

## Control Methods for Distributed Optimization

# Distributed gradient methods for consensus optimization

Prof. Ivano Notarnicola

Dept. of Electrical, Electronic, Information Eng.  
Università di Bologna  
ivano.notarnicola@unibo.it

Prof. Ruggero Carli

Dept. of Information Engineering  
Università di Padova  
ruggero.carli@unipd.it

**SIDRA Ph.D. Summer School**  
July, 10-12 2025 • Bertinoro, Italy

# Lecture outline

---

- The gradient method for consensus optimization
- Distributed gradient methods for consensus optimization
- Accelerated distributed gradient methods for consensus optimization

# Consensus optimization (recall)

A *consensus optimization* (scalar) problem is

$$\min_{\mathbf{x} \in \mathbb{R}} \sum_{i=1}^N f_i(\mathbf{x})$$

where each  $f_i : \mathbb{R} \rightarrow \mathbb{R}$  is strongly convex and has Lipschitz continuous gradient

Recalling  $f(x) := \sum_{i=1}^N f_i(x_i)$  with  $x := (x_1, \dots, x_N)$ , the gradient method expressed as

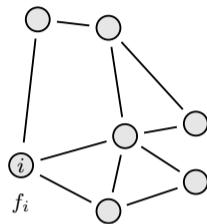
$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{1}^\top \nabla f(\mathbf{1} \mathbf{x}_k)$$

where  $\alpha > 0$  is the stepsize and  $\mathbf{1} \in \mathbb{R}^N$  is the all-one vector

The gradient method can be *replicated*  $N$  times to obtain a parallel algorithm given by

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1} \\ \vdots \\ \mathbf{x}_{k+1} \end{bmatrix} = J \mathbf{x}_k - \alpha J \nabla f(J \mathbf{x}_k)$$

where  $\mathbf{x}_k \in \mathbb{R}^N$  and  $J := \frac{1}{N} \mathbf{1} \mathbf{1}^\top$  (factorize  $\frac{1}{N}$  from  $\alpha$ )



# Frequency-domain characterization of the parallel gradient method

Consider the parallel gradient method

$$\begin{aligned}x_{k+1} &= Jx_k - \alpha J u_k \\ y_k &= Jx_k\end{aligned}$$

in feedback with  $u_k = \nabla f(y_k)$ . The transfer matrix from  $u_k$  to  $y_k$  is given by

$$G(z) = -\alpha J(zI - J)^{-1}J = -\frac{\alpha}{z-1}J$$

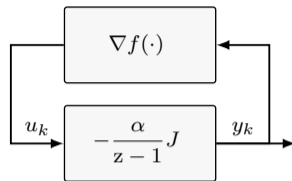
where we used the fact that  $J$  can be jointly diagonalized<sup>1</sup> with  $I$ , and hence the following identities hold

$$(zI - J)^{-1} = T \begin{bmatrix} \frac{1}{z-1} & \\ & \frac{1}{z} I_{N-1} \end{bmatrix} T^{-1}, \quad (zI - J)^{-1}J = T \begin{bmatrix} \frac{1}{z-1} & \\ & \frac{1}{z} I_{N-1} \end{bmatrix} \begin{bmatrix} 1 & \\ & 0_{N-1} \end{bmatrix} T^{-1} = \frac{1}{z-1}J$$

**Remark.** One integrator in the direction of  $\mathbf{1}$  and  $N-1$  *deadbeat* dynamics in its orthogonal complement

**Remark.** The gradient operator  $\nabla f : \mathbb{R}^N \rightarrow \mathbb{R}^N$  satisfies

$$(\nabla f(y) - \nabla f(\mathbf{1}x_*))^\top (y - \mathbf{1}x_*) \geq \frac{1}{\mu+L} \|\nabla f(y) - \nabla f(\mathbf{1}x_*)\|^2 + \frac{\mu L}{\mu+L} \|y - \mathbf{1}x_*\|^2$$



<sup>1</sup>It holds  $J = T \begin{bmatrix} 1 & \\ & 0_{N-1} \end{bmatrix} T^{-1}$  where the first row of  $T$  is  $\mathbf{1}^\top$  and the other  $N-1$  rows complete an orthonormal basis

# Towards a distributed gradient method for consensus optimization

Consider the replicated gradient method

$$\begin{aligned}x_{k+1} &= Jx_k - \alpha Ju_k, & x_0 &= \mathbf{1}x^0 \\ y_k &= Jx_k\end{aligned}$$

in feedback with  $u_k = \nabla f(y_k)$

**Remark.** The static map  $J$  is crucial to enforce *consensus* and compute the correct *average* descent direction

**Remark.** The algorithm is not amenable to distributed implementation due to the aggregating terms

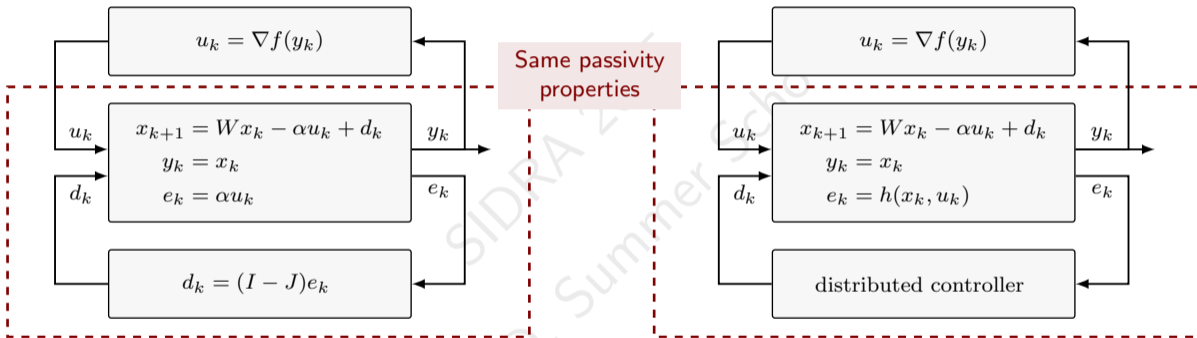
Consider the following slightly modified scheme

$$\begin{aligned}x_{k+1} &= Wx_k - \alpha \nabla f(y_k) + \underbrace{\alpha(I - J)\nabla f(y_k)}_{d_k}, & x_0 &= x^0 \\ y_k &= x_k\end{aligned}$$

where  $W \in \mathbb{R}^{N \times N}$  is a symmetric doubly-stochastic matrix, and  $d_k$  is a *correction term*

# Unleashing distributed consensus optimization

**Idea.** Compensate for the (centralized and static) correction term  $d_k$  using a *distributed* and *dynamic* controller



Two alternative strategies

1. *dynamic average consensus* to track the average of the gradients based on  $e_k := \alpha u_k$
2. *integral action* to reject the consensus error  $e_k := (I - W)x_k$

# Strategy 1: distributed gradient method based on dynamic average consensus

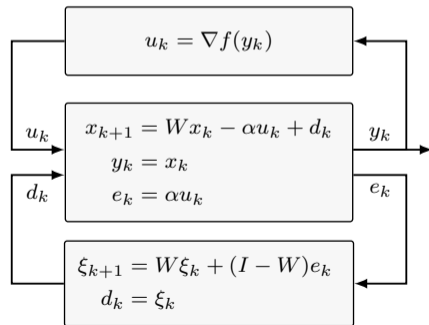
Consider the reference signal be  $e_k := \alpha u_k = \alpha \nabla f(x_k)$ , the *dynamic average consensus* reads

$$\begin{aligned}\xi_{k+1} &= W\xi_k + (I - W)e_k, & \xi_0 &= 0_N \\ d_k &= \xi_k\end{aligned}$$

Then, the closed-loop system results in

$$\begin{aligned}\begin{bmatrix} x_{k+1} \\ \xi_{k+1} \end{bmatrix} &= \begin{bmatrix} W & I \\ 0 & W \end{bmatrix} \begin{bmatrix} x_k \\ \xi_k \end{bmatrix} - \alpha \begin{bmatrix} I \\ W - I \end{bmatrix} \nabla f(y_k), & \begin{bmatrix} x_0 \\ \xi_0 \end{bmatrix} &= \begin{bmatrix} x^0 \\ 0_N \end{bmatrix} \\ y_k &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \xi_k \end{bmatrix}\end{aligned}$$

**Remark.** The initialization is *not* arbitrary



# Agent perspective of dynamic-average-consensus-based gradient tracking

Each agent  $i$  implements the following local updates

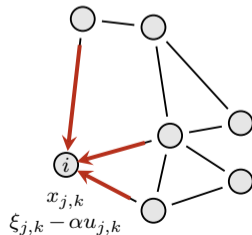
$$x_{i,k+1} = \sum_{j \in N_i} w_{ij} x_{j,k} + \xi_{i,k} - \alpha u_{i,k}, \quad x_{i,0} \in \mathbb{R}$$

$$\xi_{i,k+1} = \sum_{j \in N_i} w_{ij} \xi_{j,k} + \alpha \left( u_{i,k} - \sum_{j \in N_i} w_{ij} u_{j,k} \right), \quad \xi_{i,0} = 0$$

$$y_{i,k} = x_{i,k}$$

with  $u_{i,k} = \nabla f_i(y_{i,k})$ , where  $w_{ij}$  are the entries of  $W$  and  $N_i$  is the neighbor set of agent  $i$

**Remark.** Each agent  $i$  receives  $x_{j,k}$  and  $\xi_{j,k} - \alpha u_{j,k}$  from its neighbors  $j \in N_i$



# Equilibrium manifold for the gradient tracking

Consider the gradient tracking

$$\begin{bmatrix} x_{k+1} \\ \xi_{k+1} \end{bmatrix} = \begin{bmatrix} W & I \\ 0 & W \end{bmatrix} \begin{bmatrix} x_k \\ \xi_k \end{bmatrix} - \alpha \begin{bmatrix} I \\ W - I \end{bmatrix} \nabla f(x_k), \quad \begin{bmatrix} x_0 \\ \xi_0 \end{bmatrix} = \begin{bmatrix} x^0 \\ 0_N \end{bmatrix}$$

The equilibrium point  $(x_{\text{eq}}, \xi_{\text{eq}})$  satisfies

$$\begin{aligned} (I - W)x_{\text{eq}} &= \xi_{\text{eq}} - \alpha \nabla f(x_{\text{eq}}) \\ (I - W)\xi_{\text{eq}} &= \alpha(I - W)\nabla f(x_{\text{eq}}) \end{aligned}$$

Premultiplying the first equation by  $\mathbf{1}^\top$  yields

$$\mathbf{1}^\top \xi_{\text{eq}} - \alpha \mathbf{1}^\top \nabla f(x_{\text{eq}}) = 0$$

which, combined with the second equation and the mass preservation property, results in

$$\xi_{\text{eq}} = \alpha \nabla f(x_{\text{eq}})$$

Thus it must be  $(I - W)x_{\text{eq}} = 0_N$ . Using the invariance of  $\mathbf{1}^\top \xi_k - \alpha \mathbf{1}^\top \nabla f(x_k)$ , implies that the equilibrium is

$$\begin{bmatrix} x_{\text{eq}} \\ \xi_{\text{eq}} \end{bmatrix} = \begin{bmatrix} \mathbf{1}x_\star \\ \alpha \nabla f(\mathbf{1}x_\star) \end{bmatrix}$$

# The transfer matrix of the gradient tracking

The evolution in the error coordinates  $(x, \xi) \mapsto (\tilde{x}, \tilde{\xi}) := (x - \mathbf{1}x_*, \xi - \alpha \nabla f(\mathbf{1}x_*))$  is

$$\begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{\xi}_{k+1} \end{bmatrix} = \begin{bmatrix} W & I \\ 0 & W \end{bmatrix} \begin{bmatrix} \tilde{x}_k \\ \tilde{\xi}_k \end{bmatrix} - \alpha \begin{bmatrix} I \\ W - I \end{bmatrix} \tilde{u}_k, \quad \begin{bmatrix} \tilde{x}_0 \\ \tilde{\xi}_0 \end{bmatrix} = \begin{bmatrix} x^0 - \mathbf{1}x_* \\ -\nabla f(\mathbf{1}x_*) \end{bmatrix}$$

$$\tilde{y}_k := y_k - \mathbf{1}x_* = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_k \\ \tilde{\xi}_k \end{bmatrix}$$

with  $\tilde{u}_k := \nabla f(\tilde{y}_k + \mathbf{1}x_*) - \nabla f(\mathbf{1}x_*)$

The transfer matrix from  $\tilde{u}_k$  to  $\tilde{y}_k$  is given by

$$\begin{aligned} G(z) &= C(zI - A)^{-1}B = -\alpha \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} zI - W & -I \\ 0 & zI - W \end{bmatrix}^{-1} \begin{bmatrix} I \\ W - I \end{bmatrix} \\ &= -\alpha \begin{bmatrix} (zI - W)^{-1} & (zI - W)^{-2} \end{bmatrix} \begin{bmatrix} I \\ W - I \end{bmatrix} \\ &= -\alpha (zI - W)^{-1} \left( I + (zI - W)^{-1}(W - I) \right) \\ &= -\alpha (z - 1)(zI - W)^{-2} \end{aligned}$$

**Remark.** Do the parallel, i.e.,  $-\frac{\alpha}{z-1}J$ , and the GT algorithms share the same passivity properties?

# Gradient tracking analysis: first loop transformation

First, actuate a (positive) feedback action on the plant

$$\tilde{u} = \hat{u} + \mu \tilde{y}$$

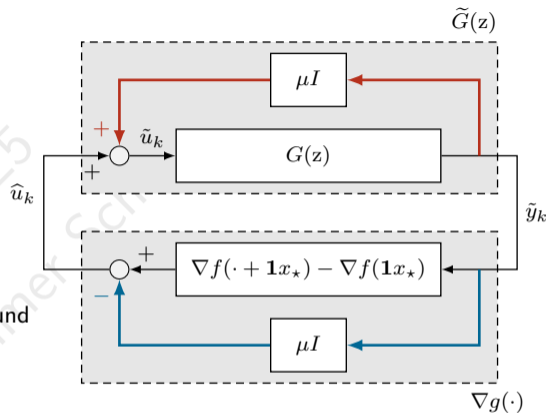
“stealing” strong convexity from the cost function  $f$

The transfer function from  $\hat{u}_k$  to  $\tilde{y}_k$  is

$$\tilde{G}(z) = (I - \mu G(z))^{-1} G(z)$$

with  $\hat{u}_k = \tilde{u}_k - \mu \tilde{y}_k$  and  $\tilde{y}_k$  satisfying the co-coercivity bound

$$\hat{u}_k^\top \tilde{y}_k \geq \frac{1}{L - \mu} \|\hat{u}_k\|^2$$



# Gradient tracking analysis: second loop transformation

Second, consider a *feedforward action* on the plant

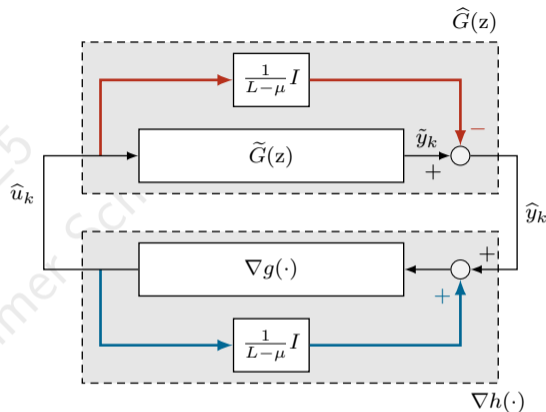
$$\tilde{y} \mapsto \hat{y} = \tilde{y}_k - \frac{1}{L-\mu} \hat{u}$$

The transfer function from  $\hat{u}_k$  to  $\hat{y}_k$  is

$$\hat{G}(z) = \tilde{G}(z) - \frac{1}{L-\mu} I_N$$

with  $\hat{u}_k$  and  $\hat{y}_k$  satisfying the monotonicity bound

$$\hat{u}_k^\top \hat{y}_k \geq 0$$



**Proposition.** For a sufficiently small  $\alpha$  the transfer function  $-\hat{G}(z)$  is *strictly discrete positive real*

Invoking the Passivity theorem, one can show that the origin  $\tilde{x} = 0_N$  is an exponentially stable equilibrium, i.e.,

$$\lim_{k \rightarrow \infty} \|x_k - \mathbf{1}x_\star\| = 0 \quad \text{at a linear rate}$$

# Change of coordinates to obtain the “non-causal” formulation

The gradient tracking is

$$\begin{aligned}x_{k+1} &= Wx_k + \xi_k - \alpha \nabla f(x_k), & x_0 &= x^0 \\ \xi_{k+1} &= W\xi_k - \alpha(W - I)\nabla f(x_k), & \xi_0 &= 0_N\end{aligned}$$

Consider the (nonlinear) change of coordinates

$$\begin{bmatrix} x_k \\ \xi_k \end{bmatrix} \mapsto \begin{bmatrix} x_k \\ s_k \end{bmatrix} := \begin{bmatrix} x_k \\ -\frac{1}{\alpha}\xi_k + \nabla f(x_k) \end{bmatrix}$$

Then, the gradient tracking can be rewritten as

$$\begin{aligned}x_{k+1} &= Wx_k - \alpha s_k, & x_0 &= x^0 \\ s_{k+1} &= Ws_k + \nabla f(x_{k+1}) - \nabla f(x_k), & s_0 &= \nabla f(x^0)\end{aligned}$$

**Remark.** The initialization is crucial to guarantee convergence of  $x_k$  to the consensual optimal solution  $\mathbf{1}x_\star$

## Strategy 2: distributed gradient method based on integral action

Given the consensus error  $e_k := (I - W)x_k$ , a *Proportional-Integral (PI) controller* reads

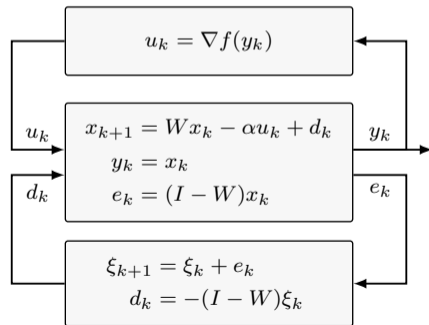
$$\begin{aligned}\xi_{k+1} &= \xi_k + e_k, & \xi_0 &= \xi^0 \\ d_k &= K_I \xi_k + K_P e_k\end{aligned}$$

for properly specified stabilizing gains  $K_I$  (integral) and  $K_P$  (proportional)

For  $K_I = -(I - W)$  and  $K_P = 0$ , the closed-loop system results in

$$\begin{aligned}x_{k+1} &= Wx_k - (I - W)\xi_k - \alpha \nabla f(x_k) \\ \xi_{k+1} &= \xi_k + (I - W)x_k\end{aligned}$$

**Remark.** The initialization is arbitrary



# Agent perspective of integral-action-based distributed gradient method

Each agent  $i$  implements the following local updates

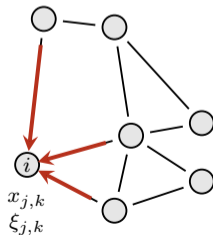
$$x_{i,k+1} = \sum_{j \in N_i} w_{ij} x_{j,k} - \xi_{i,k} + \sum_{j \in N_i} w_{ij} \xi_{j,k} - \alpha u_{i,k}, \quad x_{i,0} \in \mathbb{R}$$

$$\xi_{i,k+1} = \xi_{i,k} + x_{i,k} - \sum_{j \in N_i} w_{ij} x_{j,k}, \quad \xi_{i,0} \in \mathbb{R}$$

$$y_{i,k} = x_{i,k}$$

with  $u_{i,k} = \nabla f_i(y_{i,k})$ , where  $w_{ij}$  are the entries of the matrix  $W$  and  $N_i$  is the neighbor set of agent  $i$

**Remark.** Each agent  $i$  receives  $x_{j,k}$  and  $\xi_{j,k}$  from its neighbors  $j \in N_i$



# Equilibrium manifold for the int-act distributed gradient method

Consider the integral-action-based distributed gradient method

$$\begin{bmatrix} x_{k+1} \\ \xi_{k+1} \end{bmatrix} = \begin{bmatrix} W & -(I - W) \\ (I - W) & I \end{bmatrix} \begin{bmatrix} x_k \\ \xi_k \end{bmatrix} - \alpha \begin{bmatrix} I \\ 0 \end{bmatrix} \nabla f(x_k), \quad \begin{bmatrix} x_0 \\ \xi_0 \end{bmatrix} = \begin{bmatrix} x^0 \\ \xi^0 \end{bmatrix}$$

The equilibrium point  $(x_{\text{eq}}, \xi_{\text{eq}})$  satisfies

$$\begin{aligned} (I - W)x_{\text{eq}} &= -(I - W)\xi_{\text{eq}} - \alpha \nabla f(x_{\text{eq}}) \\ \xi_{\text{eq}} &= \xi_{\text{eq}} + (I - W)x_{\text{eq}} \end{aligned}$$

The second equation imposes  $x_{\text{eq}} \in \text{span } \mathbf{1}$ , hence  $\mathbf{1}^\top \nabla f(x_{\text{eq}}) = 0$

This implies that the equilibrium is

$$\begin{bmatrix} x_{\text{eq}} \\ \xi_{\text{eq}} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_{x_\star} \\ \alpha(I - W)^\dagger \nabla f(\mathbf{1}_{x_\star}) \end{bmatrix}$$

# Accelerated distributed consensus optimization

The (centralized) heavy-ball method is

$$x_{k+1} = (1 + \beta)x_k - \beta q_k - \alpha \nabla f(x_k)$$

$$q_{k+1} = x_k$$

for some  $\beta > 0$ . It can be replicated  $N$  times to obtain a *parallel algorithm* given by

$$x_{k+1} = \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+1} \end{bmatrix} = J \left( (1 + \beta)x_k - \beta q_k \right) - \alpha J \nabla f(Jx_k)$$

$$q_{k+1} = \begin{bmatrix} q_{k+1} \\ \vdots \\ q_{k+1} \end{bmatrix} = x_k$$

$$y_k = Jx_k$$

with  $u_k = \nabla f(u_k)$ . The accelerated algorithm is not amenable to distributed implementation because of

- the consensus mixing  $J((1 + \beta)x_k - \beta q_k)$
- the average update direction  $Ju_k$

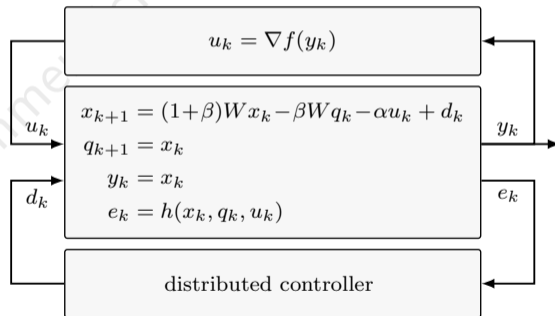
# Accelerated distributed optimization

**Idea.** Replace

$$J((1 + \beta)x_k - \beta q_k) \mapsto W((1 + \beta)x_k - \beta q_k)$$

and compensate for the aggregating term  $d_k = \alpha(I - J)u_k$  using a distributed and dynamic controller based, e.g., on the integral action

$$\begin{aligned}\xi_{k+1} &= \xi_k + (I - W)x_k \\ d_k &= -(I - W)\xi_k\end{aligned}$$



**Remark.** Do the centralized and distributed accelerated algorithms share the same passivity properties?