

Asynchronous Distributed Gradient Descent with Exponential Convergence Rate via Hybrid Methods

Katherine Hendrickson*, Dawn Hustig-Schultz†, Matthew Hale*, Ricardo Sanfelice†

* Department of Mechanical and Aerospace Engineering, University of Florida

† Department of Electrical and Computer Engineering, University of California

AFOSR Center of Excellence Review

April 30th, 2021





- Problem

- We are interested in solving convex optimization problems in a distributed, asynchronous way.

Problem Statement

Given an objective function $L : \mathbb{R}^n \rightarrow \mathbb{R}$,

$$\text{minimize } L(x), \quad x \in \mathbb{R}^n$$

across N agents while requiring

- (i) only one agent updates any entry of the decision variable x , and
- (ii) agents require only sporadic information sharing from others.

- Applications

Machine Learning



Robotics



Networks

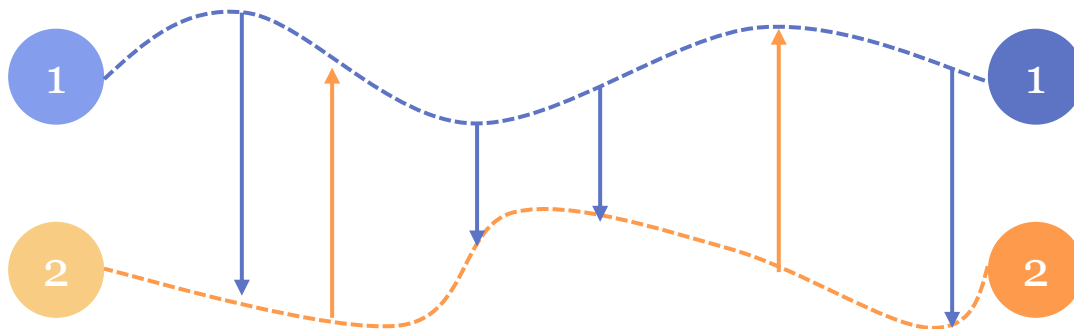


Communications





- Real-world problems require a hybrid approach.
 - Robotics is one example where physical motion occurs in continuous time while communications occur at specific instants in time.
 - To account for the sporadic nature of communications, communication events are modeled using discrete time.



- A hybrid systems framework provides advantages during analysis.
 - The framework comes with many tools for showing stability and convergence.
 - Hybrid systems also tend to provide robustness, which is especially beneficial in asynchronous or contested environments.

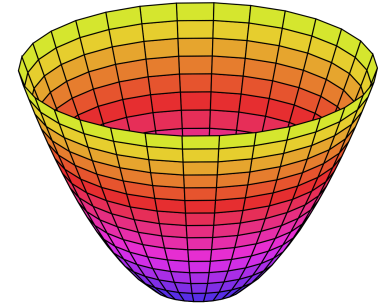


- We impose the following assumption on the objective function L .

Assumption 1

The function L is

- twice continuously differentiable,
- β -strongly convex for some positive β ,
- and K -smooth (∇L is K -Lipschitz).




- This is a common assumption in multiagent optimization that allows us to consider a large number of convex problems.
- For example, convex quadratic problems where β and K are easily determined.




- In the centralized case, we would use $\dot{x} = -\nabla L(x)$.
- We make this multi-agent where agent i is responsible for updating the i -th block of x , denoted by $x_i \in \mathbb{R}^{n_i}$. With constant communications, agent i would then have the update $\dot{x}_i = -\nabla_i L(x)$, where $\nabla_i := \frac{\partial}{\partial x_i}$.
- To account for sporadic communications, agents store the most recently communicated values in a separate variable $\eta \in \mathbb{R}^n$.
- We then implement a “sample and hold” methodology whose dynamics take the form $\dot{x}_i = -\nabla_i L(\eta)$.

$$\dot{x} = -\nabla L(x) \rightarrow \begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_i \\ \vdots \\ \dot{x}_N \end{pmatrix} = \begin{pmatrix} -\nabla_1 L(\eta) \\ \vdots \\ -\nabla_i L(\eta) \\ \vdots \\ -\nabla_N L(\eta) \end{pmatrix}$$




Agent 1

⋮



Agent i

⋮



Agent N



- Communications are modeled through a shared timer, τ , that has the following dynamics:

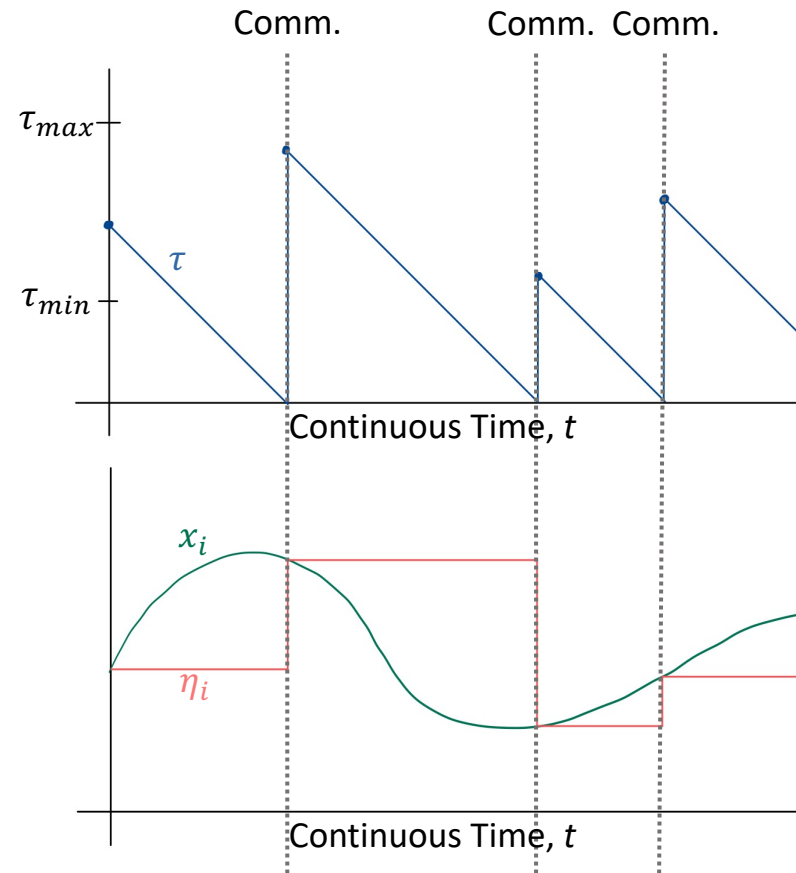
$$\dot{\tau} = -1, \quad \tau \in [0, \tau_{max}],$$

$$\tau^+ \in [\tau_{min}, \tau_{max}], \quad \tau = 0,$$

where τ_{min} and τ_{max} are some positive real numbers.

- When τ reaches zero, all agents update $\eta \in \mathbb{R}^n$ with their current state values, x_i (for agent i), and τ is reset to a value within the range $[\tau_{min}, \tau_{max}]$.
- Agents then use η in their continuous state updates.

“Sample and Hold” Communications



x values are assigned to η when $\tau = 0$



- Agent i has three variables on board, (x_i, η, τ) , which we define as state ξ_i .
- Between communication events, agent i updates both x_i and τ while η does not change. Thus, ξ_i 's continuous-time dynamics may be written as

$$\dot{\xi}_i = \begin{bmatrix} \dot{x}_i \\ \dot{\eta} \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} -\nabla_i L(\eta) \\ 0 \\ -1 \end{bmatrix}, \quad \xi_i \in \mathbb{R}^{n_i} \times \mathbb{R}^n \times [0, \tau_{max}].$$

- Communication events are triggered when $\tau = 0$: x_i stays the same, η is updated with values from all agents, and τ is reset. This is formally modeled as

$$\xi_i^+ = \begin{bmatrix} x_i^+ \\ \eta^+ \\ \tau^+ \end{bmatrix} \in \begin{bmatrix} x_i \\ x \\ [\tau_{min}, \tau_{max}] \end{bmatrix}, \quad \xi_i \in \mathbb{R}^{n_i} \times \mathbb{R}^n \times \{0\}.$$



Definition of a Hybrid System

A hybrid system \mathcal{H} has data (C, f, D, G) that takes the general form

$$\mathcal{H} = \begin{cases} \dot{x} = f(x), & x \in C \\ x^+ \in G, & x \in D \end{cases}$$

where the vector x is the system's state.

f defines the **flow map** and continuous-time dynamics for which C is the **flow set**. G is the set-valued **jump map** which captures the system's discrete behavior for the **jump set** D .

- Our subsystem definition meets the requirements of a hybrid system:

$$\dot{\xi}_i = \begin{bmatrix} \dot{x}_i \\ \dot{\eta} \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} -\nabla_i L(\eta) \\ 0 \\ -1 \end{bmatrix}, \quad \xi_i \in \mathbb{R}^{n_i} \times \mathbb{R}^n \times [0, \tau_{max}]. \quad \xi_i^+ = \begin{bmatrix} x_i^+ \\ \eta^+ \\ \tau^+ \end{bmatrix} \in \begin{bmatrix} x_i \\ x \\ [\tau_{min}, \tau_{max}] \end{bmatrix}, \quad \xi_i \in \mathbb{R}^{n_i} \times \mathbb{R}^n \times \{0\}.$$

Flow map
Flow set
Jump map
Jump set

- However, we want to create a single, combined hybrid system that captures the states of all agents for analysis.



- Towards a combined hybrid system, we define two new variables:

$$z_1 = \text{col}(x_1, \dots, x_N), \quad z_2 = \eta.$$

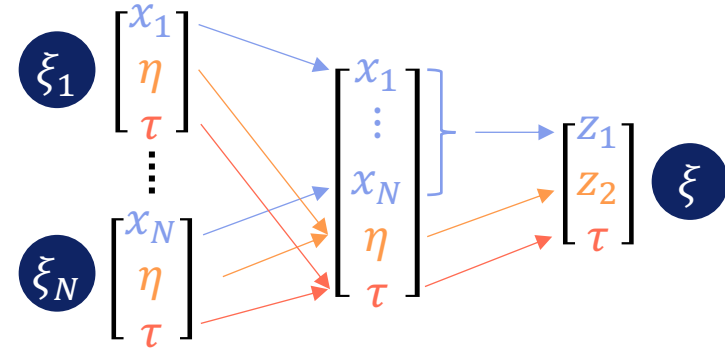
- We then define the state of the combined hybrid system as

$$\xi = (z_1, z_2, \tau).$$

- This leads to the hybrid system $\mathcal{H} = (C, f, D, G)$ given by

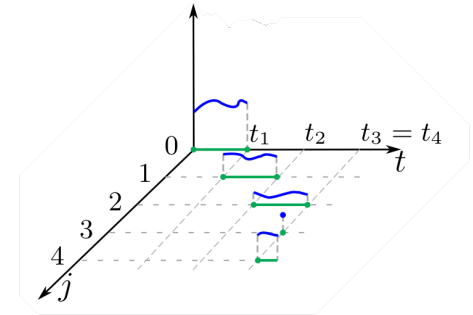
$$\dot{\xi} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} -\nabla L(z_2) \\ 0 \\ -1 \end{bmatrix} = f(\xi), \quad \xi \in C := \mathbb{R}^n \times \mathbb{R}^n \times [0, \tau_{max}]$$

$$\xi^+ = \begin{bmatrix} z_1^+ \\ z_2^+ \\ \tau^+ \end{bmatrix} \in \begin{bmatrix} z_1 \\ z_1 \\ [\tau_{min}, \tau_{max}] \end{bmatrix} = G, \quad \xi \in D := \mathbb{R}^n \times \mathbb{R}^n \times \{0\}$$





- Solutions to \mathcal{H} are denoted by $\phi = (\phi_{z_1}, \phi_{z_2}, \phi_\tau)$, which we parameterize by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$ where t denotes the ordinary (continuous) time and j denotes the jump (discrete) time.



- Under Assumption 1* and an upper bound on τ_{max} , a nontrivial solution exists from every initial point in $C \cup D$. Additionally, every maximal solution ϕ to the hybrid system \mathcal{H} is complete and not Zeno.
 - Takeaway: there are no theoretical obstructions to running this algorithm for arbitrarily long periods of time.
- We say that ϕ has converged when it reaches the set \mathcal{A} defined as

$$\mathcal{A} := \{\xi = (z_1, z_2, \tau) \in C : \nabla L(z_2) = \mathbf{0} \in \mathbb{R}^n, z_2 = z_1, \tau \in [0, \tau_{max}]\} \\ = \{x^*\} \times \{x^*\} \times [0, \tau_{max}],$$

where x^* is the unique fixed point of ∇L and thus, the unique minimizer of L .

- For some vector v , we define $|v|_{\mathcal{A}}$ as the distance between the vector v and the set \mathcal{A} .

* L is twice continuously differential, β -strongly convex, and K -smooth.



Lyapunov Function

$$V(\xi) = (L(z_1) - L(x^*))^2 + (L(z_2) - L(x^*))^2$$

where $\xi = (z_1, z_2, \tau) \in \mathcal{X}$, L is the objective function, and x^* is the unique fixed point of ∇L .

- Central to our analysis is choosing a Lyapunov function that is bounded above and below by K_∞ comparison functions.

Lemma 5: Comparison Functions

There exist $\alpha_1, \alpha_2 \in K_\infty$ such that $\alpha_1(|\xi|_{\mathcal{A}}) \leq V(\xi) \leq \alpha_2(|\xi|_{\mathcal{A}})$ for all $\xi \in C \cup D \cup G(D)$.

In particular, for all $s \geq 0$, α_1 and α_2 are given by

$$\alpha_1(s) = \frac{\beta^2}{16} s^4 \text{ and } \alpha_2(s) = \frac{K^2}{2} s^4,$$

where β is the strong convexity constant of L and K is the Lipschitz constant of ∇L .

- Both comparison functions are used as both $V(\xi)$ and $-V(\xi)$ are upper bounded or may be written as a function of the distance from \mathcal{A} .

K. Hendrickson, D. Hustig-Schultz, M. Hale, & R.G. Sanfelice. (2021). Asynchronous Distributed Gradient Descent with Exponential Convergence Rate via Hybrid Methods, Under Review. arXiv preprint: <https://arxiv.org/abs/2104.10113>



Proposition 2:

Let Assumption 1 hold and consider the hybrid system \mathcal{H} . Choose τ_{min} and τ_{max} such that $0 < \tau_{min} \leq \tau_{max} < \frac{\beta^2}{3K^3}$, where β is the strong convexity constant of L and K is the Lipschitz constant of ∇L . For each solution ϕ such that $\phi_{z_1}(0,0) = \phi_{z_2}(0,0)$, for all $(t, j) \in \text{dom } \phi$, the following is satisfied

$$|\phi(t, j)|_{\mathcal{A}} \leq \sqrt{\frac{K}{\beta}} \sqrt[4]{8} \exp\left(-\frac{\beta AB}{8K^2} t\right) |\phi(0,0)|_{\mathcal{A}}$$

where $A = \beta^2(1 - 2\tau_{max}K) - \tau_{max}K^3 > 0$ and $B = (1 - 2\tau_{max}K) \in (0,1)$.

- $|\phi(t, j)|_{\mathcal{A}} \leq \text{constant} * \exp(-\text{constant} * t) |\phi(0,0)|_{\mathcal{A}}$ for all $(t, j) \in \text{dom } \phi$.
- When agents agree on their initialization value, i.e., when $\phi_{z_1}(0,0) = \phi_{z_2}(0,0)$, exponential stability holds for all time (t, j) .
- While this initialization condition holds in some contexts, some situations preclude such agreement at initialization.

K. Hendrickson, D. Hustig-Schultz, M. Hale, & R.G. Sanfelice. (2021). Asynchronous Distributed Gradient Descent with Exponential Convergence Rate via Hybrid Methods, Under Review. arXiv preprint: <https://arxiv.org/abs/2104.10113>

Theorem 1: Global Exponential Stability

Let Assumption 1 hold and consider the hybrid system \mathcal{H} . Choose τ_{min} and τ_{max} such that $0 < \tau_{min} \leq \tau_{max} < \frac{\beta^2}{3K^3}$, where β is the strong convexity constant of L and K is the Lipschitz constant of ∇L . For each solution ϕ and for all $(t, j) \in \text{dom } \phi$ such that $j \geq 1$, the following is satisfied

$$|\phi(t, j)|_{\mathcal{A}} \leq \frac{8}{3} \sqrt{\frac{K}{\beta}} \sqrt[4]{2} \exp\left(-\frac{\beta AB}{8K^2} t\right) |\phi(0, 0)|_{\mathcal{A}}$$

where $A = \beta^2(1 - 2\tau_{max}K) - \tau_{max}K^3 > 0$ and $B = (1 - 2\tau_{max}K) \in (0, 1)$.

- $|\phi(t, j)|_{\mathcal{A}} \leq \text{constant} * \exp(-\text{constant} * t) |\phi(0, 0)|_{\mathcal{A}}$ for all $j \geq 1$.
- Exponential stability holds for all solutions, regardless of initialization, after the first jump.

K. Hendrickson, D. Hustig-Schultz, M. Hale, & R.G. Sanfelice. (2021). Asynchronous Distributed Gradient Descent with Exponential Convergence Rate via Hybrid Methods, Under Review. arXiv preprint: <https://arxiv.org/abs/2104.10113>

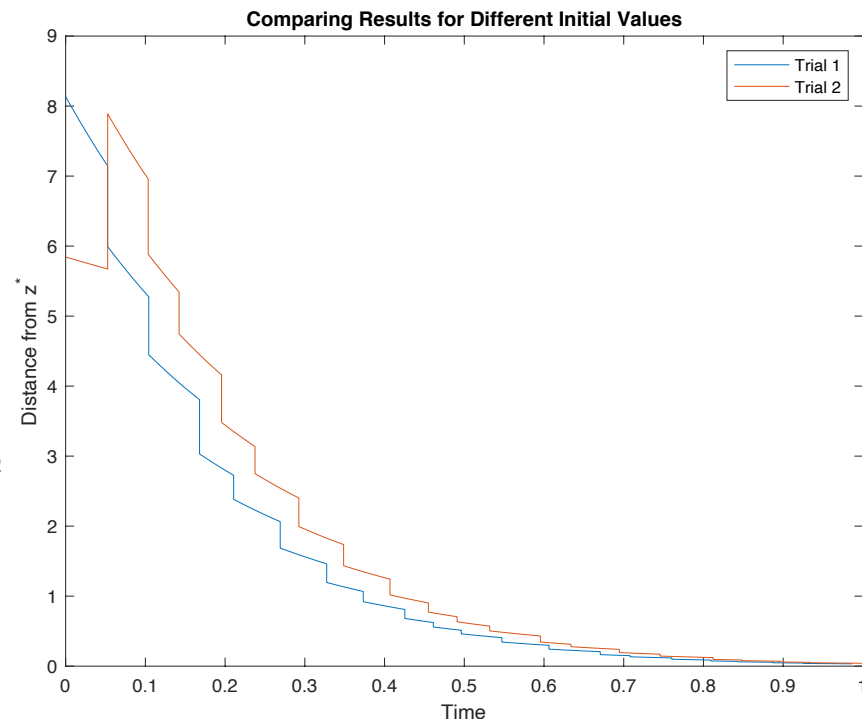


Simulation

$$\text{minimize } L(x) = \frac{1}{2}x^T Qx + b^T x$$

across $N=n$ agents where Q is a $n \times n$ symmetric, positive definite matrix and b is in \mathbb{R}^n .

- We first compared convergence for different initial values of ϕ_{z_1} and ϕ_{z_2} when using five agents.
- When agents agree on initial conditions (Trial 1), there was a consistent decrease in the distance to the minimizer, even at jumps.
- When agents disagree on initial conditions (Trial 2), it's possible that the distance from the minimizer will increase during the first jump. The system then behaves in an exponentially decreasing manner, with the difference between the two trial results decreasing over time.



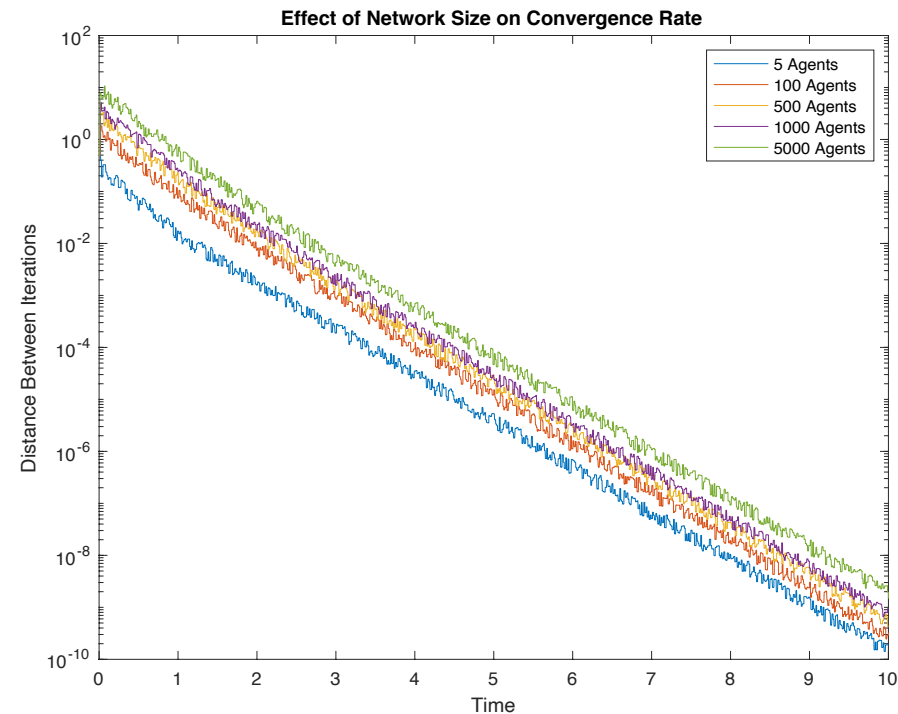


Simulation

$$\text{minimize } L(x) = \frac{1}{2}x^T Qx + b^T x$$

across $N=n$ agents where Q is a $n \times n$ symmetric, positive definite matrix and b is in \mathbb{R}^n .

- We varied the network size from 5 agents to 100, 500, 1,000, and 5,000 agents.
- We chose to initialize ϕ_{z_1} and ϕ_{z_2} with the same values for all the trials.
- As shown in the figure, even drastically changing the network size did not significantly impact the convergence.





- Summary

- We're interested in solving convex optimization problems in a distributed way for which a hybrid systems framework is intuitive and beneficial.
- We distribute a gradient descent update law among agents with communications governed by a shared timer.
- We define the hybrid subsystems as well as a combined hybrid system for analysis.
- We use Lyapunov stability analysis to show global exponential stability for our hybrid system.
- Simulation results confirm our analysis and the scalability of our model.

- Heterogeneous timers

- Each agent will use a separate timer for communication events.
- When an agent's timer reaches zero, they will retrieve updates from all other agents.
- Thus, agents will then have a different versions of η that they use in updates.

$$\dot{\xi}_i = \begin{bmatrix} \dot{x}_i \\ \dot{\eta} \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} -\nabla_i L(\eta) \\ 0 \\ -1 \end{bmatrix} \quad \rightarrow \quad \dot{\zeta}_i = \begin{bmatrix} \dot{x}_i \\ \dot{\eta}_i \\ \dot{\tau}_i \end{bmatrix} = \begin{bmatrix} -\nabla_i L(\eta_i) \\ 0 \\ -1 \end{bmatrix}$$

- Set constraints on x

- This adds the requirement that x be in some set $X \in \mathbb{R}^n$.
- This will lead to different dynamics and a need to exclude certain pathological hybrid phenomena.

Thank you



Kat Hendrickson

kat.hendrickson@ufl.edu

matthewhale@ufl.edu



Matthew Hale



Dawn Hustig-Schultz

dhustigs@ucsc.edu

ricardo@ucsc.edu



Ricardo Sanfelice

UF | UNIVERSITY of FLORIDA

UC SANTA CRUZ

