $\widehat{H}_\lambda(z)$ and $\widehat{\Phi}(z)^{-1}$. ■

We can also prove a partial converse; under certain mild technical conditions, combining consensus and optimization components as in Figure 1 yields a distributed optimization algorithm.

Theorem 2. *Suppose G_{opt} is a causal SISO LTI optimization method, G_{con} is a causal LTI second-order consensus estimator, and further assume*

- *$\widehat{G}_{\text{opt}}(z)$ and $\widehat{G}_\lambda(z)$ are minimum-phase, meaning all zeros are strictly inside the unit circle, and*
- *either $\widehat{G}_{\text{opt}}(z)$ or $\widehat{G}^{21}_{\text{con}}(z)\,\widehat{G}^{12}_{\text{con}}(z)$ is strictly proper.*

Then, the combined system H given in (2) *is a distributed optimization algorithm.*

Proof. We will verify the properties of Lemma 3. Since $H_\lambda = G_\lambda\, G_{\text{opt}}$ and G_λ is stable, $\widehat{G}_{\text{opt}}(z)$ has a single pole at $z = 1$ and there are no zeros outside the unit circle, the root locus will be stable for small gains, so (i) holds. When $\lambda = 0$, $\widehat{H}_0(z) = \widehat{G}_0(z)\,\widehat{G}_{\text{opt}}(z)$. Since $\widehat{G}_0(1) = 1$ and $\widehat{G}_{\text{opt}}(z)$ has a pole at $z = 1$ and $(z-1)\widehat{G}_{\text{opt}}(z)$ is stable, we have (ii). When $\lambda \neq 0$, $\widehat{G}_\lambda(z)$ has two zeros at $z = 1$ and $\widehat{G}_0(z)$ has a single pole at $z = 1$, therefore $\widehat{H}_\lambda(z)$ has a zero at $z = 1$ and (iii) holds. Now we examine properness. Note that $\widehat{H}^{21}(z)\,\widehat{H}^{12}(z) = \widehat{G}^{21}_{\text{con}}(z)\,\widehat{G}^{12}_{\text{con}}(z)\,\widehat{G}_{\text{opt}}(z)$, so strict properness of either term on the right-hand side implies strict properness of the left-hand side. Finally, properness of $\widehat{G}_{\text{con}}(z)$ and $\widehat{G}_{\text{opt}}(z)$ imply properness of $\widehat{H}(z)$, and strict properness of $\widehat{G}^{22}_{\text{con}}(z)$ implies strict properness of $\widehat{H}^{22}(z)$, so (iv) holds. ■

Remark 2. *The continuous-time analog of gradient-based optimization methods are called* gradient flows*, and there has been recent interest in studying iterative algorithms in the continuous limit [17]. Likewise, consensus methods are often analyzed in continuous time [12]. The decomposition described in Theorems 1–2 was developed for discrete-time distributed optimization algorithms, but an analogous decomposition exists for continuous-time systems. In this case, a distributed optimization algorithm would separate into a gradient flow and a continuous-time consensus estimator.*

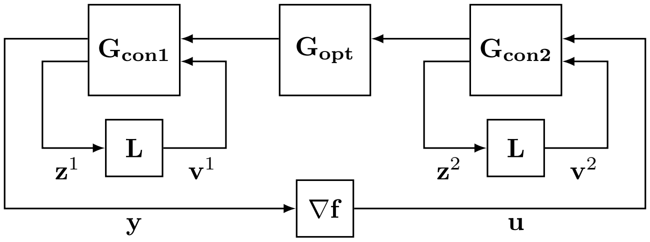


Fig. 2: Factored form of an algorithm, where the second-order consensus estimator factors into two first-order SISO estimators; this is the form proposed in [5].

3.1 Factoring the consensus estimator

The decomposition in Figure 1 is not internally stable. While the average gradient is zero at the optimizer, the gradient of each agent is not necessarily zero. This nonzero constant is integrated by the optimization method to produce an unbounded output. This can be fixed, however, if the consensus estimator *factors* into two first-order estimators.

Suppose G_{con} factors as $G_{\text{con1}}G_{\text{con2}}$, where G_{con1} and G_{con2} are both first-order estimators. The optimization method and both consensus estimators are SISO LTI systems and therefore commute, so we can swap the order of G_{opt} and G_{con2} to obtain the diagram in Figure 2. While this does not change the map from $\mathbf{u}$ to $\mathbf{y}$, it does change the realization; the steady-state input to the optimization method is now the average gradient, which is zero at optimality.

To check whether or not a consensus estimator factors, we equate a second-order estimator G_{con} with its factorization $G_{\text{con1}}G_{\text{con2}}$ to find that

$$G_{\text{con}} = \left[\begin{array}{c:cc} G^{11}_{\text{con1}}\,G^{11}_{\text{con2}} & G^{12}_{\text{con1}} & G^{11}_{\text{con1}}\,G^{12}_{\text{con2}} \\ \hdashline G^{21}_{\text{con1}}\,G^{11}_{\text{con2}} & G^{22}_{\text{con1}} & G^{21}_{\text{con1}}\,G^{12}_{\text{con2}} \\ G^{21}_{\text{con2}} & 0 & G^{22}_{\text{con2}} \end{array}\right],$$

where (z^1, v^1) are the transmitted and received variables for G_{con1}, and similarly for G_{con2}. The inputs to the combined system are then (u, v^1, v^2), and the outputs are (y, z^1, z^2). Note that the transmitted variables v^1 and v^2 need not have the same dimension. The consensus estimator has this form if and only if G^{32}_{con} is zero and its components factor as

$$\begin{bmatrix} G^{11}_{\text{con}} & G^{13}_{\text{con}} \\ G^{21}_{\text{con}} & G^{23}_{\text{con}} \end{bmatrix} = \begin{bmatrix} G^{11}_{\text{con1}} \\ G^{21}_{\text{con1}} \end{bmatrix} \begin{bmatrix} G^{11}_{\text{con2}} & G^{12}_{\text{con2}} \end{bmatrix},$$

which is the case if and only if $G^{11}_{\text{con}}\,G^{23}_{\text{con}} - G^{13}_{\text{con}}\,G^{21}_{\text{con}} = 0$. Whether an estimator factors or not depends on the transfer function G_{con} which is not unique, so we may need to first apply the transformation in Remark 1 with a suitable transfer function $\widehat{F}$ for an estimator to factor.

3.2 Decomposition of known algorithms

To illustrate our results, we first describe our decomposition technique on a well-known distributed optimization algorithm. We then state the decomposition for many other algorithms from the literature.

3.2.1 DIGing

We first illustrate our results on the DIGing algorithm [3, 18], which is described by the iterations

$$\begin{aligned} \mathbf{x}^{t+1} &= W\mathbf{x}^t - \alpha\,\mathbf{y}^t, \\ \mathbf{y}^{t+1} &= W\mathbf{y}^t + \nabla\mathbf{f}(\mathbf{x}^{t+1}) - \nabla\mathbf{f}(\mathbf{x}^t), \end{aligned}$$

where $\alpha > 0$ is the stepsize and the gossip matrix W is related to the graph Laplacian as $W = I - L$. This algorithm requires each agent to communicate $m = 2$ variables at each iteration, and the associated transfer function is

$$\widehat{H}(z) = \begin{bmatrix} \frac{-\alpha}{z-1} & \frac{-z}{z-1} & \frac{-\alpha z}{(z-1)^2} \\ \frac{-\alpha}{z\,(z-1)} & \frac{-1}{z-1} & \frac{\alpha}{(z-1)^2} \\ \frac{1}{z} & 0 & \frac{-1}{z-1} \end{bmatrix}.$$

Choose the optimization method as $\widehat{G}_{\text{opt}}(z) = \widehat{H}^{11}(z) = \frac{-\alpha}{z-1}$. Then applying the transformation in Remark 1 with the transfer matrix $\widehat{F}(z) = \text{diag}\big(z, \frac{-\alpha z}{z-1}\big)$, the consensus estimator transforms as

$$\widehat{G}_{\text{con}}(z) = \begin{bmatrix} 1 & \frac{-z}{z-1} & \frac{\alpha z}{(z-1)^2} \\ \frac{1}{z} & \frac{-1}{z-1} & \frac{\alpha}{(z-1)^2} \\ \frac{z-1}{-\alpha z} & 0 & \frac{-1}{z-1} \end{bmatrix} \mapsto \begin{bmatrix} 1 & \frac{-1}{z-1} & \frac{-1}{z-1} \\ 1 & \frac{-1}{z-1} & \frac{-1}{z-1} \\ 1 & 0 & \frac{-1}{z-1} \end{bmatrix}.$$

The estimator on the right satisfies the conditions to factor in Sec. 3.1; we chose the transformation matrix such that this is the case. Since $G_{\text{con}}^{11} = 1$, we can choose $G_{\text{con1}}^{11} = 1 = G_{\text{con2}}^{11}$, which results in the factorization $G_{\text{con}} = G_{\text{con1}}\, G_{\text{con2}}$, where both factors are the first-order estimator in (1).

The analysis for all other algorithms in this section is similar. In each case, we choose the optimization algorithm as $G_{\text{opt}} = H^{11}$ so that $G_{\text{con}}^{11} = 1$. In addition, we apply the transformation in Remark 1 to put the estimators in a similar form with $G_{\text{con}}^{21} = 1$ for comparison.

3.2.2 Non-accelerated algorithms

We first consider algorithms that use standard gradient descent for the optimization method: $\widehat{G}_{\text{opt}}(z) = \frac{-\alpha}{z-1}$ where $\alpha > 0$ is the stepsize. Several such algorithms have been proposed whose consensus estimator factors (see Section 3.1). In particular, each factor is typically one of the following first-order estimators:

$$\widehat{G}_{\text{con1}}(z),\, \widehat{G}_{\text{con2}}(z) = \begin{bmatrix} 1 & \frac{-1}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & \frac{-z}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix}.$$

Every combination of these factors has been proposed in the literature: DIGing [3, 18] uses the estimator on the left for both factors, $\mathcal{AB}$ [19] uses one of each[2], and AugDGM [20] uses the one on the right for both factors.

Not every algorithm uses a consensus estimator that factors into two first-order estimators. To check whether or not an algorithm factors, we search for a transfer matrix $\widehat{F}$ with full normal rank such that the transformed consensus estimator in Remark 1 satisfies the necessary conditions for factorization in Section 3.1. Here are the second-order consensus estimators for some algorithms

[2] The $\mathcal{AB}$ method is described in terms of two gossip matrices $\mathcal{A}$ and $\mathcal{B}$, where the Laplacian is $L = I - \mathcal{A} = I - \mathcal{B}$.

that do not factor:

$$\widehat{G}_{\text{con}}(z) = \begin{bmatrix} 1 & \frac{-\frac{1}{2}z^2}{(z-1)^2} \\ 1 & \frac{\frac{1}{2}-z}{(z-1)^2} \end{bmatrix} \qquad \text{Exact Diffusion [21]}$$

$$\widehat{G}_{\text{con}}(z) = \begin{bmatrix} 1 & \frac{-\frac{1}{2}z^2}{(z-1)^2} \\ 1 & \frac{-\frac{1}{2}+z-z^2}{(z-1)^2} \end{bmatrix} \qquad \text{NIDS [22]}$$

$$\widehat{G}_{\text{con}}(z) = \begin{bmatrix} 1 & \frac{\frac{1}{2}-z}{(z-1)^2} \\ 1 & \frac{\frac{1}{2}-z}{(z-1)^2} \end{bmatrix} \qquad \text{EXTRA [2]}$$

$$\widehat{G}_{\text{con}}(z) = \begin{bmatrix} 1 & \frac{-z(z+\beta-1)}{(z-1)^2} \\ 1 & \frac{1-(1+\beta)z}{(z-1)^2} \end{bmatrix} \qquad \text{SVL [6]}$$

3.2.3 Accelerated algorithms

Our decomposition also applies to accelerated algorithms. The optimization method then has the form [7, 9]

$$\widehat{G}_{\text{opt}}(z) = -\alpha \, \frac{(1+\gamma)\, z - \gamma}{(z-1)(z-\beta)},$$

where β and γ are additional parameters. Examples include $\mathcal{AB}m$ [23] based on the heavy-ball optimization method [11] with $\gamma = 0$, and $\mathcal{ABN}$ [24] based on Nesterov's accelerated method [10] with $\gamma = \beta$. For each of these algorithms, the consensus estimator factors into the two first-order estimators

$$\widehat{G}_{\text{con1}}(z) = \begin{bmatrix} 1 & \frac{1}{\alpha}\widehat{G}_{\text{opt}}(z) \\ 1 & \frac{1}{\alpha}\widehat{G}_{\text{opt}}(z) \end{bmatrix} \quad \text{and} \quad \widehat{G}_{\text{con2}}(z) = \begin{bmatrix} 1 & \frac{-1}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix}.$$

4 Perspectives

Our decomposition of an algorithm into its optimization and consensus components leads to some perspectives that may prove useful for algorithm design.

Robust optimization Using our decomposition, we can interpret an algorithm for distributed optimization as an optimization method that, along with the gradient, includes an additional consensus estimator in the loop. If this consensus estimator were to converge arbitrarily fast, then the iterates would never be in disagreement and the system would reduce to that of the centralized optimization method. Because the consensus estimator is not ideal, however, the optimization method must be *robust* to the dynamics of the estimator; see [25–28] for robust optimization methods.

Consensus with feedback Alternatively, we can view an algorithm as a second-order consensus estimator whose input is obtained by feeding back the output through the gradient and the optimization method. In this interpretation, the consensus estimator must be stable when connected in feedback. This feedback loop is linear when the local objective functions are quadratic (gradients are linear), but is otherwise *nonlinear*.

Each of these interpretations provides a certain perspective on the combined algorithm. Ideally, the design of the optimization and consensus components would decouple, enabling researchers to make use of the abundant literature on optimization and consensus. Our decomposition provides a first step towards this decoupling, with these perspectives indicating that proper measures of robustness must be taken into account in the algorithm design.

References

[1] D. Jakovetić, "A unification and generalization of exact distributed first-order methods," *IEEE Trans. Sig. Inf. Process. Netw.*, vol. 5, no. 1, pp. 31–46, 2018. (Cited on p. 1)

[2] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015. (Cited on pp. 1 and 11)

[3] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017. (Cited on pp. 1, 9, and 10)

[4] A. Sundararajan, B. Van Scoy, and L. Lessard, "A canonical form for first-order distributed optimization algorithms," in *Amer. Contr. Conf.*, 2019, pp. 4075–4080. (Cited on p. 1)

[5] S. Han, "Systematic design of decentralized algorithms for consensus optimization," *IEEE Contr. Syst. Lett.*, vol. 3, no. 4, pp. 966–971, 2019. (Cited on pp. 1, 2, and 8)

[6] A. Sundararajan, B. Van Scoy, and L. Lessard, "Analysis and design of first-order distributed optimization algorithms over time-varying graphs," *IEEE Trans. Contr. Netw. Syst.*, vol. 7, no. 4, pp. 1597–1608, 2020. (Cited on pp. 2, 6, and 11)

[7] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM J. Optim.*, vol. 26, no. 1, pp. 57–95, 2016. (Cited on pp. 2, 3, and 11)

[8] A. Sundararajan, B. Hu, and L. Lessard, "Robust convergence analysis of distributed optimization algorithms," in *Allerton Conf. Commun. Contr. Comput.*, 2017, pp. 1206–1212. (Cited on pp. 2 and 6)

[9] B. Van Scoy, R. A. Freeman, and K. M. Lynch, "The fastest known globally convergent first-order method for minimizing strongly convex functions," *IEEE Contr. Syst. Lett.*, vol. 2, no. 1, pp. 49–54, 2018. (Cited on pp. 2 and 11)

[10] Y. Nesterov, *Lectures on Convex Optimization.* Springer Optimization and Its Applications, 2018, vol. 137. (Cited on pp. 2 and 11)

[11] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, 1964. (Cited on pp. 2 and 11)

[12] S. S. Kia, B. Van Scoy, J. Cortés, R. A. Freeman, K. M. Lynch, and S. Martínez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Contr. Syst. Mag.*, vol. 39, no. 3, pp. 40–72, 2019. (Cited on pp. 2, 4, and 8)

[13] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, pp. 322–329, 2010. (Cited on p. 2)

[14] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control. Lett.*, vol. 53, no. 1, pp. 65–78, 2004. (Cited on p. 2)

[15] W. S. Levine, *The Control Handbook (three volume set).* CRC press, 2018, vol. 1. (Cited on p. 3)

[16] R. A. Freeman, T. R. Nelson, and K. M. Lynch, “A complete characterization of a class of robust linear average consensus protocols,” in *Amer. Contr. Conf.*, 2010, pp. 3198–3203. (Cited on p. 4)

[17] W. Su, S. Boyd, and E. Candès, “A differential equation for modeling Nesterov’s accelerated gradient method: Theory and insights,” *Adv. Neur. Inf. Process. Syst.*, vol. 27, 2014. (Cited on p. 8)

[18] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” *IEEE Trans. Contr. Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, 2017. (Cited on pp. 9 and 10)

[19] R. Xin and U. A. Khan, “A linear algorithm for optimization over directed graphs with geometric convergence,” *IEEE Contr. Syst. Lett.*, vol. 2, no. 3, pp. 315–320, 2018. (Cited on p. 10)

[20] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, “Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes,” in *IEEE Conf. Decis. Contr.*, 2015, pp. 2055–2060. (Cited on p. 10)

[21] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, “Exact diffusion for distributed optimization and learning—Part I: Algorithm development,” *IEEE Trans. Sig. Process.*, vol. 67, no. 3, pp. 708–723, 2018. (Cited on p. 11)

[22] Z. Li, W. Shi, and M. Yan, “A decentralized proximal-gradient method with network independent stepsizes and separated convergence rates,” *IEEE Trans. Sig. Process.*, vol. 67, no. 17, pp. 4494–4506, 2019. (Cited on p. 11)

[23] R. Xin and U. A. Khan, “Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking,” *IEEE Trans. Automat. Contr.*, vol. 65, no. 6, pp. 2627–2633, 2020. (Cited on p. 11)

[24] R. Xin, D. Jakovetić, and U. A. Khan, “Distributed Nesterov gradient methods over arbitrary graphs,” *IEEE Sig. Process. Lett.*, vol. 26, no. 8, pp. 1247–1251, 2019. (Cited on p. 11)

[25] S. Cyrus, B. Hu, B. Van Scoy, and L. Lessard, “A robust accelerated optimization algorithm for strongly convex functions,” in *Amer. Contr. Conf.*, Jun. 2018, pp. 1376–1381. (Cited on p. 11)

[26] N. S. Aybat, A. Fallah, M. Gürbüzbalaban, and A. Ozdaglar, “Robust accelerated gradient methods for smooth strongly convex functions,” *SIAM J. Optim.*, vol. 30, no. 1, pp. 717–751, 2020. (Cited on p. 11)

[27] S. Michalowsky, C. Scherer, and C. Ebenbauer, “Robust and structure exploiting optimisation algorithms: an integral quadratic constraint approach,” *Int. J. Contr.*, vol. 94, no. 11, pp. 2956–2979, 2020. (Cited on p. 11)

[28] H. Mohammadi, M. Razaviyayn, and M. R. Jovanović, “Robustness of accelerated first-order algorithms for strongly convex optimization problems,” *IEEE Trans. Automat. Contr.*, vol. 66, no. 6, pp. 2480–2495, 2021. (Cited on p. 11)