Control Methods for Distributed Optimization

ADMM and distributed ADMM

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

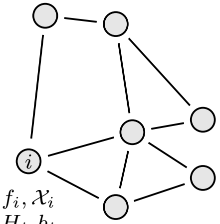
- The ADMM for constraint-coupled optimization
- The distributed ADMM fo constraint-coupled optimization

Constraint-coupled optimization (recall)

A *constraint-coupled optimization* problem is

$$\begin{aligned} \min_{x_1,\dots,x_N} \quad & \sum_{i=1}^{N} f_i(x_i) \\ \text{subj. to} \quad & \sum_{i=1}^{N} (H_i x_i - b_i) = 0 \\ & x_i \in \mathcal{X}_i, \qquad i = 1,\dots,N \end{aligned}$$

with $x_i \in \mathbb{R}^{n_i}$, $H_i \in \mathbb{R}^{p \times n_i}$, $b_i \in \mathbb{R}^p$, and $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$



Let

- $f(x) := \sum_{i=1}^{N} f_i(x_i)$ with $x := (x_1, \dots, x_N)$
- $H_{\rm d} := \text{diag}(H_1, \dots, H_N)$
- $b := (b_1, \dots, b_N)$, so that $\mathbf{1}^\top b = \sum_{i=1}^{N} b_i$
- $\mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_N$

ADMM for constraint-coupled optimization

Recall that the ADMM results in the following updates: for all $k \in \mathbb{N}$ perform

$$\begin{aligned} x_{k+1} &\in \underset{x \in \mathcal{X}}{\operatorname{argmin}}\ f(x) + \frac{1}{2c} \left\| c \left(H_{\mathrm{d}} x - H_{\mathrm{d}} x_k \right) + \mathbf{1}\lambda_k + \mathbf{1}\sigma_k \right\|^2 \\ \sigma_{k+1} &= \frac{c}{N} \mathbf{1}^\top (H_{\mathrm{d}} x_{k+1} - b) \\ \lambda_{k+1} &= \lambda_k + \sigma_{k+1} \end{aligned}$$

with $c > 0$, where $\sigma_k \in \mathbb{R}^p$ is the *feasibility error* and $\lambda_k \in \mathbb{R}^p$ is the *Lagrange multiplier*

Remark. It is a *parallel* optimization algorithm:

- N "workers" solve local optimization problems, for all $i = 1, \ldots, N$ perform

$$x_{i,k+1} \in \underset{x_i \in \mathcal{X}_i}{\operatorname{argmin}}\ f_i(x_i) + \frac{1}{2c} \left\| c \left(H_i x_i - H_i x_{i,k} \right) + \lambda_k + \sigma_k \right\|^2$$

- a master node updates the feasibility error and the dual variable

Convergence result of the ADMM algorithm

Theorem. Let the constraint-coupled optimization problem be a *convex program*, then

- the dual variable $\{\lambda_k\}_{k\in\mathbb{N}}$ converges to the optimal Lagrange multiplier $\lambda_\star$
- the primal variables $\{x_{1,k},\dots,x_{N,k}\}_{k\in\mathbb{N}}$ converge to the optimal primal solution $x_\star := (x_{1,\star},\dots,x_{N,\star})$

Remark. Uniqueness of the primal-dual solution pair $(x_\star, \lambda_\star)$ can be relaxed

Control-oriented ADMM reformulation

Absorbing the variable $\sigma_k = \frac{c}{N}\mathbf{1}^\top(H_\mathrm{d} x_k - b)$ yields

$$\begin{aligned} x_{k+1} &\in \operatorname*{argmin}_{x\in\mathcal{X}} \; f(x) + \tfrac{1}{2c}\left\|c\,(H_\mathrm{d}x - H_\mathrm{d}x_k) + \mathbf{1}\lambda_k + c\,J(H_\mathrm{d}x_k - b)\right\|^2 \\ \lambda_{k+1} &= \lambda_k + \tfrac{c}{N}\mathbf{1}^\top(H_\mathrm{d}x_{k+1} - b) \end{aligned}$$

with initial conditions $x_0 \in \mathcal{X}$ and $\lambda_0 \in \mathbb{R}^p$

Goal. Want to highlight a *Lur'e system*

The updates can be further manipulated to obtain

$$\begin{aligned} x_{k+1} &\in \operatorname*{argmin}_{x\in\mathcal{X}} \; f(x) + \tfrac{1}{2c}\Big\|c\,(H_\mathrm{d}x - b) + \underbrace{\mathbf{1}\lambda_k - c\,(I - J)(H_\mathrm{d}x_k - b)}_{\text{exogenous information}}\Big\|^2 \\ \lambda_{k+1} &= \lambda_k + \tfrac{c}{N}\mathbf{1}^\top\underbrace{(H_\mathrm{d}x_{k+1} - b)}_{\text{update direction}} \end{aligned}$$

Remark. The exogenous information involves a delayed version of the update direction

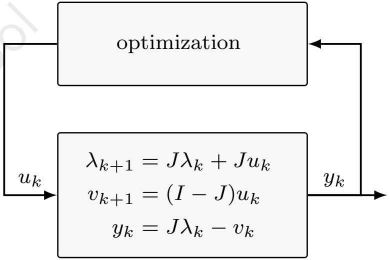
The ADMM for constraint-coupled optimization is a feedback system

Introducing v_k as a filtered, delayed version of the update direction $c\,(H_\mathrm{d} x_{k+1} - b)$ yields

$$\begin{aligned} \lambda_{k+1} &= \lambda_k + \tfrac{1}{N}\mathbf{1}^\top u_k \\ v_{k+1} &= (I - J)u_k \\ y_k &= \mathbf{1}\lambda_k - v_k \end{aligned}$$

where the *output* y_k represents the exogenous information necessary to compute the *input* u_k by solving the following optimization step

$$\begin{aligned} x_{k+1} &\in \operatorname*{argmin}_{x\in\mathcal{X}}\ f(x) + \tfrac{1}{2c}\left\|c\,(H_\mathrm{d} x - b) + y_k\right\|^2 \\ u_k &= c\,(H_\mathrm{d} x_{k+1} - b) \end{aligned}$$



Remark. The optimization step represents a *static (memoryless) nonlinearity* from y_k to u_k

Remark. The feedback system is a Lur'e system

Algorithm analysis: error coordinates reformulation

An equivalent (*though not implementable*) reformulation is obtained by “replacing” b with $H_{\mathrm{d}} x_\star$

$$x_{k+1} \in \underset{x \in \mathcal{X}}{\operatorname{argmin}} \ f(x) + \frac{1}{2c} \| c\,(H_{\mathrm{d}} x - H_{\mathrm{d}} x_\star) + \mathbf{1}\lambda_\star + \underbrace{y_k - \mathbf{1}\lambda_\star + c\,(H_{\mathrm{d}} x_\star - H_{\mathrm{d}} b)}_{\tilde{y}_k := \underbrace{\mathbf{1}\lambda_k - \mathbf{1}\lambda_\star}_{\tilde{\lambda}_k} - \underbrace{(v_k - v_\star)}_{\tilde{v}_k}} \|^2$$

$$\underbrace{\lambda_{k+1} - \lambda_\star}_{\tilde{\lambda}_{k+1}} = \lambda_k - \lambda_\star + \frac{1}{N}\mathbf{1}^\top \underbrace{c\,(H_{\mathrm{d}} x_{k+1} - H_{\mathrm{d}} x_\star)}_{\tilde{u}_k}$$

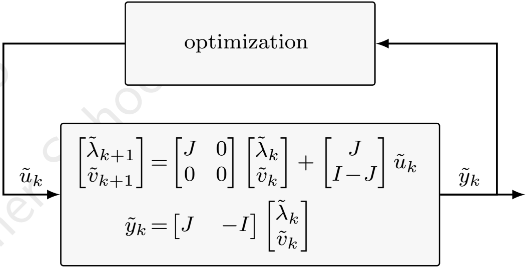
$$\underbrace{v_{k+1} - v_\star}_{\tilde{v}_{k+1}} = (I - J)\tilde{u}_k$$

where $(x_\star, \lambda_\star)$ is the primal-dual solution of the problem and $v_\star := c\,(H_{\mathrm{d}} x_\star - b)$

Algorithm analysis: error coordinates reformulation

Finally, we obtain the *error dynamics* given by

$$\begin{aligned}\tilde{\lambda}_{k+1} &= \tilde{\lambda}_k + \tfrac{1}{N}\mathbf{1}^\top \tilde{u}_k \\ \tilde{v}_{k+1} &= (I - J)\tilde{u}_k \\ \tilde{y}_k &= \mathbf{1}\tilde{\lambda}_k - \tilde{v}_k\end{aligned}$$



in feedback with $\tilde{u}_k := \phi(\tilde{y}_k)$, given by

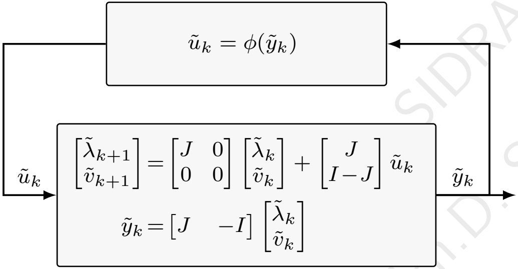
$$\begin{aligned}x^+ &\in \operatorname*{argmin}_{x\in\mathcal{X}}\; f(x) + \tfrac{1}{2c}\left\|c\,(H_\mathrm{d}x - H_\mathrm{d}x_\star) + \mathbf{1}\lambda_\star + \tilde{y}_k\right\|^2 \\ \tilde{u}_k &= c\,(H_\mathrm{d}x^+ - H_\mathrm{d}x_\star)\end{aligned}$$

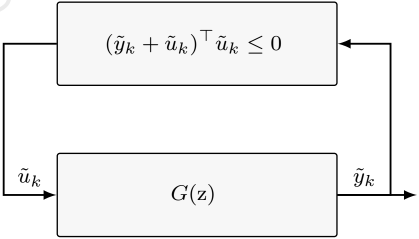
Goal. Study the properties of the interconnection focusing on the individual components

Passivity-based stability analysis

For the convergence/stability analysis of ADMM, let

- the (replicated) linear plant be represented with its *transfer matrix* $G(\mathrm{z})$
- the nonlinearity be replaced by its *sector bound* characterization $(\tilde{y}_k + \tilde{u}_k)^\top \tilde{u}_k \leq 0$ for all $k \in \mathbb{N}$





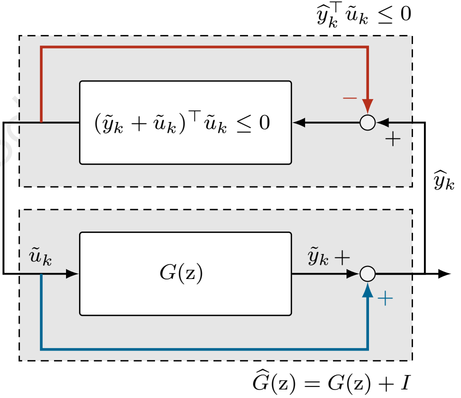
Passivity-based analysis: loop transformation

The optimization step exhibits an *excess of passivity* in its output $\tilde{u}_k$ (OFP) that can be *transferred* through a loop transformation

The transfer matrix from $\tilde{u}_k$ to $\widehat{y}_k := \tilde{y}_k + \tilde{u}_k$ is

$$\begin{aligned}\widehat{G}(z) &= C(zI - A)^{-1}B + I_{pN} \\ &= \begin{bmatrix} J & -I \end{bmatrix} \begin{bmatrix} (z-1)I & 0 \\ 0 & zI \end{bmatrix}^{-1} \begin{bmatrix} J \\ I - J \end{bmatrix} + I \\ &= \frac{1}{z-1} J - \frac{1}{z}(I - J) + I \\ &= T \begin{bmatrix} \frac{z}{z-1} I_p & \\ & \frac{z-1}{z} I_{p(N-1)} \end{bmatrix} T^{-1}\end{aligned}$$

where $\widehat{y}_k$ and $\tilde{u}_k$ satisfies the monotonicity condition $\widehat{y}_k^\top \tilde{u}_k \leq 0$



Remark. The diagonal entries of $\widehat{G}(z)$ are *discrete positive real*. Hence, the system is *passive*

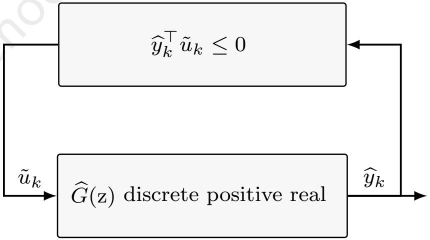
Convergence result for the ADMM

Proposition. The feedback interconnection of two passive systems is passive

Being $\widehat{G}(\mathrm{z})$ *discrete positive real*, there exists a quadratic storage function V and matrices M_y and M_u satisfying

$$V\left(\begin{bmatrix}\tilde{\lambda}_{k+1}\\ \tilde{v}_{k+1}\end{bmatrix}\right) - V\left(\begin{bmatrix}\tilde{\lambda}_{k}\\ \tilde{v}_{k}\end{bmatrix}\right) \leq \widehat{y}_k^\top \tilde{u}_k - \frac{1}{2}\left\| M_y \begin{bmatrix}\tilde{\lambda}_{k}\\ \tilde{v}_{k}\end{bmatrix} + M_u \tilde{u}_k \right\|^2$$

with $\tilde{u}_k = \tilde{\phi}(\widehat{y}_k)$ such that $\widehat{y}_k^\top \tilde{\phi}(\widehat{y}_k) \leq 0$



It implies that $\lim_{k\to\infty} \widehat{y}_k^\top \tilde{u}_k = 0$ and a Lasalle argument (with a refined feedforward gain $D \neq I$) ensures that also

$$\lim_{k\to\infty} \lambda_k = \lambda_\star$$

$$\lim_{k\to\infty} x_{k+1} = x_\star$$

Some questions

The ADMM for constraint-coupled optimization is

$$\begin{aligned}
\tilde{\lambda}_{k+1} &= J\tilde{\lambda}_k + J\tilde{u}_k \\
\tilde{v}_{k+1} &= (I - J)\tilde{u}_k \\
\tilde{y}_k &= J\tilde{\lambda}_k - \tilde{v}_k
\end{aligned}$$

with $\tilde{u}_k = \phi(\tilde{y}_k)$

Remark. It enjoys a *sparsity pattern*, e.g., in the nonlinear map ϕ, but also an aggregating averaging term $I - J$

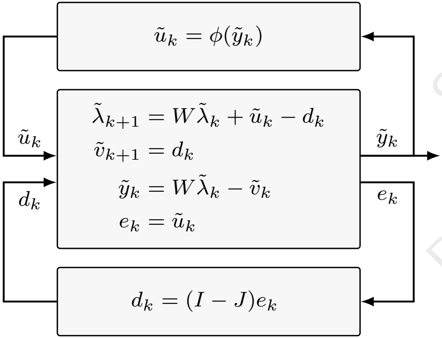
- Is it possible to implement the ADMM in a distributed fashion?
- Is it possible to exploit the system-theoretic approach to design a distributed algorithm?

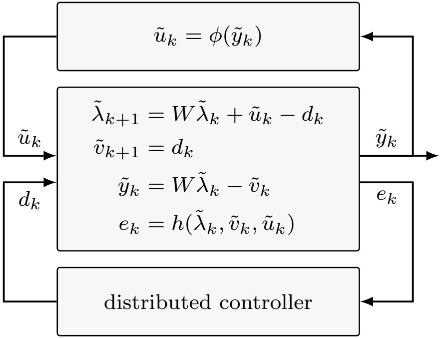
Unleashing distributed constraint-coupled optimization

Isolating the aggregating terms in the linear update yields

$$\tilde{\lambda}_{k+1} = J\tilde{\lambda}_k + \tilde{u}_k - (I - J)\tilde{u}_k$$
$$\tilde{v}_{k+1} = (I - J)\tilde{u}_k$$
$$\tilde{y}_k = J\tilde{\lambda}_k - \tilde{v}_k$$

As before, replace $J\tilde{\lambda}_k \longmapsto W\tilde{\lambda}_k$ and handle the aggregating term $d_k := (I-J)\tilde{u}_k$ through a *distributed controller*





Toward a distributed implementation of the ADMM

The nonlinearity stays unchanged and, hence, *decoupled* across the agents

$$\tilde{y}_k = \begin{bmatrix} \tilde{y}_{1,k} \\ \vdots \\ \tilde{y}_{N,k} \end{bmatrix} \quad \longmapsto \quad \tilde{u}_k = \phi(\tilde{y}_k) = \begin{bmatrix} \phi_1(\tilde{y}_{1,k}) \\ \vdots \\ \phi_N(\tilde{y}_{N,k}) \end{bmatrix}$$

Two alternative strategies for the *distributed controller* are

1. the *dynamic average consensus* to track the average of the update direction $e_k := \tilde{u}_k = \phi(\tilde{y}_k)$
2. the *integral action* to reject the consensus error $e_k := (I - W)\tilde{\lambda}_k$