System Theory Tools for Optimization in Learning and Control

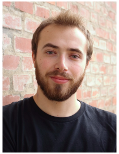
Giuseppe Notarstefano

Department of Electrical, Electronic, and Information Engineering
Alma Mater Studiorum Università di Bologna
giuseppe.notarstefano@unibo.it

Workshop on Optimization for Learning and Control
IMT - Lucca, Italy - 6^{th} June 2025

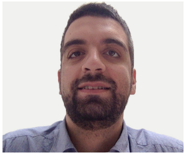
Thanks to...

Guido Carnevale



Lorenzo Sforni

Nicola Mimmo



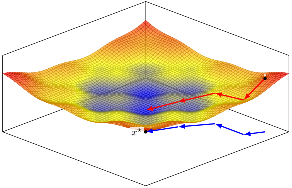
Ivano Notarnicola

Lorenzo Pichierri

System Theory Tools for Optimization and Learning

System Theory Tools

LaSalle and Lyapunov Theory



Timescale Separation

Slow System

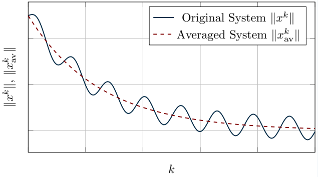
$$x^{k+1} = x^k + \delta f(x^k, z^k)$$

z^k x^k

Fast System

$$z^{k+1} = g(z^k, x^k)$$

Averaging theory



Distributed optimization algorithms

Optimization-oriented dynamics

Consensus-oriented dynamics

Accelerated optimization algorithms

Gradient-like algorithm

Extra dynamics

Learning-driven Optimal Control

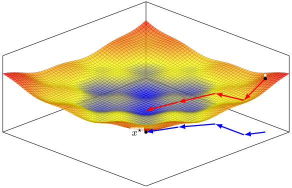
Learning Algorithm

Physical Plant

System Theory Tools for Optimization and Learning

System Theory Tools

LaSalle and Lyapunov Theory



Timescale Separation

Slow System

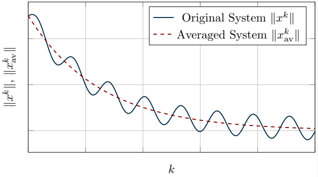
$$x^{k+1} = x^k + \delta f(x^k, z^k)$$

z^k $\quad$ x^k

Fast System

$$z^{k+1} = g(z^k, x^k)$$

Averaging theory



Distributed optimization algorithms

Optimization-oriented dynamics

Consensus-oriented dynamics

Accelerated optimization algorithms

Gradient-like algorithm

Extra dynamics

Learning-driven Optimal Control

Learning Algorithm

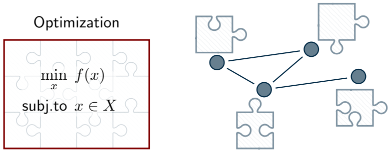
Physical Plant

Distributed Algorithms for Optimization and Games

Network of N peer agents aim at solving optimization-based tasks without a central coordinator

Each agent i

- knows only part of the problem (local, private data)
- performs local computations
- communicates only with neighboring agents (digraph $\mathcal{G}$)



Distributed Algorithms for Optimization and Games

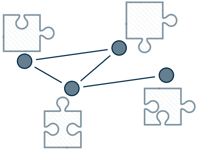
Network of N peer agents aim at solving optimization-based tasks without a central coordinator

Each agent i

- knows only part of the problem (local, private data)
- performs local computations
- communicates only with neighboring agents (digraph $\mathcal{G}$)

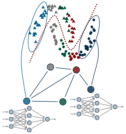
Optimization

$$\min_{x} \; f(x)$$
$$\text{subj.to } \; x \in X$$



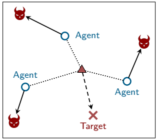
Different scenarios to model a large variety of tasks in both cooperative and competitive frameworks

$$\min_{x} \sum_{i=1}^{N} f_i(x)$$

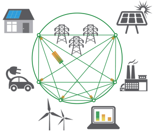


$$\min_{x_1,\dots,x_N} \sum_{i=1}^{N} f_i(x_i)$$
$$\text{subj.to } \sum_{i=1}^{N} g_i(x_i) \leq 0$$

$$\min_{x_1,\dots,x_N} \sum_{i=1}^{N} f_i(x_i, \sigma(x))$$
$$\text{subj.to } \; x_i \in X_i \; \forall\, i$$
$$\sigma(x) = \frac{1}{N} \sum_{i=1}^{N} \phi_i(x_i)$$



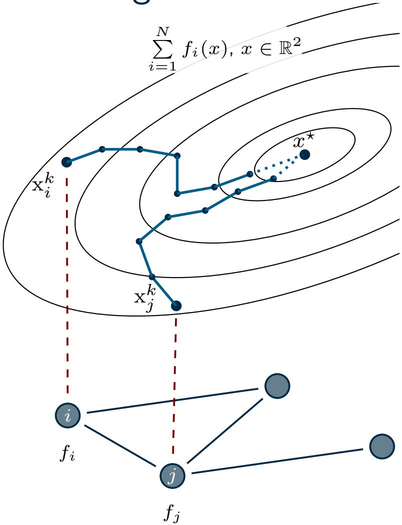
$$\min_{x_i} \; J_i(x_i, \sigma(x))$$
$$\text{subj.to } \; x_i \in X_i \; \forall\, i$$
$$\sigma(x) = \frac{1}{N} \sum_{i=1}^{N} \phi_i(x_i)$$



Distributed Consensus Optimization via Gradient Tracking

Consider the optimization problem

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



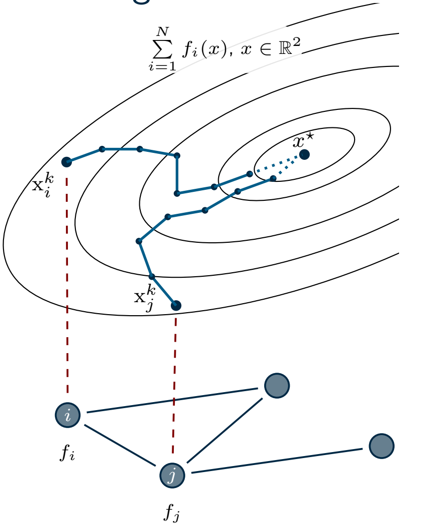
Distributed Consensus Optimization via Gradient Tracking

Consider the optimization problem

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

Centralized "gradient" method

$$x_i^{k+1} = \overbrace{\frac{1}{N}\sum_{j=1}^{N} x_j^k}^{\text{average (global)}} - \gamma \overbrace{\sum_{j=1}^{N} \nabla f_j(x_j^k)}^{\text{gradient (global)}}$$



Distributed Consensus Optimization via Gradient Tracking

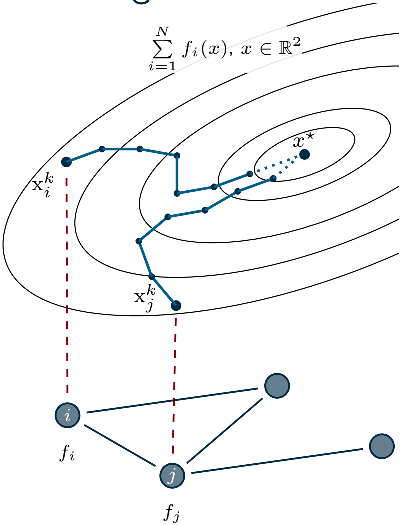
Consider the optimization problem

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

Gradient Tracking algorithm

$$\mathrm{x}_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} \mathrm{x}_j^k - \gamma \overbrace{\mathrm{s}_i^k}^{\text{proxy for } \sum_{j=1}^N \nabla f_j(\mathrm{x}_j^k)}$$

$$\mathrm{s}_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} \mathrm{s}_j^k + \nabla f_i(\mathrm{x}_i^{k+1}) - \nabla f_i(\mathrm{x}_i^k)$$



Distributed Consensus Optimization via Gradient Tracking

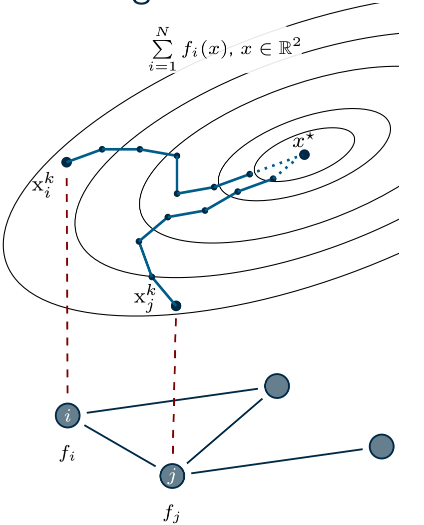
Consider the optimization problem

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

Gradient Tracking algorithm

$$\mathrm{x}_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} \mathrm{x}_j^k - \gamma(\overbrace{\mathrm{z}_i^k + \nabla f_i(\mathrm{x}_i^k)}^{\text{proxy for } \sum_{j=1}^N \nabla f_j(\mathrm{x}_j^k)})$$

$$\mathrm{z}_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} \mathrm{z}_j^k + \sum_{j \in \mathcal{N}_i} a_{ij} \nabla f_j(\mathrm{x}_j^k) - \nabla f_i(\mathrm{x}_i^k)$$

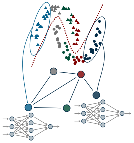


Convergence of Gradient Tracking - Nonconvex setting

Gradient Tracking

$$
\begin{aligned}
\mathrm{x}_i^{k+1} &= \sum_{j\in\mathcal{N}_i} a_{ij}\mathrm{x}_j^k - \gamma \mathrm{z}_i^k - \gamma\nabla f_i(\mathrm{x}_i^k)\\
\mathrm{z}_i^{k+1} &= \sum_{j\in\mathcal{N}_i} a_{ij}(\mathrm{z}_j^k + \nabla f_j(\mathrm{x}_j^k)) - \nabla f_i(\mathrm{x}_i^k)
\end{aligned}
$$

$$\boxed{\min_x \sum_{i=1}^N f_i(x)}$$



Intuition: Gradient Tracking is a Two-Time-Scale System

Slow system

$$\bar{\mathrm{x}}^{k+1} = \bar{\mathrm{x}}^k - \frac{\gamma}{N}\mathbf{1}^\top G(\mathbf{1}\bar{\mathrm{x}}^k + R\mathrm{x}_\perp^k)$$

$\mathrm{x}_\perp^k$ $\qquad$ $\bar{\mathrm{x}}^k$

$$\begin{bmatrix}\mathrm{x}_\perp^{k+1}\\ \mathrm{z}_\perp^{k+1}\end{bmatrix} = \begin{bmatrix}R^\top AR & -I\\ 0 & R^\top AR\end{bmatrix}\begin{bmatrix}\mathrm{x}_\perp^{k}\\ \mathrm{z}_\perp^{k}\end{bmatrix} + \gamma\begin{bmatrix}-R^\top\\ R^\top(A-I)\end{bmatrix} G(\mathbf{1}\bar{\mathrm{x}}^k + R\mathrm{x}_\perp^k)$$

Fast system

- $\bar{\mathrm{x}}^k := \frac{1}{N}\sum_{i=1}^N \mathrm{x}_i^k$, $\mathrm{x}_\perp^k := R^\top \mathrm{x}^k$, $\mathrm{z}_\perp^k := R^\top \mathrm{z}^k$, with $R^\top\mathbf{1} = 0$ and $R^\top R = I$
- G stacking local gradients ∇f_i, $A := \mathcal{A}\otimes I_d$ with $\mathcal{A}$ consensus matrix

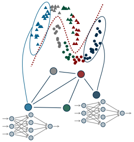
Carnevale, Notarstefano, " Nonconvex Distributed Optimization via Lasalle and Singular Perturbations." (L-CSS '22)

Convergence of Gradient Tracking - Nonconvex setting

Gradient Tracking

$$
\begin{aligned}
\mathrm{x}_i^{k+1} &= \sum_{j \in \mathcal{N}_i} a_{ij} \mathrm{x}_j^k - \gamma \mathrm{z}_i^k - \gamma \nabla f_i(\mathrm{x}_i^k) \\
\mathrm{z}_i^{k+1} &= \sum_{j \in \mathcal{N}_i} a_{ij} (\mathrm{z}_j^k + \nabla f_j(\mathrm{x}_j^k)) - \nabla f_i(\mathrm{x}_i^k)
\end{aligned}
$$

$$\min_x \sum_{i=1}^N f_i(x)$$



Theorem

Consider Gradient Tracking initialized so that $\sum_{i=1}^N \mathrm{z}_i^0 = 0$. Assume

- Graph $\mathcal{G}$ strongly connected and $\mathcal{A}$ doubly stochastic
- $\sum_{i=1}^N f_i$ radially unbounded and $\nabla f_1, \ldots, \nabla f_N$ Lipschitz continuous (possibly nonconvex)

Then, there exists $\bar{\gamma} > 0$ s.t. for any $\gamma \in (0, \bar{\gamma})$, it holds

$$\lim_{k \to \infty} \operatorname{DIST}(\mathrm{x}_i^k, \mathcal{X}^\star) = 0 \qquad \forall i \in \{1, \ldots, N\}$$

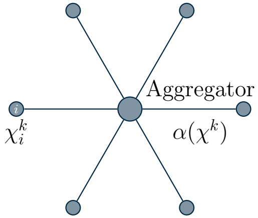
with $\mathcal{X}^\star$ set of stationary points of the optimization problem.

Carnevale, Notarstefano, " Nonconvex Distributed Optimization via Lasalle and Singular Perturbations." (L-CSS '22)

Optimization-oriented Centralized Algorithm

Optimization meta-algorithm:

$$\begin{aligned}\chi_i^{k+1} &= g_i(\chi_i^k, \overbrace{\alpha(\chi^k)}^{\text{aggregate information}})\\ \mathrm{x}_i^k &= \eta_i(\chi_i^k)\end{aligned}$$

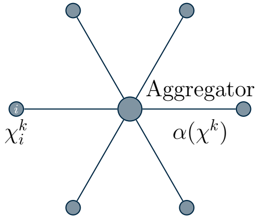


Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Optimization-oriented Centralized Algorithm

Optimization meta-algorithm:

$$\chi_i^{k+1} = g_i(\chi_i^k, \overbrace{\alpha(\chi^k)}^{\text{aggregate information}})$$
$$\mathrm{x}_i^k = \eta_i(\chi_i^k)$$



- Consensus optimization

$$g_i(x_i, \alpha(x)) = \frac{1}{N}\sum_{j=1}^{N} x_j - \gamma \sum_{j=1}^{N} \nabla f_j(x_j)$$

- Constraint-coupled optimization

$$g_i(\chi_i, \alpha(\chi)) = \begin{bmatrix} x_i - \gamma \nabla_{x_i} L(x_i, \sum_{j=1}^{N} g_j(x_j), \lambda_i) \\ \frac{1}{N}\sum_{j=1}^{N} \lambda_j + \gamma \nabla_{\lambda_i} L(x_i, \sum_{j=1}^{N} g_j(x_j), \lambda_i) \end{bmatrix}$$

- Aggregative optimization

$$g_i(x_i, \alpha(x)) = x_i - \gamma \left[\nabla \sum_{j=1}^{N} f_j(x_j, \sigma(x)) \right]_i$$

- Aggregative games

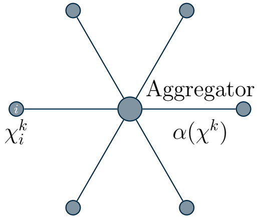
$$g_i(x_i, \alpha(x)) = x_i - \gamma \left[\nabla J_i(x_i, \sigma(x)) \right]_i$$

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Optimization-oriented Centralized Algorithm

Optimization meta-algorithm:

$$\chi_i^{k+1} = g_i(\chi_i^k, \overbrace{\alpha(\chi^k)}^{\text{aggregate information}})$$
$$\mathrm{x}_i^k = \eta_i(\chi_i^k)$$



- Consensus optimization

$$\alpha(\chi) := \begin{bmatrix} \frac{1}{N}\sum_{i=1}^N x_i \\ \sum_{i=1}^N \nabla f_i(x_i) \end{bmatrix}$$

- Constraint-coupled optimization

$$\alpha(\chi) := \begin{bmatrix} \frac{1}{N}\sum_{i=1}^N \lambda_i \\ \sum_{i=1}^N g_i(x_i) \end{bmatrix}$$

- Aggregative optimization

$$\alpha(\chi) := \begin{bmatrix} \frac{1}{N}\sum_{i=1}^N \phi_i(x_i) \\ \sum_{i=1}^N \nabla_2 f_i(x_i, \sigma(x)) \end{bmatrix}$$

- Aggregative games

$$\alpha(\chi) := \frac{1}{N}\sum_{i=1}^N \phi_i(x_i)$$

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Systematic Transition to a Fully-distributed Algorithm

Idea: introduce variables z_i reconstructing unavailable global data $\sigma(\chi)$ running consensus-based dynamics

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Systematic Transition to a Fully-distributed Algorithm

Idea: introduce variables z_i reconstructing unavailable global data $\sigma(\chi)$ running consensus-based dynamics

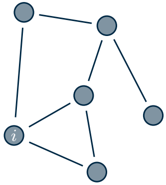
Naive distributed version: double-scale algorithm only using neighbors' variables $(\chi^k_{\mathcal{N}_i}, \mathrm{z}^k_{\mathcal{N}_i})$

Optimization For $k = 0, 1, \ldots$

$$\chi_i^{k+1} = g_i(\chi_i^k, \overbrace{\hat{\alpha}_i(\chi_i^k, \mathrm{z}_i^{k,\infty})}^{\text{proxy for } \alpha(\chi^k)})$$

Consensus For $\tau = 0, 1, \ldots$

$$\mathrm{z}_i^{k,\tau+1} = h_i(\chi^k_{\mathcal{N}_i}, \mathrm{z}^{k,\tau}_{\mathcal{N}_i})$$



Example: causal perturbed consensus dynamics for $\alpha(\chi^k) = \frac{1}{N}\sum_{i=1}^N \varphi_i(\chi_i^k)$

$$\mathrm{z}_i^{k,\tau+1} = \underbrace{\sum_{j\in\mathcal{N}_i} a_{ij}(\mathrm{z}_j^{k,\tau} + \varphi_j(\chi_j^k)) - \varphi_i(\chi_i^k)}_{h_i(\chi^k_{\mathcal{N}_i}, z^{k,\tau}_{\mathcal{N}_i})}$$

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

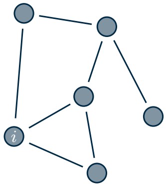
Systematic Transition to a Fully-distributed Algorithm

Idea: introduce variables z_i reconstructing unavailable global data $\sigma(\chi)$ running consensus-based dynamics

Distributed single-scale algorithm: plain interconnection with no guarantees

$$\chi_i^{k+1} = g_i\big(\chi_i^k, \hat{\alpha}_i(\chi_i^k, \mathrm{z}_i^k)\big) \qquad \text{(optimization)}$$

$$\mathrm{z}_i^{k+1} = h_i(\chi_{\mathcal{N}_i}^k, z_{\mathcal{N}_i}^k) \qquad \text{(consensus)}$$



Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

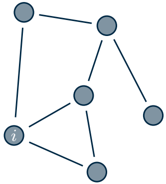
Systematic Transition to a Fully-distributed Algorithm

Idea: introduce variables z_i reconstructing unavailable global data $\sigma(\chi)$ running consensus-based dynamics

Distributed single-scale algorithm: design two-time-scale system

$$\chi_i^{k+1} = \chi_i^k + \delta \left(g_i(\chi_i^k, \hat{\alpha}_i(\chi_i^k, \mathrm{z}_i^k)) - \chi_i^k \right) \qquad \text{(optimization)}$$

$$\mathrm{z}_i^{k+1} = h_i(\chi_{\mathcal{N}_i}^k, z_{\mathcal{N}_i}^k) \qquad \text{(consensus)}$$

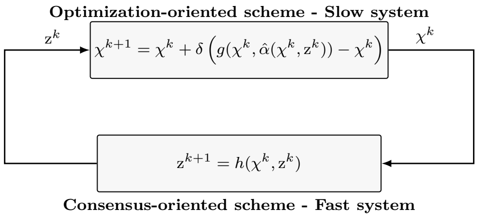


with $\delta > 0$ tuning parameter to "modulate" the speed of variation of χ_i^k

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Distributed Algorithm as a Two-Time-Scale System

Intuition: distributed algorithm is an interconnected Two-Time-Scale System



$$g,\ \hat{\alpha},\ h,\ \chi^k,\ \mathrm{z}^k,\ \mathrm{x}^k,\ \eta$$

stacking

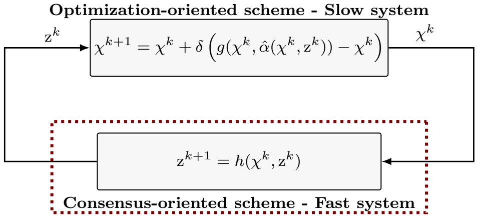
$$g_i,\ \hat{\alpha}_i,\ h_i,\ \chi_i^k,\ \mathrm{z}_i^k,\ \mathrm{x}_i^k,\ \eta_i$$

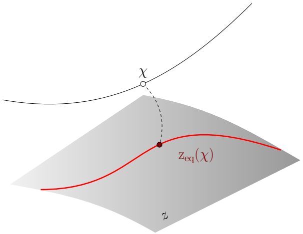
$$\chi^k := \begin{bmatrix} \chi_1^k \\ \vdots \\ \chi_N^k \end{bmatrix}, \quad \mathrm{z}^k := \begin{bmatrix} \mathrm{z}_1^k \\ \vdots \\ \mathrm{z}_N^k \end{bmatrix}, \quad \ldots$$

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Distributed Algorithm as a Two-Time-Scale System

Intuition: distributed algorithm is an interconnected Two-Time-Scale System



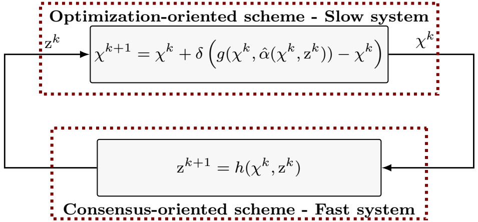


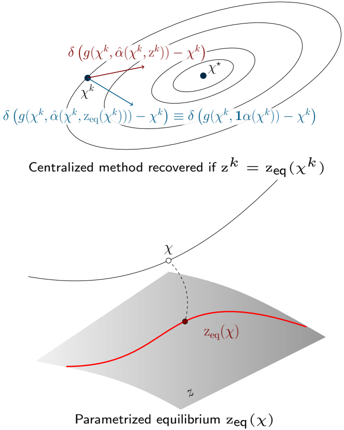
Parametrized equilibrium $z_{eq}(\chi)$

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Distributed Algorithm as a Two-Time-Scale System

Intuition: distributed algorithm is an interconnected Two-Time-Scale System





Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Convergence Results

Theorem

Let $\mathcal{X}^\star$ be the set of problem critic points (stationary points, Nash equilibria, etc...).
Assume Lipschitz continuity of g, h, $\hat{\alpha}$, and that

- Convergence of centralized optimization algorithm
  there exists radially unbounded function W proving $\lim_{k\to\infty} \text{DIST}(\mathrm{x}^k, \mathcal{X}^\star) = 0$ along the trajectories of

$$\chi^{k+1} = g(\chi^k, \mathbf{1}\alpha(\chi^k))$$
$$\mathrm{x}_{\mathsf{cntr}}^k = \eta(\chi^k);$$

- Consensus exponential stability
  there exists a Lipschitz continuous equilibrium function z_{eq} such that
  - $\hat{\alpha}(\chi, \mathrm{z}_{\mathsf{eq}}(\chi)) = \mathbf{1}\alpha(\chi)$ for all χ;
  - $\mathrm{z}_{\mathsf{eq}}(\chi)$ globally exponentially stable uniformly in χ for $\mathrm{z}^{k+1} = h(\chi, \mathrm{z}^k)$.

Then, for small $\delta > 0$, the trajectories of the distributed algorithm satisfy

$$\lim_{k\to\infty} \text{DIST}(\mathrm{x}^k, \mathcal{X}^\star) = 0$$

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Convergence Results

Corollary
Let $x^\star = \eta(\chi^\star)$ be the unique problem solution (e.g., strong convexity, strong monotonicity).
Assume Lipschitz continuity of g, h, $\hat{\alpha}$, and that

- Convergence of centralized optimization algorithm
  there exists a Lyapunov function W proving global exponential stability of $\chi^\star$ for

$$\chi^{k+1} = g(\chi^k, \mathbf{1}\alpha(\chi^k))$$
$$\mathrm{x}_{\text{cntr}}^k = \eta(\chi^k);$$

- Consensus exponential stability
  there exists a Lipschitz continuous equilibrium function z_{eq} such that
  - $\hat{\alpha}(\chi, \mathrm{z}_{\text{eq}}(\chi)) = \mathbf{1}\alpha(\chi)$ for all χ;
  - $\mathrm{z}_{\text{eq}}(\chi)$ is globally exponentially stable uniformly in χ for $\mathrm{z}^{k+1} = h(\chi, \mathrm{z}^k)$.

Then, for small $\delta > 0$ and some $c_1, c_2 > 0$, the trajectories of the distributed algorithm satisfy

$$\left\|\mathrm{x}^k - x^\star\right\| \leq c_1 \left\|\mathrm{x}^0 - x^\star\right\| e^{-c_2 k}$$

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Convergence Analysis based on Timescale Separation

Sketch of the proof

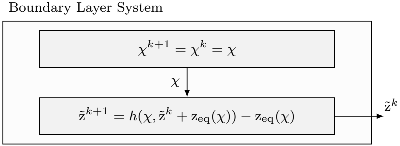
1. Build two auxiliary systems called boundary layer system and reduced system

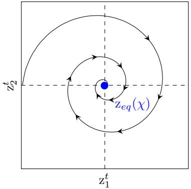
Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Convergence Analysis based on Timescale Separation

Sketch of the proof

1. Build two auxiliary systems called boundary layer system and reduced system
2. Boundary layer system: z dynamics with arbitrarily fixed χ and $\tilde{\mathrm{z}} := \mathrm{z} - \mathrm{z}_{\mathrm{eq}}(\chi)$





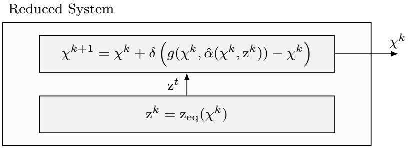
Lyapunov function U proving that the origin is globally exponentially stable (uniformly in χ)

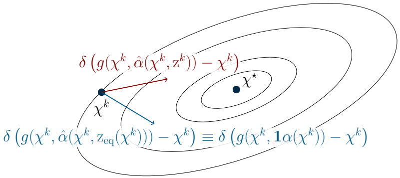
Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Convergence Analysis based on Timescale Separation

Sketch of the proof

1. Build two auxiliary systems called boundary layer system and reduced system
2. Boundary layer system: z dynamics with arbitrarily fixed χ and $\tilde{\mathrm{z}} := \mathrm{z} - \mathrm{z}_{\mathsf{eq}}(\chi)$
3. Reduced system: χ dynamics with $\mathrm{z}^k = \mathrm{z}_{\mathsf{eq}}(\chi^k)$





Radially unbounded function W proving attractiveness of the set of solutions for small $\delta > 0$

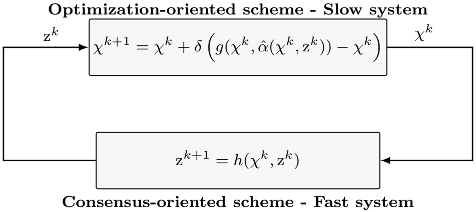
$$\lim_{k \to \infty} \textsc{dist}(\mathrm{x}^k, \mathcal{X}^\star) = 0$$

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Convergence Analysis based on Timescale Separation

Sketch of the proof

1. Build two auxiliary systems called boundary layer system and reduced system
2. Boundary layer system: z dynamics with arbitrarily fixed χ and $\tilde{\mathrm{z}} := \mathrm{z} - \mathrm{z}_{\mathsf{eq}}(\chi)$
3. Reduced system: χ dynamics with $\mathrm{z}^k = \mathrm{z}_{\mathsf{eq}}(\chi^k)$
4. Prove convergence in a LaSalle sense of the interconnection using $V(\chi, \tilde{\mathrm{z}}) = W(\chi) + U(\tilde{\mathrm{z}})$.



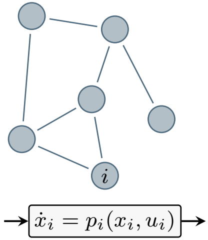
Custom theorem!

Carnevale, Mimmo, Notarstefano, "A Unifying System Theory Framework for Distributed Optimization and Games." (TAC, 2025)

Distributed Feedback Optimization

Setup: N control systems communicating over a graph

$$\dot{x}_i = p_i(x_i, u_i) \quad \forall i \in \{1, \dots, N\}$$



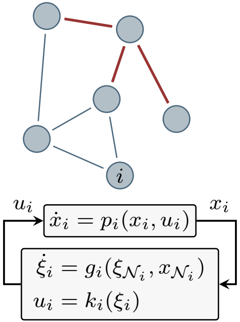
Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Distributed Feedback Optimization

Setup: N control systems communicating over a graph

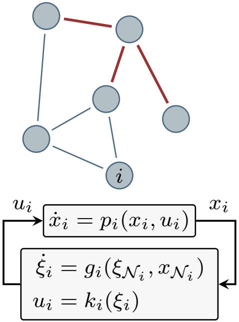
$$\dot{x}_i = p_i(x_i, u_i) \quad \forall i \in \{1, \dots, N\}$$

Distributed feedback paradigm:
control law of system i depending on neighboring systems $j \in \mathcal{N}_i$



Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Distributed Feedback Optimization



Setup: N control systems communicating over a graph

$$\dot{x}_i = p_i(x_i, u_i) \quad \forall i \in \{1, \dots, N\}$$

Distributed feedback paradigm:
control law of system i depending on neighboring systems $j \in \mathcal{N}_i$

Optimization goal: design $u = (u_1, \dots, u_N)$ such that $x = (x_1, \dots, x_N) \to x^\star$ optimal with respect to

$$\min_{\substack{x_1,\dots,x_N \\ u_1,\dots,u_N}} \sum_{i=1}^{N} f_i(x_i, \sigma(x)), \qquad \text{with } \sigma(x) = \frac{1}{N}\sum_{i=1}^{N} \phi_i(x_i)$$

$$\text{subj. to } x_i = h_i(u_i), \quad \forall i$$

Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Aggregative Tracking Feedback

Reduced optimization problem

$$\min_{u_1,\ldots,u_N} \sum_{i=1}^{N} f_i(h_i(u_i), \sigma(h(u)))$$

with $p_i(h_i(u_i), u_i) = 0$ and $x_i = h_i(u_i)$ globally exponentially stable.

Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Aggregative Tracking Feedback

Reduced optimization problem

$$\min_{u_1,\dots,u_N} \sum_{i=1}^{N} f_i(h_i(u_i), \sigma(h(u)))$$

with $p_i(h_i(u_i), u_i) = 0$ and $x_i = h_i(u_i)$ globally exponentially stable.

$$\dot{u}_i = -\delta_1 \nabla h_i(u_i) \left(\nabla_1 f_i(h_i(u_i), \sigma(h(u))) + \frac{\nabla \phi(h_i(u_i))}{N} \sum_{j=1}^{N} \nabla_2 f_j(h_j(u_j), \sigma(h(u))) \right)$$

Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Aggregative Tracking Feedback

Reduced optimization problem

$$\min_{u_1,\dots,u_N} \sum_{i=1}^{N} f_i(h_i(u_i), \sigma(h(u)))$$

with $p_i(h_i(u_i), u_i) = 0$ and $x_i = h_i(u_i)$ globally exponentially stable.

$$\dot{u}_i = -\delta_1 \nabla h_i(u_i) \left(\nabla_1 f_i(h_i(u_i), \sigma(h(u))) + \frac{\nabla \phi(h_i(u_i))}{N} \sum_{j=1}^{N} \nabla_2 f_j(h_j(u_j), \sigma(h(u))) \right)$$

Issue 1: generates a flow of equilibria (dynamics is ignored)

Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Aggregative Tracking Feedback

Reduced optimization problem

$$\min_{u_1,\dots,u_N} \sum_{i=1}^{N} f_i(h_i(u_i), \sigma(h(u)))$$

with $p_i(h_i(u_i), u_i) = 0$ and $x_i = h_i(u_i)$ globally exponentially stable.

$$\dot{u}_i = -\delta_1 \nabla h_i(u_i)\left(\nabla_1 f_i(x_i, \sigma(x)) + \frac{\nabla\phi(x_i)}{N}\sum_{j=1}^{N} \nabla_2 f_j(x_j, \sigma(x))\right)$$

Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Aggregative Tracking Feedback

Reduced optimization problem

$$\min_{u_1,\dots,u_N} \sum_{i=1}^{N} f_i(h_i(u_i), \sigma(h(u)))$$

with $p_i(h_i(u_i), u_i) = 0$ and $x_i = h_i(u_i)$ globally exponentially stable.

$$\dot{u}_i = -\delta_1 \nabla h_i(u_i) \left(\nabla_1 f_i(x_i, \sigma(x)) + \frac{\nabla \phi(x_i)}{N} \sum_{j=1}^{N} \nabla_2 f_j(x_j, \sigma(x)) \right)$$

Issue 2: this law uses unavailable global information

Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Aggregative Tracking Feedback

Reduced optimization problem

$$\min_{u_1,\dots,u_N} \sum_{i=1}^{N} f_i(h_i(u_i), \sigma(h(u)))$$

with $p_i(h_i(u_i), u_i) = 0$ and $x_i = h_i(u_i)$ globally exponentially stable.

$$\dot{u}_i = -\delta_1 \nabla h_i(u_i)\Big(\nabla_1 f_i(x_i, \underbrace{\phi_i(x_i) + w_i}_{\text{proxy for } \sigma(x)}) + \nabla \phi_i(x_i)(\underbrace{\nabla_2 f_i(x_i, \phi_i(x_i) + w_i) + z_i}_{\text{proxy for } \frac{1}{N}\sum_{j=1}^{N} \nabla_2 f_j(x_j, \sigma(x))})\Big)$$

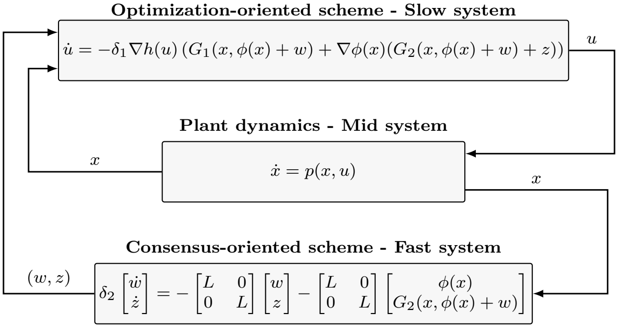
$$\delta_2 \dot{w}_i = -\sum_{j \in \mathcal{N}_i} a_{ij}(w_i - w_j) - \sum_{j \in \mathcal{N}_i} a_{ij}(\phi_i(x_i) - \phi_j(x_j))$$

$$\delta_2 \dot{z}_i = -\sum_{j \in \mathcal{N}_i} a_{ij}(z_i - z_j) - \sum_{j \in \mathcal{N}_i} a_{ij}(\nabla_2 f_i(x_i, w_i + \phi_i(x_i)) - \nabla_2 f_j(x_j, w_j + \phi_j(x_j)))$$

Remark: $\delta_1 > 0$ slows down $\dot{u}_i$ and $\delta_2 > 0$ speeds up $\dot{w}_i$ and $\dot{z}_i$

Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Overall Closed-loop System... Three Timescales



- x, w, z, ϕ, G_1, G_2, p stacking local quantities x_i, w_i, z_i, ϕ_i, $\nabla_1 f_i$, $\nabla_2 f_i$, and p_i
- $L := \mathcal{L} \otimes I_d$ with $\mathcal{L}$ Laplacian matrix of the graph $\mathcal{G}$

Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, 2024)

Convergence Properties of Aggregative Tracking Feedback

Theorem

Consider Aggregative Tracking Feedback. If

- Graph $\mathcal{G}$ strongly connected and weight-balanced
- $x_i = h_i(u_i)$ globally exponentially stable uniformly in u_i
- $\sum_{i=1}^{N} f_i(\cdot, \sigma(\cdot))$ radially unbounded, $\nabla_1 f_i$, $\nabla_2 f_i$, and ϕ_i Lipschitz continuous (possibly nonconvex)
- Initialization $\sum_{i=1}^{N} w_i(0) = \sum_{i=1}^{N} z_i(0) = 0$

Then, for small $\delta_1 > 0$ and $\delta_2 > 0$

$$\lim_{t\to\infty} \text{DIST}\left(\begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \mathcal{X}^\star\right)$$

with $\mathcal{X}^\star$ the set of stationary points of the aggregative optimization problem.

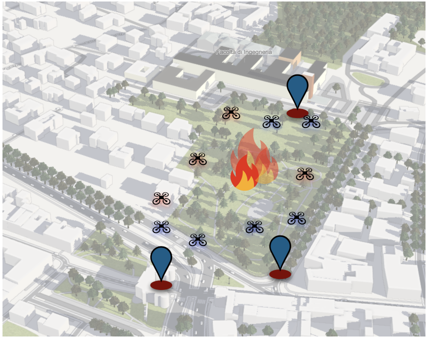
Carnevale, Mimmo, Notarstefano, "Nonconvex Distributed Feedback Optimization for Aggregative Cooperative Robotics." (Automatica, '24)

Multi-robot Monitoring via Distributed Feedback Optimization

Distributed Optimization Setup

- Network of N robots monitoring an area
- Target to cordone
- Set of M points of interest to monitor
- Monitoring and Cordoning strategy modelled by Aggregative Feedback optimization setup

$$\min_{x,u} \sum_{i=1}^{N} f_i^k(x_i, \sigma(x))$$
$$\text{subj.to } x_i = h_i(u_i), \quad \forall i \in \{1, \ldots, N\}$$

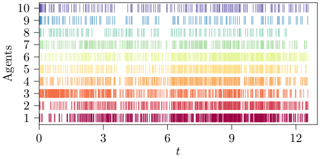


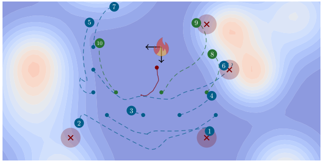
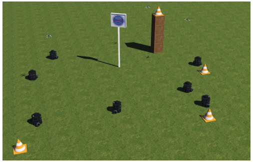
Pichierri, Carnevale, Sforni, Notarstefano "Multi-Robot Target Monitoring and Encirclement via Triggered Distributed Feedback Optimization." (submitted to journal, https://arxiv.org/pdf/2409.20399)

Numerical Experiments in ROS 2 Simulator

Realistic simulation of multi-robot monitoring and cordoning experiment[a]:

- ROS 2 implementation with ChoiRbot
- Ground team: $N = 10$ Turtlebot 3 Burger
- A fixed target to cordone
- $M = 5$ points of interest to monitor



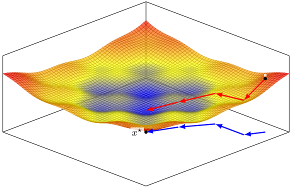



[a] https://www.youtube.com/watch?v=iIUChcNUdr4

System Theory Tools for Optimization and Learning

System Theory Tools

LaSalle and Lyapunov Theory



Timescale Separation

Slow System

$$x^{k+1} = x^k + \delta f(x^k, z^k)$$

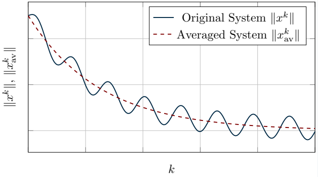
Fast System

$$z^{k+1} = g(z^k, x^k)$$

z^k (fast system output, fed to the slow system)

x^k (slow system output, fed to the fast system)

Averaging theory



Distributed optimization algorithms

Optimization-oriented dynamics

Consensus-oriented dynamics

Accelerated optimization algorithms

Gradient-like algorithm

Extra dynamics

Learning-driven Optimal Control

Learning Algorithm

Physical Plant

Accelerated Gradient Schemes

Consider the unconstrained program

$$\min_{x\in\mathbb{R}^n} \; f(x)$$

Acclerated gradient schemes typically enjoy this second-order form

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \gamma\nabla f\big(\mathrm{x}^k + \beta_1(\mathrm{x}^k - \mathrm{x}^{k-1})\big) + \beta_2(\mathrm{x}^k - \mathrm{x}^{k-1})$$

Method	β_1	β_2
Gradient Descent	0	0
Heavy-Ball	0	β
Nesterov	β	β
Triple Momentum	β_1	β_2

Carnevale, Notarnicola, Notarstefano, "Timescale Separation for Nonconvex Accelerated Optimization." (arXiv soon)

Accelerated Gradient Schemes

Consider the unconstrained program

$$\min_{x \in \mathbb{R}^n} \; f(x)$$

Acclerated gradient schemes typically enjoy this second-order form

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \gamma \nabla f\big(\mathrm{x}^k + \beta_1(\mathrm{x}^k - \mathrm{x}^{k-1})\big) + \beta_2(\mathrm{x}^k - \mathrm{x}^{k-1})$$

Method	β_1	β_2
Gradient Descent	0	0
Heavy-Ball	0	β
Nesterov	β	β
Triple Momentum	β_1	β_2

In state-space form

$$\begin{aligned}
\mathrm{x}^{k+1} &= \mathrm{x}^k - \gamma \nabla f\big(\mathrm{x}^k + \beta_1(\mathrm{x}^k - \mathrm{z}^k)\big) + \beta_2(\mathrm{x}^k - \mathrm{z}^k) \\
\mathrm{z}^{k+1} &= \mathrm{x}^k
\end{aligned}$$

Carnevale, Notarnicola, Notarstefano, "Timescale Separation for Nonconvex Accelerated Optimization." (arXiv soon)

Accelerated Gradient Schemes

Consider the unconstrained program

$$\min_{x \in \mathbb{R}^n} \; f(x)$$

Acclerated gradient schemes typically enjoy this second-order form

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \gamma \nabla f\big(\mathrm{x}^k + \beta_1(\mathrm{x}^k - \mathrm{x}^{k-1})\big) + \beta_2(\mathrm{x}^k - \mathrm{x}^{k-1})$$

Method	β_1	β_2
Gradient Descent	0	0
Heavy-Ball	0	β
Nesterov	β	β
Triple Momentum	β_1	β_2

In state-space form

$$\begin{aligned} \mathrm{x}^{k+1} &= \mathrm{x}^k - \gamma \nabla f\big(\mathrm{x}^k + \beta_1(\mathrm{x}^k - \mathrm{z}^k)\big) + \beta_2(\mathrm{x}^k - \mathrm{z}^k) \\ \mathrm{z}^{k+1} &= \mathrm{x}^k \end{aligned}$$

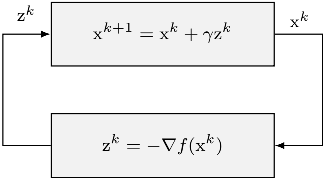
Idea: frame accelerated gradient schemes as two-time-scale systems

Carnevale, Notarnicola, Notarstefano, "Timescale Separation for Nonconvex Accelerated Optimization." (arXiv soon)

Accelerated Gradient Schemes: System Theory Perspective

Gradient method: static feedback

$$x^{k+1} = x^k - \gamma z^k$$
$$z^k = -\nabla f(x^k)$$

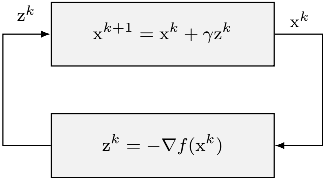


Carnevale, Notarnicola, Notarstefano, “Timescale Separation for Nonconvex Accelerated Optimization.” (arXiv soon)

Accelerated Gradient Schemes: System Theory Perspective

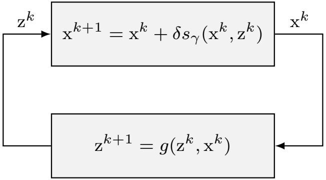
Gradient method: static feedback

$$\begin{aligned} x^{k+1} &= x^k - \gamma z^k \\ z^k &= -\nabla f(x^k) \end{aligned}$$



Accelerated methods: dynamic feedback

$$\begin{aligned} x^{k+1} &= x^k + \delta s_\gamma(x^k, z^k) \\ z^{k+1} &= g(x^k, z^k) \end{aligned}$$



Carnevale, Notarnicola, Notarstefano, "Timescale Separation for Nonconvex Accelerated Optimization." (arXiv soon)

Accelerated Gradient Schemes: Meta-Algorithm Requirements

Meta algorithm for accelerated gradient scheme as the interconnected system

$$\mathrm{x}^{k+1} = \mathrm{x}^k + \delta s_\gamma(\mathrm{x}^k, \mathrm{z}^k)$$
$$\mathrm{z}^{k+1} = g(\mathrm{x}^k, \mathrm{z}^k)$$

Requirements:

- g admits equilibria parametrized in x via $z_{\mathsf{eq}}(x)$, namely for all $x \in \mathbb{R}^n$

$$z_{\mathsf{eq}}(x) = g(x, z_{\mathsf{eq}}(x))$$

- s_γ recovers gradient descent for $z = z_{\mathsf{eq}}(x)$, namely

$$s_\gamma(x, z_{\mathsf{eq}}(x)) = -\gamma \nabla f(x)$$

Carnevale, Notarnicola, Notarstefano, "Timescale Separation for Nonconvex Accelerated Optimization." (arXiv soon)

Accelerated Gradient Schemes: Convergence Properties

Theorem

Consider the meta accelerated gradient scheme. Assume

- f radially unbounded (possibly nonconvex) with L-Lipschitz continuous gradient
- $s_\gamma, g, z_{\text{eq}}$ Lipschitz continuous
- $z_{\text{eq}}(x)$ exponentially stable equilibrium for extra-dynamics $\mathrm{z}^{k+1} = g(x, \mathrm{z}^k)$ uniformly in x
- s_γ recovers gradient descent for $z = z_{\text{eq}}(x)$, i.e., $s_\gamma(x, z_{\text{eq}}(x)) = -\gamma \nabla f(x)$

Then, for any given $\gamma \in (0, 2/L)$ and a sufficiently small $\delta > 0$, it holds

$$\lim_{k \to \infty} \text{DIST}(\mathrm{x}^k, \mathcal{X}^\star) = 0$$

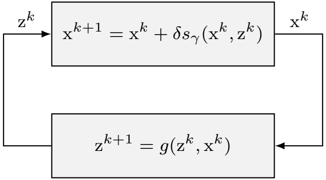
with $\mathcal{X}^\star$ set of stationary points of the optimization problem.

Moreover, if f is strongly convex, then the convergence is linear.

Carnevale, Notarnicola, Notarstefano, "Timescale Separation for Nonconvex Accelerated Optimization." (arXiv soon)

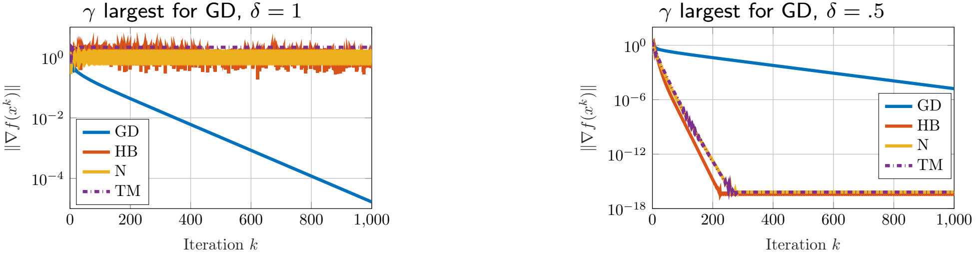
Accelerated Gradient Schemes: Examples

$$x^{k+1} = x^k + \delta s_\gamma(x^k, z^k)$$
$$z^{k+1} = x^k$$



- Heavy ball: $s_\gamma(x^k, z^k) = -\gamma \nabla f(x^k) + (x^k - z^k)$
- Nesterov: $s_\gamma(x^k, z^k) = -\gamma \nabla f\big(x^k + \delta(x^k - z^k)\big) + (x^k - z^k)$
- Triple momentum: $s_\gamma(x^k, z^k) = -\gamma \nabla f\big(x^k + \beta_1(x^k - z^k)\big) + (x^k - z^k)$

Nonconvex logistic regression



Carnevale, Notarnicola, Notarstefano, “Timescale Separation for Nonconvex Accelerated Optimization.” (arXiv soon)

Model-free Optimization

Consider an unconstrained program where f is unknown but we can query an oracle providing cost evaluations

$$\min_{x \in \mathbb{R}^n} \; f(x)$$

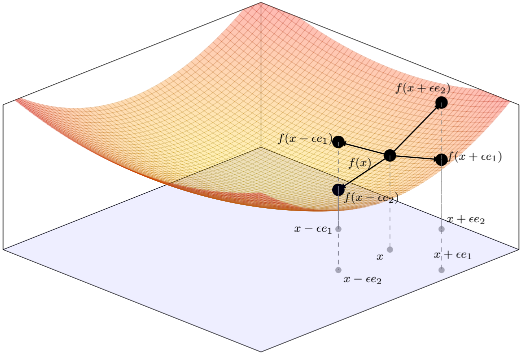
Carnevale, Notarstefano, "Accelerating Model-Free Optimization via Averaging of Cost Samples." (submitted)

Model-free Optimization

Consider an unconstrained program where f is unknown but we can query an oracle providing cost evaluations

$$\min_{x \in \mathbb{R}^n} \; f(x)$$

Existing approaches rely on estimating the gradient of f via finite differences



Carnevale, Notarstefano, "Accelerating Model-Free Optimization via Averaging of Cost Samples." (submitted)

Model-free Optimization

Suppose to have a generic gradient estimation technique relying on a finite set of perturbing directions $\{\mathrm{d}_j\}_{j=1}^{D}$

$$\sum_{j=1}^{D} g_\epsilon(f(x + \epsilon \mathrm{d}_j), \mathrm{d}_j) = \nabla f(x) + e_\epsilon(x)$$

where $e_\epsilon(x)$ is an error term tunable via $\epsilon > 0$, namely for any compact $\mathcal{S} \subset \mathbb{R}^n$, there exists $L > 0$ such that

$$\|e_\epsilon(x)\| \leq \epsilon L$$

for all $x \in \mathcal{S}$.

Carnevale, Notarstefano, "Accelerating Model-Free Optimization via Averaging of Cost Samples." (submitted)

Model-free Optimization

Suppose to have a generic gradient estimation technique relying on a finite set of perturbing directions $\{\mathrm{d}_j\}_{j=1}^{D}$

$$\sum_{j=1}^{D} g_\epsilon(f(x+\epsilon \mathrm{d}_j), \mathrm{d}_j) = \nabla f(x) + e_\epsilon(x)$$

where $e_\epsilon(x)$ is an error term tunable via $\epsilon > 0$, namely for any compact $\mathcal{S} \subset \mathbb{R}^n$, there exists $L > 0$ such that

$$\|e_\epsilon(x)\| \leq \epsilon L$$

for all $x \in \mathcal{S}$.

Example: finite differences using the canonical basis $\{e_j, -e_j\}_{j=1}^{n}$ as perturbing directions, namely

$$g_\epsilon(f(x+\epsilon e_j), e_j) = \frac{f(x+\epsilon e_j)}{2\epsilon} e_j \implies \sum_{j=1}^{D} g_\epsilon(f(x+\epsilon \mathrm{d}_j), \mathrm{d}_j) = \sum_{j=1}^{n} \frac{f(x+\epsilon e_j) - f(x-\epsilon e_j)}{2\epsilon} e_j$$

Carnevale, Notarstefano, "Accelerating Model-Free Optimization via Averaging of Cost Samples." (submitted)

Model-free Optimization

Suppose to have a generic gradient estimation technique relying on a finite set of perturbing directions $\{\mathrm{d}_j\}_{j=1}^{D}$

$$\sum_{j=1}^{D} g_\epsilon(f(x+\epsilon \mathrm{d}_j), \mathrm{d}_j) = \nabla f(x) + e_\epsilon(x)$$

where $e_\epsilon(x)$ is an error term tunable via $\epsilon > 0$, namely for any compact $\mathcal{S} \subset \mathbb{R}^n$, there exists $L > 0$ such that

$$\|e_\epsilon(x)\| \leq \epsilon L$$

for all $x \in \mathcal{S}$.

Example: finite differences using the canonical basis $\{e_j, -e_j\}_{j=1}^{n}$ as perturbing directions, namely

$$g_\epsilon(f(x+\epsilon e_j), e_j) = \frac{f(x+\epsilon e_j)}{2\epsilon} e_j \implies \sum_{j=1}^{D} g_\epsilon(f(x+\epsilon \mathrm{d}_j), \mathrm{d}_j) = \sum_{j=1}^{n} \frac{f(x+\epsilon e_j) - f(x-\epsilon e_j)}{2\epsilon} e_j$$

Multiple cost evaluations per iteration unwanted or not possible

Carnevale, Notarstefano, “Accelerating Model-Free Optimization via Averaging of Cost Samples.” (submitted)

Model-free Meta-Algorithm Design

Algorithm with multiple cost evaluations per iteration

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \gamma \underbrace{\sum_{j=1}^{D} g_\epsilon(f(\mathrm{x}^k + \epsilon \mathrm{d}_j), \mathrm{d}_j)}_{\approx \nabla f(\mathrm{x}^k)}$$

Carnevale, Notarstefano, "Accelerating Model-Free Optimization via Averaging of Cost Samples." (submitted)

Model-free Meta-Algorithm Design

Algorithm with multiple cost evaluations per iteration

$$x^{k+1} = x^k - \gamma \underbrace{\sum_{j=1}^{D} g_\epsilon(f(x^k + \epsilon d_j), d_j)}_{\text{not simultaneously available}}$$

Carnevale, Notarstefano, "Accelerating Model-Free Optimization via Averaging of Cost Samples." (submitted)

Model-free Meta-Algorithm Design

Algorithm with multiple cost evaluations per iteration

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \gamma \underbrace{\sum_{j=1}^{D} g_\epsilon(f(\mathrm{x}^k + \epsilon \mathrm{d}_j), \mathrm{d}_j)}_{\text{not simultaneously available}}$$

Design based on timescale separation

Introduce memory mechanism based on auxiliary variables $\{z_j\}_{j=1}^D$ storing cost evaluations when available

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \gamma \sum_{j=1}^{D} g_\epsilon(\mathrm{z}_j^k, \mathrm{d}_j) \qquad \text{slow dynamics}$$

$$\mathrm{z}_j^{k+1} = \begin{cases} f(\mathrm{x}^k + \epsilon \mathrm{d}_j) & \text{if direction } \mathrm{d}_j \text{ is selected} \\ \mathrm{z}_j^k & \text{otherwise} \end{cases} \qquad \text{fast dynamics}$$

Note: cost evaluations per iteration do not increase with respect to a naive zeroth-order method

Carnevale, Notarstefano, "Accelerating Model-Free Optimization via Averaging of Cost Samples." (submitted)

Model-Free Meta-Algorithm: Convergence Properties

Theorem

Consider our Model-Free Meta-Algorithm. Assume

- f strongly convex with Lipschitz continuous gradient
- generic gradient estimation technique g_ϵ available
- perturbing directions $\{\mathrm{d}_j\}_{j=1}^{D}$ selected via essentially cyclic rule

Then, for all $\rho > 0$ and $(x^0, z^0) \in \mathbb{R}^n \times \mathbb{R}^D$, for small $\gamma > 0$ and $\epsilon > 0$, there exists $a \in (0,1)$ such that

$$\left\| \mathrm{x}^k - x^\star \right\| \leq (1 - \gamma a)^k a_0 + \rho$$

for all $k \in \mathbb{N}$, where $x^\star$ is the optimal solution.

Proof idea: combine timescale separation with practical stability analysis tools

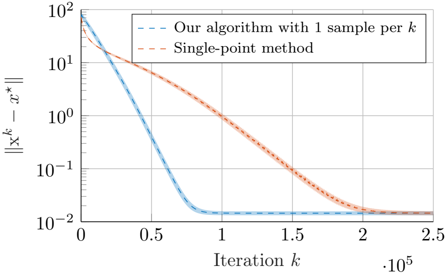
Carnevale, Notarstefano, "Accelerating Model-Free Optimization via Averaging of Cost Samples." (submitted)

Model-Free Meta-Algorithm: Numerical Simulations

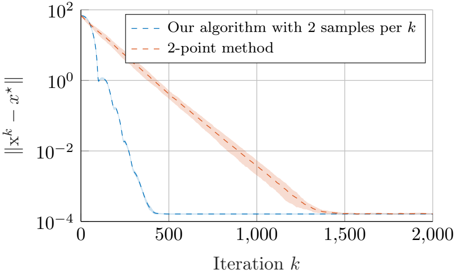
Logistic regression problem

$$\min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{h=1}^{m} \log\left(1 + e^{-l_h(x^\top p_h)}\right) + \frac{C}{2} \|x\|^2 ,$$

Single-point methods



Two-point methods

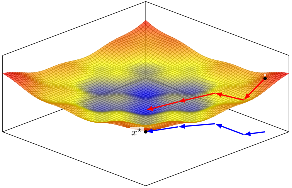


Carnevale, Notarstefano, "Accelerating Model-Free Optimization via Averaging of Cost Samples." (submitted)

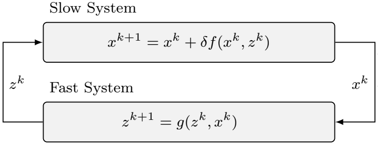
System Theory Tools for Optimization and Learning

System Theory Tools

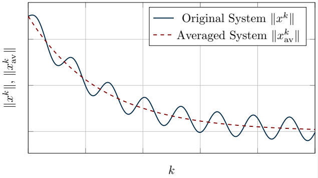
LaSalle and Lyapunov Theory



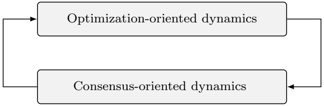
Timescale Separation



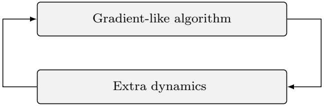
Averaging theory



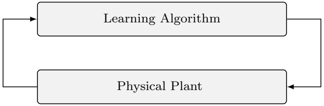
Distributed optimization algorithms



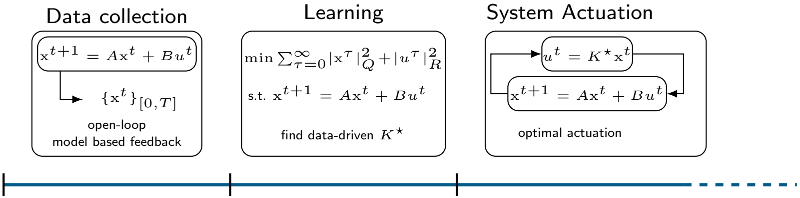
Accelerated optimization algorithms



Learning-driven Optimal Control



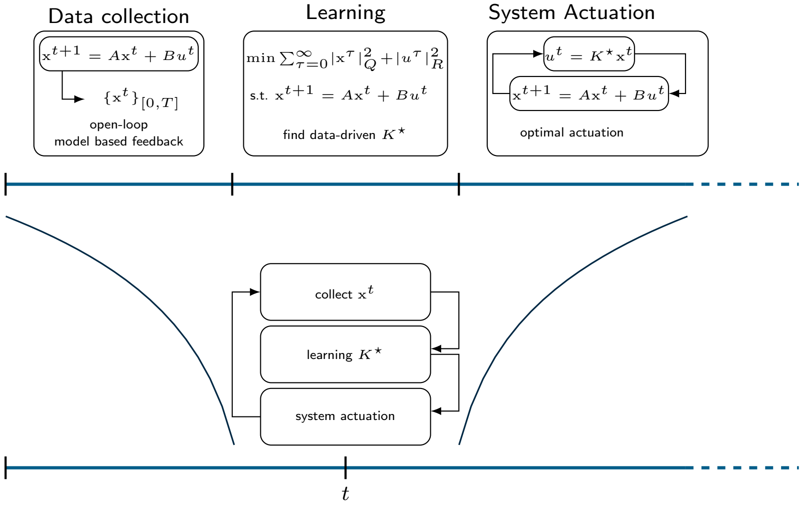
Data-driven Linear Quadratic Regulator: from Off-policy to On-policy



Off policy

data-driven LQR

Data-driven Linear Quadratic Regulator: from Off-policy to On-policy



Off policy
data-driven LQR

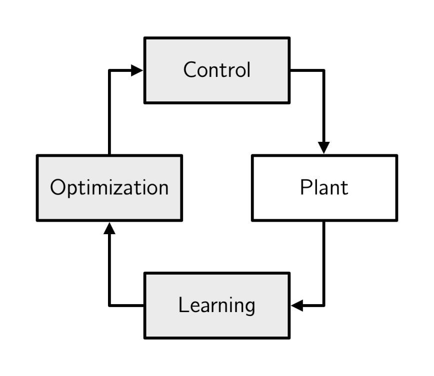
On policy
data-driven LQR

Stability-Certified On Policy Data-driven LQR

LQR problem For a linear system with unknown A and B, solve

$$\begin{aligned}\min_{\substack{x^1, x^2, \ldots \\ u^0, u^1, \ldots}} \quad & \frac{1}{2}\sum_{\tau=0}^{\infty}\Big((x^\tau)^\top Q x^\tau + (u^\tau)^\top R u^\tau\Big) \\ \text{subj.to} \quad & x^{t+1} = Ax^t + Bu^t \\ & x^0 \sim \mathcal{X}^0\end{aligned}$$

with $x^k \in \mathbb{R}^n$, $u^k \in \mathbb{R}^m$, $\mathcal{X}^0$ known, $Q = Q^\top > 0$, $R = R^\top > 0$.



Stability-Certified On Policy Data-driven LQR

LQR problem For a linear system with unknown A and B, solve

$$\begin{aligned}\min_{\substack{x^1, x^2, \ldots \\ u^0, u^1, \ldots}} \quad & \frac{1}{2} \sum_{\tau=0}^{\infty} \Big((x^\tau)^\top Q x^\tau + (u^\tau)^\top R u^\tau \Big) \\ \text{subj.to} \quad & x^{t+1} = A x^t + B u^t \\ & x^0 \sim \mathcal{X}^0\end{aligned}$$

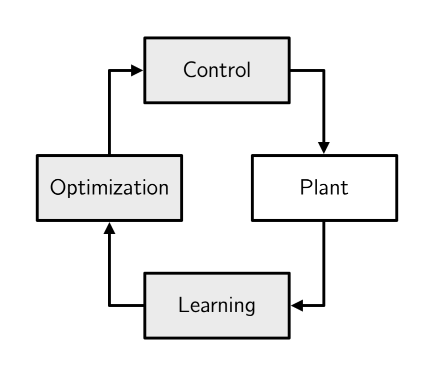
with $x^k \in \mathbb{R}^n$, $u^k \in \mathbb{R}^m$, $\mathcal{X}^0$ known, $Q = Q^\top > 0$, $R = R^\top > 0$.

On-policy framework

Learning: sample data from real system

Optimization: iteratively refine feedback gain policy

Control: actuate real system with tentative (non-optimal) policy

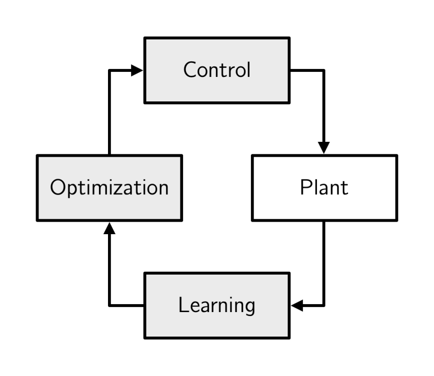


Stability-Certified On Policy Data-driven LQR

LQR problem For a linear system with unknown A and B, solve

$$\min_{\substack{x^1, x^2, \ldots \\ u^0, u^1, \ldots}} \ \frac{1}{2} \sum_{\tau=0}^{\infty} \Big((x^\tau)^\top Q x^\tau + (u^\tau)^\top R u^\tau \Big)$$
$$\text{subj.to } x^{t+1} = A x^t + B u^t$$
$$x^0 \sim \mathcal{X}^0$$

with $x^k \in \mathbb{R}^n$, $u^k \in \mathbb{R}^m$, $\mathcal{X}^0$ known, $Q = Q^\top > 0$, $R = R^\top > 0$.



On-policy framework

- Learning: sample data from real system
- Optimization: iteratively refine feedback gain policy
- Control: actuate real system with tentative (non-optimal) policy

Stability certificate guarantee whole closed-loop learning-optimization-control system asymptotically stable

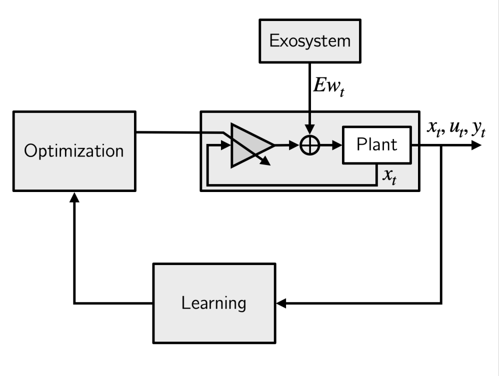
RELEARN LQR: On Policy Strategy

Learning

$$S^{k+1} = \lambda S^k + \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix} \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix}^\top$$

$$H^{k+1} = \lambda H^k + \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix} \mathrm{y}^k$$

$$\theta^{k+1} = \theta^k - \gamma\,(H^k)^\dagger (H^k \theta^k - S^k\,)$$



Optimization

$$K^{k+1} = K^k - \gamma\,\underbrace{G(K^k, \theta^k)}_{\text{data-driven gradient}}$$

Control

$$\mathrm{w}^{k+1} = F\mathrm{w}^k$$

$$\mathrm{u}^k = K^k \mathrm{x}^k + E\mathrm{w}^k$$

$$\mathrm{x}^{k+1} = A\mathrm{x}^k + B\mathrm{u}^k$$

$$\mathrm{y}^k = (\mathrm{x}^{k+1})^\top$$

Sforni, Carnevale, Notarnicola, Notarstefano, "Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient." (submitted to Automatica)

RELEARN LQR: On Policy Strategy

Learning – Recursive Least Square with $\theta^k = [A^k \; B^k]^\top$

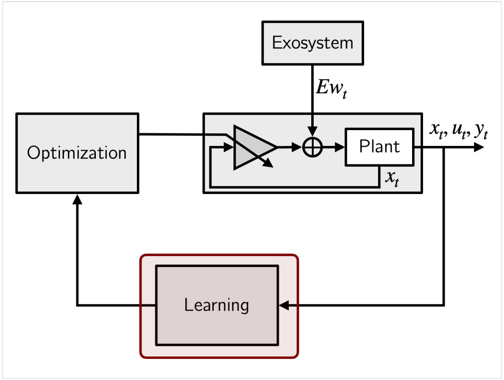
$$\boxed{\begin{aligned} S^{k+1} &= \lambda S^k + \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix} \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix}^\top \\ H^{k+1} &= \lambda H^k + \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix} \mathrm{y}^k \\ \theta^{k+1} &= \theta^k - \gamma\,(H^k)^\dagger (H^k \theta^k - S^k) \end{aligned}}$$

Optimization

$$K^{k+1} = K^k - \gamma \underbrace{G(K^k, \theta^k)}_{\text{data-driven gradient}}$$

Control

$$\begin{aligned} \mathrm{w}^{k+1} &= F \mathrm{w}^k \\ \mathrm{u}^k &= K^k \mathrm{x}^k + E \mathrm{w}^k \\ \mathrm{x}^{k+1} &= A \mathrm{x}^k + B \mathrm{u}^k \\ \mathrm{y}^k &= (\mathrm{x}^{k+1})^\top \end{aligned}$$



Sforni, Carnevale, Notarnicola, Notarstefano, "Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient." (submitted to Automatica)

RELEARN LQR: On Policy Strategy

Learning – Recursive Least Square with $\theta^k = [A^k \ B^k]^\top$

$$S^{k+1} = \lambda S^k + \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix} \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix}^\top$$

$$H^{k+1} = \lambda H^k + \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix} \mathrm{y}^k$$

$$\theta^{k+1} = \theta^k - \gamma\,(H^k)^\dagger(H^k\theta^k - S^k\,)$$

Optimization – Policy Gradient

$$\boxed{K^{k+1} = K^k - \gamma\, \underbrace{G(K^k, \theta^k)}_{\text{data-driven gradient}}}$$

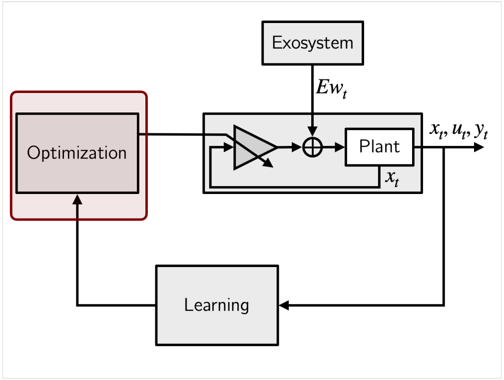
Control

$$\mathrm{w}^{k+1} = F\mathrm{w}^k$$

$$\mathrm{u}^k = K^k\mathrm{x}^k + E\mathrm{w}^k$$

$$\mathrm{x}^{k+1} = A\mathrm{x}^k + B\mathrm{u}^k$$

$$\mathrm{y}^k = (\mathrm{x}^{k+1})^\top$$



Sforni, Carnevale, Notarnicola, Notarstefano, "Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient." (submitted to Automatica)

RELEARN LQR: On Policy Strategy

Learning – Recursive Least Square with $\theta^k = [A^k \; B^k]^\top$

$$S^{k+1} = \lambda S^k + \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix} \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix}^\top$$

$$H^{k+1} = \lambda H^k + \begin{bmatrix} \mathrm{x}^k \\ \mathrm{u}^k \end{bmatrix} \mathrm{y}^k$$

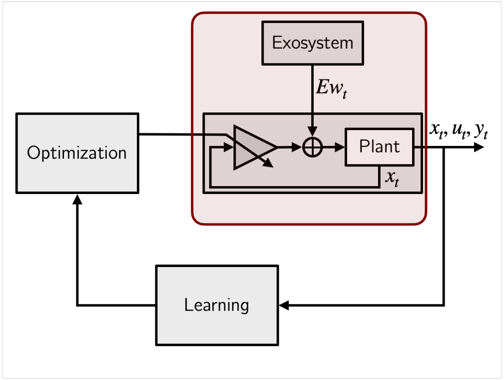
$$\theta^{k+1} = \theta^k - \gamma \, (H^k)^\dagger (H^k \theta^k - S^k \,)$$

Optimization – Policy Gradient

$$K^{k+1} = K^k - \gamma \, \underbrace{G(K^k, \theta^k)}_{\text{data-driven gradient}}$$

Control – Dynamics + Exogenous System (Persist. Excit.)

$$\boxed{\begin{aligned} \mathrm{w}^{k+1} &= F\mathrm{w}^k \\ \mathrm{u}^k &= K^k \mathrm{x}^k + E\mathrm{w}^k \\ \mathrm{x}^{k+1} &= A\mathrm{x}^k + B\mathrm{u}^k \\ \mathrm{y}^k &= (\mathrm{x}^{k+1})^\top \end{aligned}}$$



Sforni, Carnevale, Notarnicola, Notarstefano, "Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient." (submitted to Automatica)

Closed-loop System: Convergence Properties

Theorem

Let $(A^\star, B^\star)$ (unknown matrices) controllable and exosystem satisfy proper persistency of excitation conditions.

For any K^0 s.t. $A^0 + B^0 K^0$, there exist $(\Pi_{\mathrm{x}}, \Pi_H, \Pi_S)$ and $\bar{\gamma} > 0$ such that, for all $\gamma \in (0, \bar{\gamma})$, it holds

$$\mathrm{x}^k \xrightarrow{k \to \infty} \Pi_{\mathrm{x}} \mathrm{w}^k$$

$$\begin{bmatrix} H^k \\ S^k \end{bmatrix} \xrightarrow{k \to \infty} \begin{bmatrix} \mathrm{vec}^{-1}(\Pi_H W^k) \\ \mathrm{vec}^{-1}(\Pi_S W^k) \end{bmatrix}$$

$$\begin{bmatrix} \theta^k \\ K^k \end{bmatrix} \xrightarrow{k \to \infty} \begin{bmatrix} \theta^\star \\ K^\star \end{bmatrix}$$

with geometric rate, where $W^k := \textsc{vec}\left\{\mathrm{w}^k (\mathrm{w}^k)^\top\right\}$

Sforni, Carnevale, Notarnicola, Notarstefano, "Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient." (submitted to Automatica)

Sketch of the Proof (I)

Analyze overall (learning-optimization-control) closed-loop system

$$
\begin{aligned}
\mathrm{w}^{k+1} &= F\mathrm{w}^k \\
\mathrm{x}^{k+1} &= A\mathrm{x}^k + BK^k\mathrm{x}^k + BE\mathrm{w}^k \\
S^{k+1} &= \lambda S^k + \begin{bmatrix} \mathrm{x}^k \\ K^k\mathrm{x}^k + E\mathrm{w}^k \end{bmatrix} \begin{bmatrix} \mathrm{x}^k \\ K^k\mathrm{x}^k + E\mathrm{w}^k \end{bmatrix}^\top \\
H^{k+1} &= \lambda H^k + \begin{bmatrix} \mathrm{x}^k \\ K^k\mathrm{x}^k + E\mathrm{w}^k \end{bmatrix} x_{t+1}^\top \\
\theta^{k+1} &= \theta^k - \gamma(H^k)^\dagger(H^k\theta^k - S^k) \\
K^{k+1} &= K^k - \gamma\, G(K^k, \theta^k)
\end{aligned}
$$

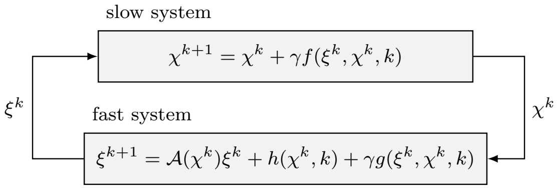
Highlight a two-time-scale interconnected system

Exploit averaging theory to analyze the asymptotic stability properties

Sforni, Carnevale, Notarnicola, Notarstefano, "Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient." (submitted to Automatica)

Sketch of the Proof (II)

In error coordinates, the overall closed-loop system can be seen as two feedback interconnected subsystems



Sforni, Carnevale, Notarnicola, Notarstefano, "Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient." (submitted to Automatica)

Sketch of the Proof (II)

In error coordinates, the overall closed-loop system can be seen as two feedback interconnected subsystems

slow system

$$\chi^{k+1} = \chi^k + \gamma f(\xi^k, \chi^k, k)$$

ξ^k $\qquad$ χ^k

fast system

$$\xi^{k+1} = \mathcal{A}(\chi^k)\xi^k + h(\chi^k, k) + \gamma g(\xi^k, \chi^k, k)$$

with

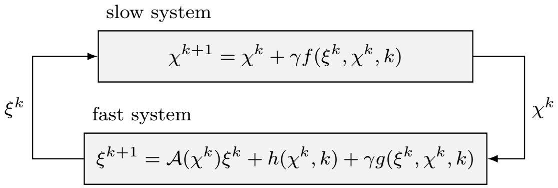
$$\chi := \begin{bmatrix} \tilde{\theta} \\ \tilde{K} \end{bmatrix} := \begin{bmatrix} \theta - \theta^\star \\ K - K^\star \end{bmatrix}, \qquad \xi := \begin{bmatrix} \mathrm{x} - \Pi \mathrm{w}^k \\ \gamma \left(H - \mathrm{vec}^{-1}(\Pi_H W^k) \right) \\ \gamma \left(S - \mathrm{vec}^{-1}(\Pi_S W^k) \right) \end{bmatrix}, \qquad \mathcal{A}(\chi^k) := \begin{bmatrix} A + B(\tilde{K} + K^\star) & 0 & 0 \\ 0 & \lambda I & 0 \\ 0 & 0 & \lambda I \end{bmatrix}$$

while f, h, and g properly contain the terms describing the overall dynamics in the new coordinates

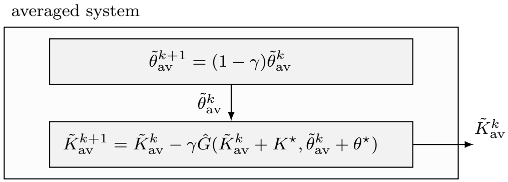
Sforni, Carnevale, Notarnicola, Notarstefano, "Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient." (submitted to Automatica)

Sketch of the Proof (II)

In error coordinates, the overall closed-loop system can be seen as two feedback interconnected subsystems



Exploit averaging theory by studying the *averaged system*



Sforni, Carnevale, Notarnicola, Notarstefano, "Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient." (submitted to Automatica)

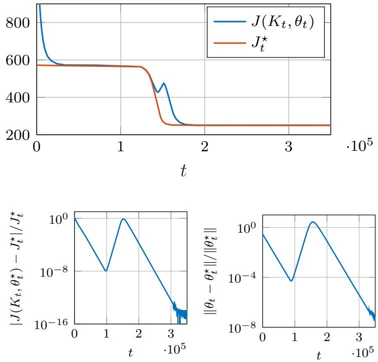
Simulations on a Linearized Aircraft Model

Linearized aircraft model

Slowly varying matrices $A^\star$ and $B^\star$

Apply RELEARN LQR strategy

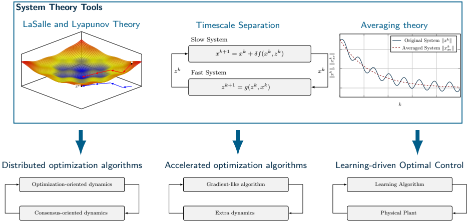
No re-initialization required



Conclusions

Summary

- System theory tools for algorithms' design and analysis (algorithms as dynamical systems)
- Systematic design of distributed optimization algorithms
- Accelerated optimization as feedback interconnection
- Data-Driven LQR: online on-policy certified stability



Some references

- A Unifying System Theory Framework for Distributed Optimization and Games
  `https://ieeexplore.ieee.org/document/11015566`
- Stability-Certified On-Policy Data-Driven LQR via Recursive Learning and Policy Gradient
  `https://arxiv.org/pdf/2403.05367`