Updates on Research and Collaborations

Matthew Hale
Department of Mechanical and Aerospace Engineering
University of Florida

Center of Excellence for Assured Autonomy in Contested Environments
Fall 2023 Review
December 7th, 2023

UF | Department of Mechanical & Aerospace Engineering
UNIVERSITY of FLORIDA

UF | UNIVERSITY of FLORIDA
Duke UNIVERSITY
TEXAS The University of Texas at Austin
UC SANTA CRUZ

Updates on Privacy

- Privacy for symbolic systems has appeared
  - Bo Chen, Kevin Leahy, Austin Jones, Matthew Hale, "Differential privacy for symbolic systems with application to Markov Chains," *Automatica*, Volume 152, 2023, 110908.
- Collaboration with Mustafa Karabag, Cyrus Neary, and Ufuk Topcu (UT-Austin)
  - Two CDC 2023 papers on private RL and private stochastic matrices
  - UAI paper on private multi-agent planning
    - Bo Chen, Calvin Hawkins, Mustafa O. Karabag, Cyrus Neary, Matthew Hale, and Ufuk Topcu, "Differential privacy in cooperative multiagent planning," *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence* (UAI), Vol. 216, 347–357.
- Privacy for networks has been accepted
  - C. Hawkins, B. Chen, K. Yazdani, and M.T. Hale, "Node and edge differential privacy for graph Laplacian spectra: Mechanisms and scaling laws", *IEEE Transactions on Network Science and Engineering*, To appear.
- Our work will appear in the mainstream privacy literature
  - Bo Chen and Matthew Hale, "The Bounded Gaussian Mechanism for Differential Privacy," *Journal of Privacy and Confidentiality,* To appear.

Collaborations with Air Force Colleagues

- In summer 2023:
  - William Warke was at RW with Kevin Brink
    - Working on collaborative paper on localization
  - Alexander Benvenuti was at RW with Brendan Bialy
    - The joint paper “Differentially Private Reward Functions for Multi-Agent Markov Decision Processes” is under review
  - Gabriel Behrendt was at RW with Zach Bell
    - The joint paper “Distributed Asynchronous Discrete-Time Feedback Optimization” is under review
  - Calvin Hawkins was at RY with Ben Robinson
    - Working on collaborative paper on changepoint detection

CORELab → Georgia Tech

UF Herbert Wertheim College of Engineering
UNIVERSITY *of* FLORIDA

Mechanical and Aerospace Engineering

Distributed Asynchronous Discrete-Time Feedback Optimization

Gabriel Behrendt, Matthew Longmire (AFRL), Zachary Bell (AFRL), Matthew Hale

POWERING THE NEW ENGINEER TO TRANSFORM THE FUTURE

My Experience With AFRL

- 4th year PhD Candidate at the University of Florida
  - A Totally Asynchronous Algorithm for Time-Varying Convex Optimization Problems [1]
- Two Internships as an AFRL Summer Scholar
  1. Space Vehicles Directorate, Kirtland Air Force Base (Albuquerque, NM)
     - Local Intelligent Networked Collaborative Satellites (LINCS) Lab, Mentor: Dr. Sean Phillips
     - Autonomous Satellite Rendezvous and Proximity Operations with Time-Constrained Sub-Optimal Model Predictive Control [2]
  2. Munitions Directorate, Eglin Air Force Base (Fort Walton Beach, FL)
     - UF Research & Engineering Education Facility (REEF), Mentor: Dr. Zachary Bell
     - Current work under review at IEEE Transactions on Automatic Control

AFRL SCHOLARS PROGRAM

[1] G. Behrendt and M. Hale, “A totally asynchronous algorithm for tracking solutions to time-varying convex optimization problems,” in Proceedings of the 22nd IFAC World Congress, 2023.

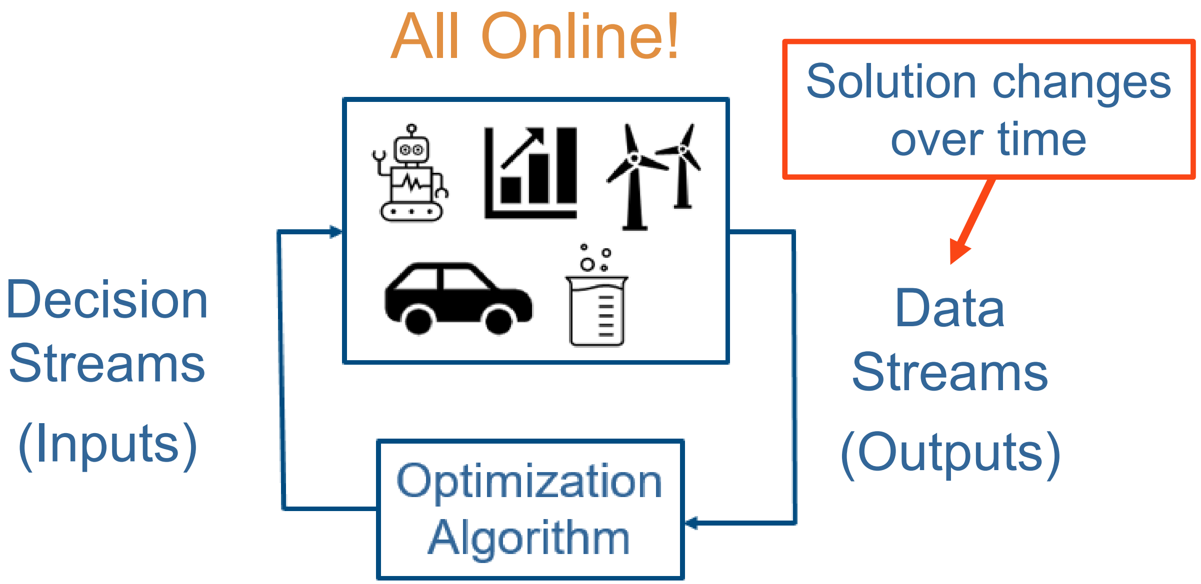
[2] G. Behrendt, A. Soderlund, S. Phillips and M. Hale, “Autonomous Satellite Rendezvous and Proximity Operations with Time-Constrained Sub-Optimal Model Predictive Control” in Proceedings of the 22nd IFAC World Congress, 2023.

Time-Varying Optimization Problems

- Time-varying convex optimization problems arise in…
  - Machine Learning
  - Signal Processing
  - Robotics
  - Power Distribution Systems
- These types of problems can model…
  - Time-varying demands in power distribution systems
  - Real-time control of autonomous vehicles in uncertain or dynamic environments
- Often times, controlling these systems requires feedback from the dynamic system
- We can use feedback to steer the system toward an optimal operating point using the time-varying optimization framework *online*

Feedback Optimization Problems

- Feedback Optimization utilizes feedback from the system to generate timely decisions *online* to steer a system toward an optimal operating point
- Feedback Optimization Process
  1. Optimization algorithm makes some progress toward the solution to generate a timely decision
  2. Decision is applied to the system
  3. Data is collected and fed back to the optimization algorithm
  4. Repeat
- This is *not* Model Predictive Control
- Often times, these types of problems are solved over a network of agents
- These networks are subject to *asynchrony* in their "operations"

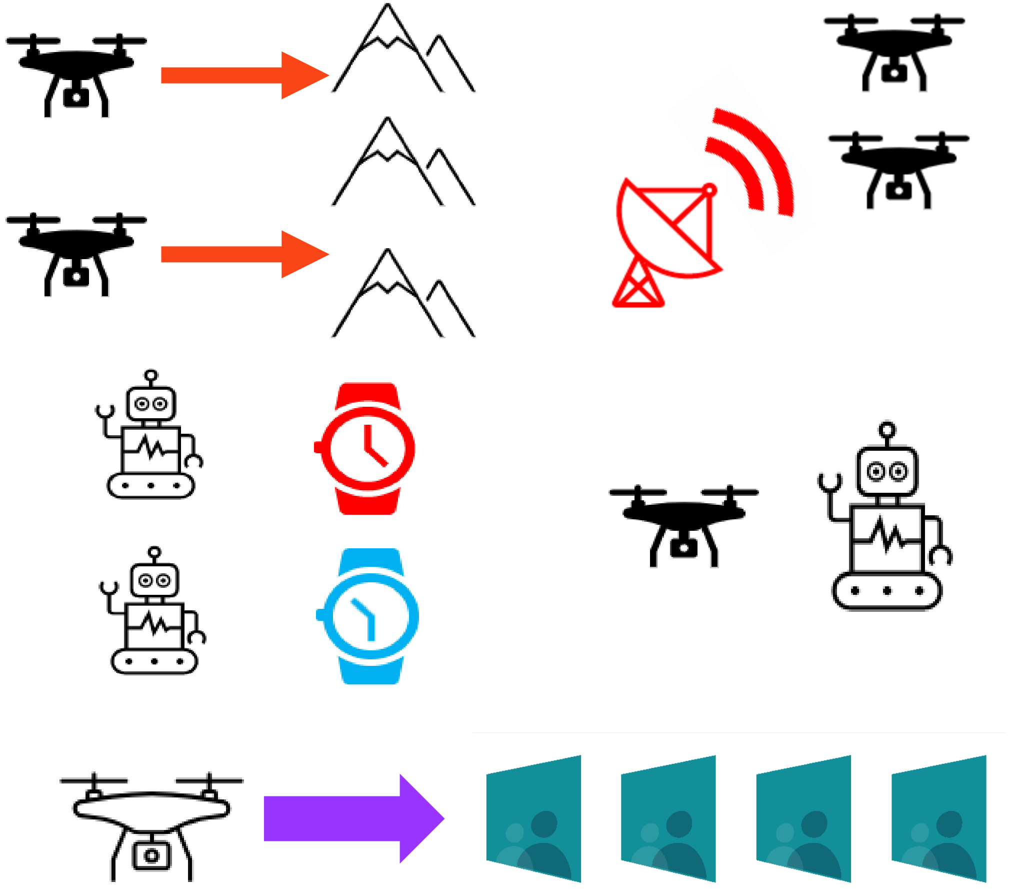


Feedback Optimization Can Account For…

- Unforeseen Disturbances
- Dynamic/Unknown Environments
- New Incoming Data

Multi-Agent Systems are Subject to Asynchrony

- Many multi-agent systems face asynchrony in agents' operations
- "Operations" = communications, computations, output measurements
  - Asynchronous Communications
  - Asynchronous Computations
  - Asynchronous Output Measurements
- As a result, agents may generate, measure, and share information with unpredictable timing



Goal: Develop a multi-agent algorithm to track the solutions of feedback optimization problems where agents' operations are subject to asynchrony

Problem Statement

Problem 1. Given $f: \mathbb{R}^n \times \mathbb{N}_0 \to \mathbb{R}$, $g: \mathbb{R}^m \times \mathbb{N}_0 \to \mathbb{R}$, over a network of N agents, indexed over the set $[N] := \{1, \dots, N\}$, asynchronously track the solution of

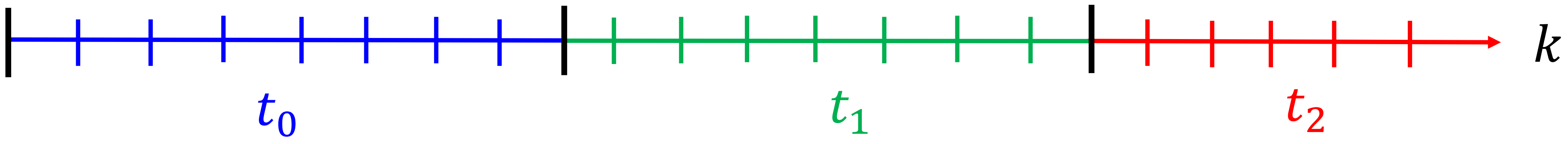
$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad J(x, y; t_\ell) := f(x; t_\ell) + g(y; t_\ell)$$
$$\text{subject to} \quad y = Cx,$$

over a time horizon $\mathcal{T} = \{0, \dots, T\}$ where $C \in \mathbb{R}^{m \times n}$.

$$\underset{x \in \mathcal{X}}{\text{minimize}} \; J(x, y; t_0)$$
$$x^*(t_0) := \underset{x \in \mathcal{X}}{\arg\min} \, f(x; t_0) + g(Cx; t_0)$$
$$y^*(t_0) := Cx^*(t_0)$$

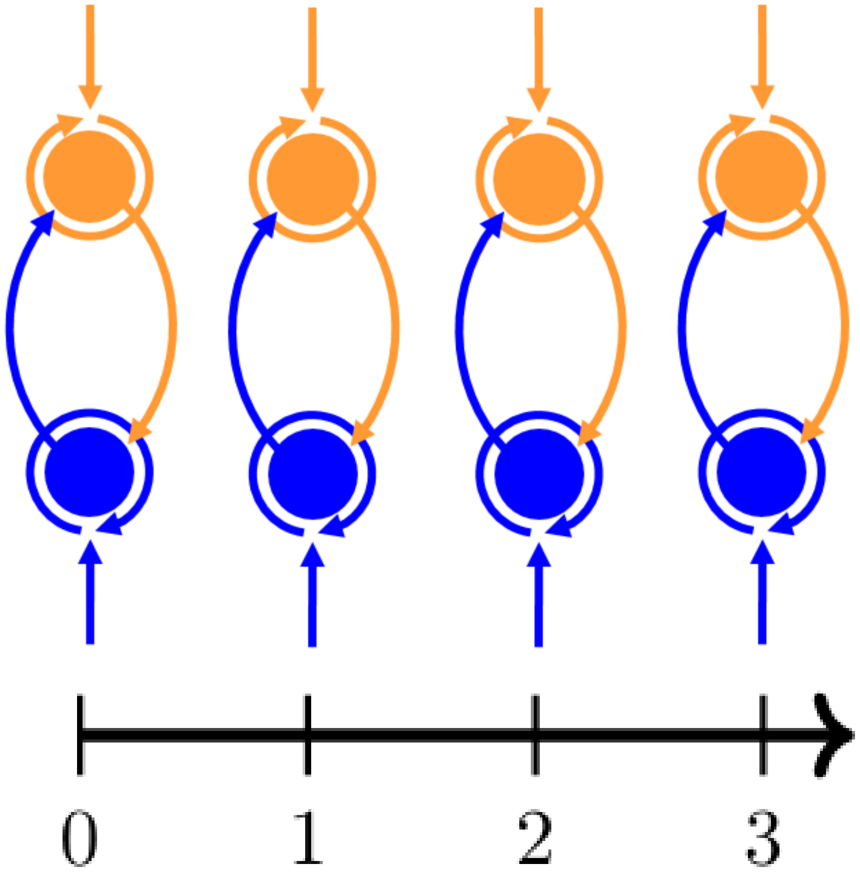
$$\underset{x \in \mathcal{X}}{\text{minimize}} \; J(x, y; t_1)$$
$$x^*(t_1) := \underset{x \in \mathcal{X}}{\arg\min} \, f(x; t_1) + g(Cx; t_1)$$
$$y^*(t_1) := Cx^*(t_1)$$

$$\underset{x \in \mathcal{X}}{\text{minimize}} \; J(x, y; t_2)$$
$$x^*(t_2) := \underset{x \in \mathcal{X}}{\arg\min} \, f(x; t_2) + g(Cx; t_2)$$
$$y^*(t_2) := Cx^*(t_2)$$



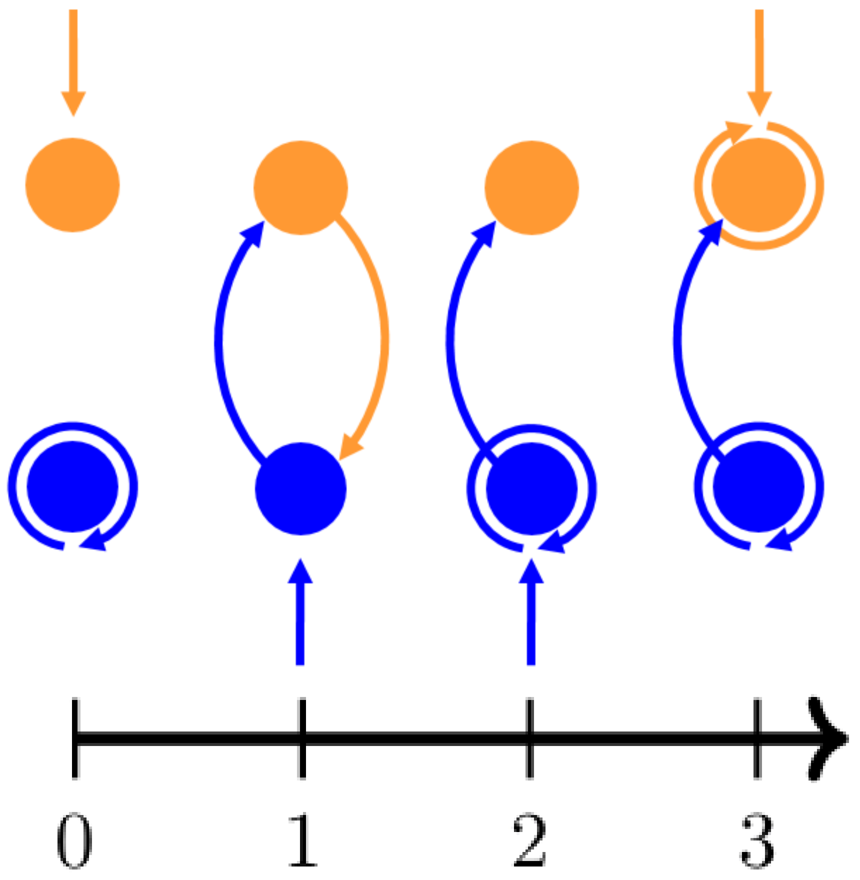
Asynchrony in Networks of Agents

- Ideally want *Synchronous* setting
  - Agents perform computations and communications at the same time and at every time step
- Not always the case
  - Agents may compute and measure at different rates
  - Communications may be intermittent, delayed, or lost
- Can lead to a *straggler penalty*
  - If the network has to wait for the slowest agent, then the network is only as fast as its slowest agent
- Has led to interest in *asynchronous settings*

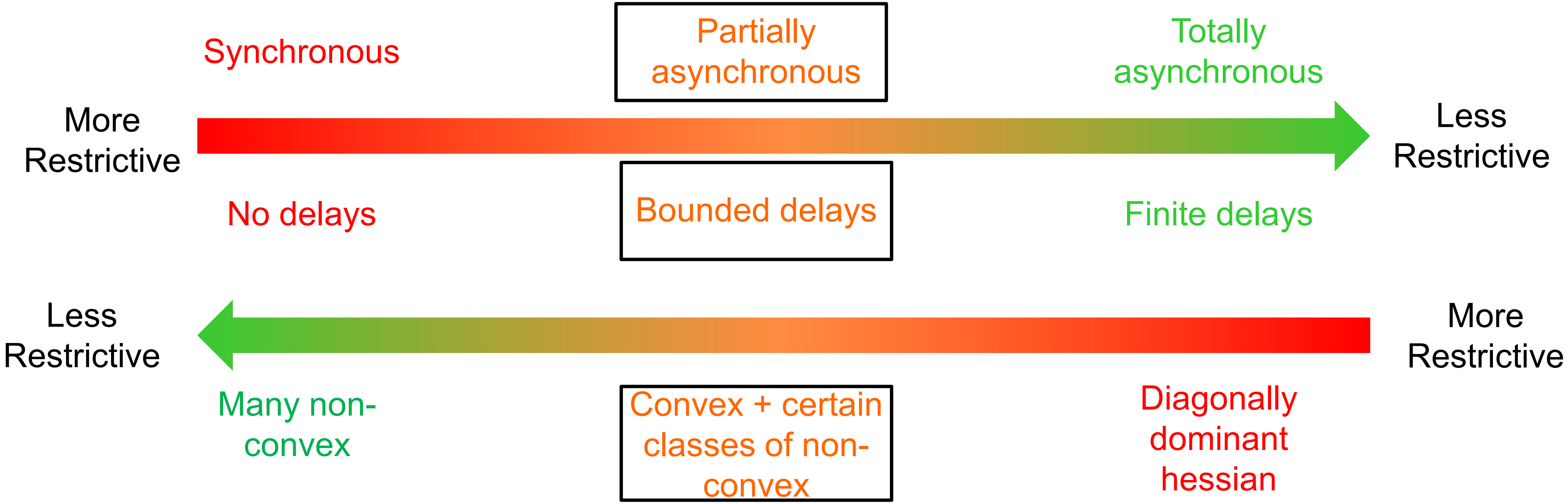


Asynchronous Settings

- Two types *asynchronous* settings
  - *Partially Asynchronous*
  - *Totally Asynchronous*
- *Totally Asynchronous* settings have no bound on operation delays (but delays are finite)
- *Partially Asynchronous* settings require bounded delays in agents' operations
- Advantages
  - Robust to intermittent operations
  - Networks in asynchronous settings do not suffer from the straggler penalty



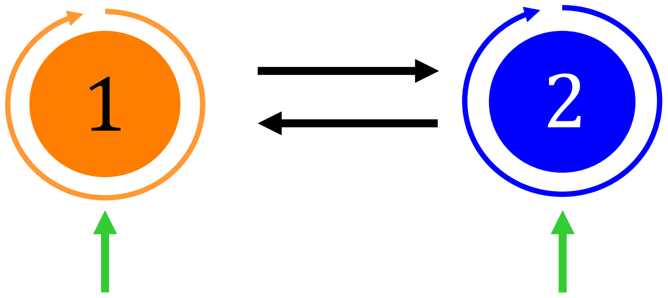
Asynchronous Settings Summarized

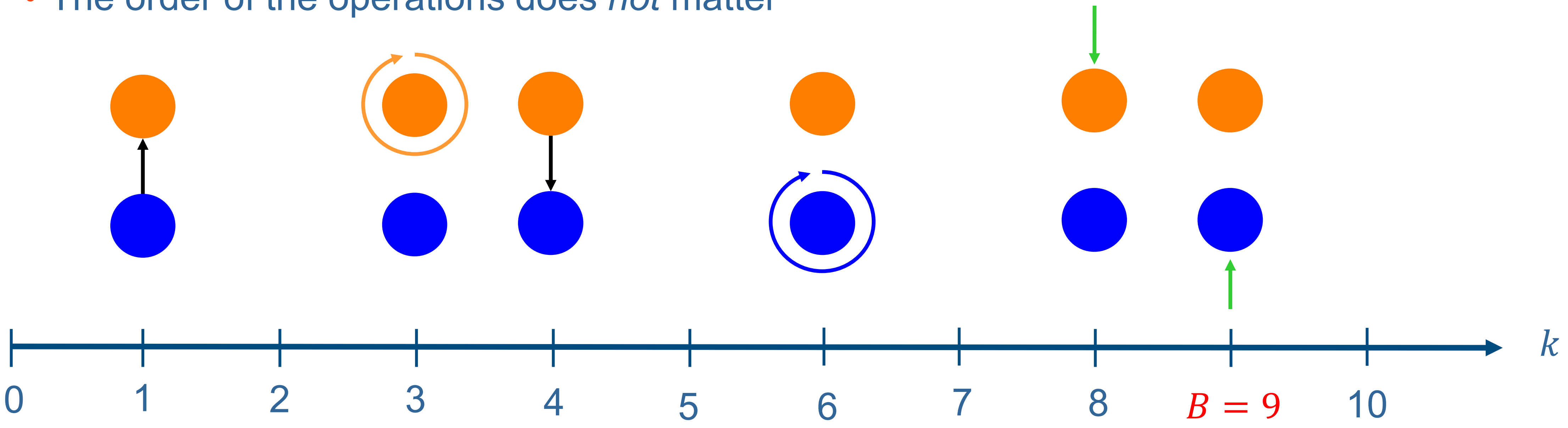


- We would like to propose an algorithm that is robust to asynchrony while considering wide class of problems
- Therefore, we propose to use an algorithm in the *Partially Asynchronous* setting

Example of Bounded Delays

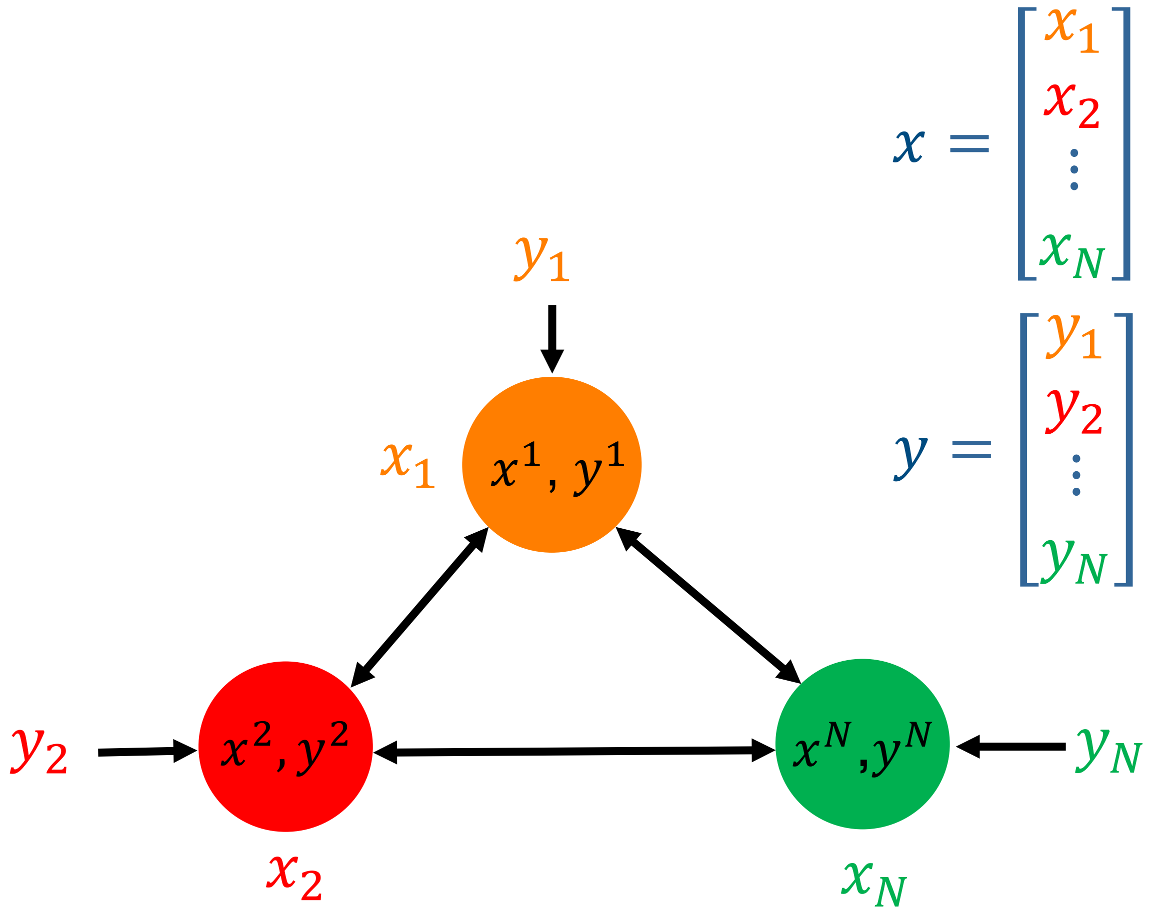
- Operations Delay is upper bounded by B
  - Computations – 3,6
  - Communications – 1,4
  - Measurements – 8,9
- The order of the operations does *not* matter





Partially Asynchronous Block Coordinate Descent

- Block Coordinate Descent
  - Each agent i has a local copy of the network's decision vector x^i and output vector y^i
  - The decision vector x and output vector y are partitioned into blocks and assigned to each agent
  - Agent i updates its block of the decision vector x_i and measures its outputs y_i
  - Agent i communicates its update of x_i and measurements of y_i to other agents in the network



Formal Algorithm Statement

- Algorithm 1
  - Update Law

$$x_i^i(k+1) = \begin{cases} \Pi_{X_i}\left[x_i^i(k) - \gamma_\ell\left(\nabla_{x_i} f\left(x^i(k); t_\ell\right) + C_i^T \nabla_y g\left(y^i(k); t_\ell\right)\right)\right] & \text{agent } i \text{ updates at time } k \\ x_i^i(k) & \text{otherwise} \end{cases}$$

$$x_j^i(k+1) = \begin{cases} x_j^j\left(\tau_j^i(k)\right) & \text{agent } i \text{ receives } x_j^j \text{ at time } k \\ x_j^i(k) & \text{otherwise} \end{cases}$$

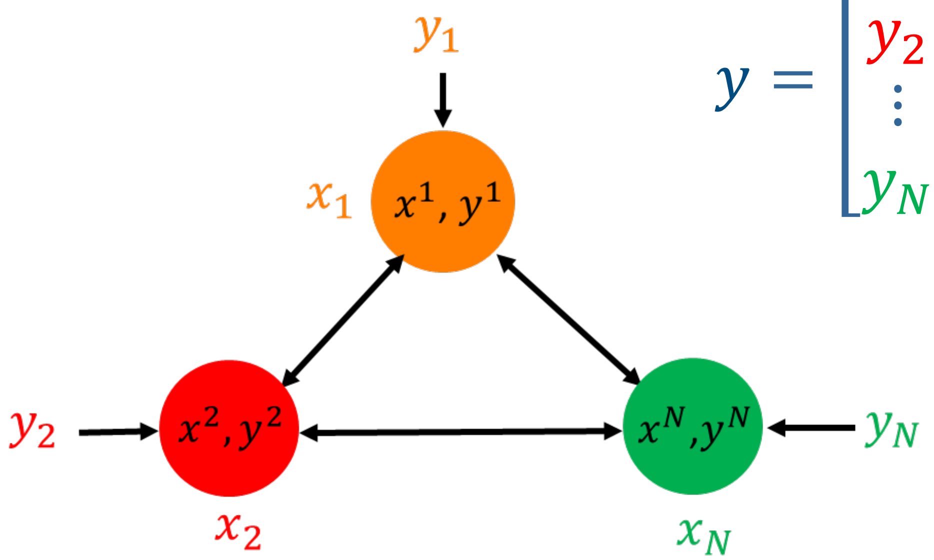
  - Measurement Law

$$y_i^i(k+1) = \begin{cases} y_i(k) & \text{agent } i \text{ measures at time } k \\ y_i^i(k) & \text{otherwise} \end{cases}$$

$$y_j^i(k+1) = \begin{cases} y_j^j\left(\mu_j^i(k)\right) & \text{agent } i \text{ receives } y_j^j \text{ at time } k \\ y_j^i(k) & \text{otherwise} \end{cases}$$

- Due to asynchrony, updates often times are computed with outdated information, but delays are bounded

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$



Time-Invariant Convergence Result

Theorem 1. For a fixed t_0, a step size $\gamma_0 \in (0, \gamma_{0,\max}) < 1$, the sequence $\{x(k), y(k)\}_{k \in \mathbb{N}_0}$ generated by N agents executing Algorithm 1 satisfies

$$\alpha(r_0 B; t_0) \leq a_0 \rho_0^{r_0 - 1}$$
$$\beta(r_0 B) \leq b_0 \rho_0^{r_0 - 1}$$
$$\delta(r_0 B) \leq d_0 \rho_0^{r_0 - 1}$$

where $a_0, b_0, d_0 > 0$, $\rho_0 \in (0,1)$, and

$$\alpha(k; t_0) \coloneqq J(x(k), y(k); t_0) - J(x^*(t_0), y^*(t_0); t_0)$$ ← Sub-optimality Gap

$$\beta(k) \coloneqq \sum_{\tau = k-B}^{k-1} \|s(\tau)\|^2$$ ← Difference of Agent gradient updates

$$\delta(k) \coloneqq \sum_{\tau = k-B}^{k-1} \|q(\tau)\|^2$$ ← Difference of Agent measurements

Takeaway: Our Partially Asynchronous Block Coordinate Descent algorithm converges toward the minimizer linearly for a *time-invariant feedback optimization* problem

Time-Varying Tracking Result

Theorem 2. For a fixed $T \in \mathbb{N}$ and fix $\mathcal{T} = \{t_0, \dots, t_T\}$. For all $t_\ell \in \mathcal{T}$, and a step size a step size $\gamma_\ell \in \left(0, \gamma_{\ell,\max}\right) < 1$, the sequence $\{x(k), y(k)\}_{k \in \mathbb{N}_0}$ generated by N agents executing Algorithm 1 satisfies

$$\alpha(\eta_{\ell-1} + r_\ell B; t_\ell) \leq a_\ell \rho_\ell^{r_\ell - 1}$$

$$\beta(\eta_{\ell-1} + r_\ell B) \leq b_\ell \rho_\ell^{r_\ell - 1}$$

$$\delta(\eta_{\ell-1} + r_\ell B) \leq d_\ell \rho_\ell^{r_\ell - 1}$$

where $a_\ell, b_\ell, d_\ell > 0$, $\rho_\ell \in (0,1)$, and

$$\alpha(k; t_\ell) \coloneqq J(x(k), y(k); t_\ell) - J(x^*(t_\ell), y^*(t_\ell); t_\ell)$$

$$\beta(k) \coloneqq \sum_{\tau = k - B_\ell}^{k-1} \|s(\tau)\|^2$$

$$\delta(k) \coloneqq \sum_{\tau = k - B_\ell}^{k-1} \|q(\tau)\|^2$$

What's the difference?

Time-Varying Tracking Result

Theorem 2. For a fixed $T \in \mathbb{N}$ and fix $\mathcal{T} = \{t_0, \dots, t_T\}$. For all $t_\ell \in \mathcal{T}$, and a step size a step size $\gamma_\ell \in (0, \gamma_{\ell,\max}) < 1$, the sequence $\{x(k), y(k)\}_{k\in\mathbb{N}_0}$ generated by N agents executing Algorithm 1 satisfies

$$\alpha(\eta_{\ell-1} + r_\ell B; t_\ell) \le a_\ell \rho_\ell^{r_\ell - 1}$$
$$\beta(\eta_{\ell-1} + r_\ell B) \le b_\ell \rho_\ell^{r_\ell - 1}$$
$$\delta(\eta_{\ell-1} + r_\ell B) \le d_\ell \rho_\ell^{r_\ell - 1}$$

where $a_\ell, b_\ell, d_\ell > 0$, $\rho_\ell \in (0,1)$, and

$$\alpha(k; t_\ell) \coloneqq J(x(k), y(k); t_\ell) - J(x^*(t_\ell), y^*(t_\ell); t_\ell)$$
$$\beta(k) \coloneqq \sum_{\tau=k-B_\ell}^{k-1} \|s(\tau)\|^2$$
$$\delta(k) \coloneqq \sum_{\tau=k-B_\ell}^{k-1} \|q(\tau)\|^2$$

What's the difference?

$$a_\ell = \underbrace{a_{\ell-1}\rho_{\ell-1}^{r_{\ell-1}-1}} + \underbrace{2\Delta L_t} + \cdots$$

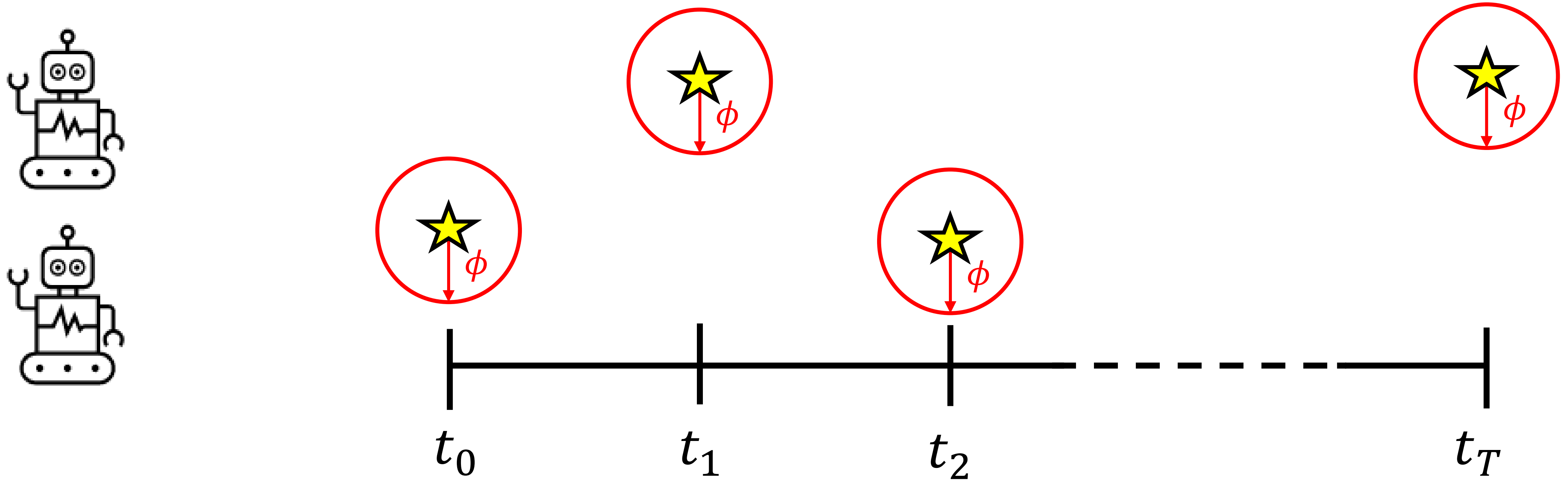
How close the network tracked the minimizer at $t_{\ell-1}$

How far the minimizer has moved from $t_{\ell-1} \to t_\ell$

Takeaway: Our Partially Asynchronous Block Coordinate Descent algorithm converges toward the minimizer linearly for a *time-varying feedback optimization* problem for all $t_\ell \in \mathcal{T}$.

Performance Requirement

- For all $t_\ell \in \mathcal{T}$, suppose agents complete $r_\ell B$ iterations where $r_\ell \equiv r$, track the minimizer $\left(x^*(t_\ell), y^*(t_\ell)\right)$ within a bounded error $\phi > 0$ prior to each objective function change $J(\cdot,\cdot; t_\ell) \to J(\cdot,\cdot; t_{\ell+1})$.

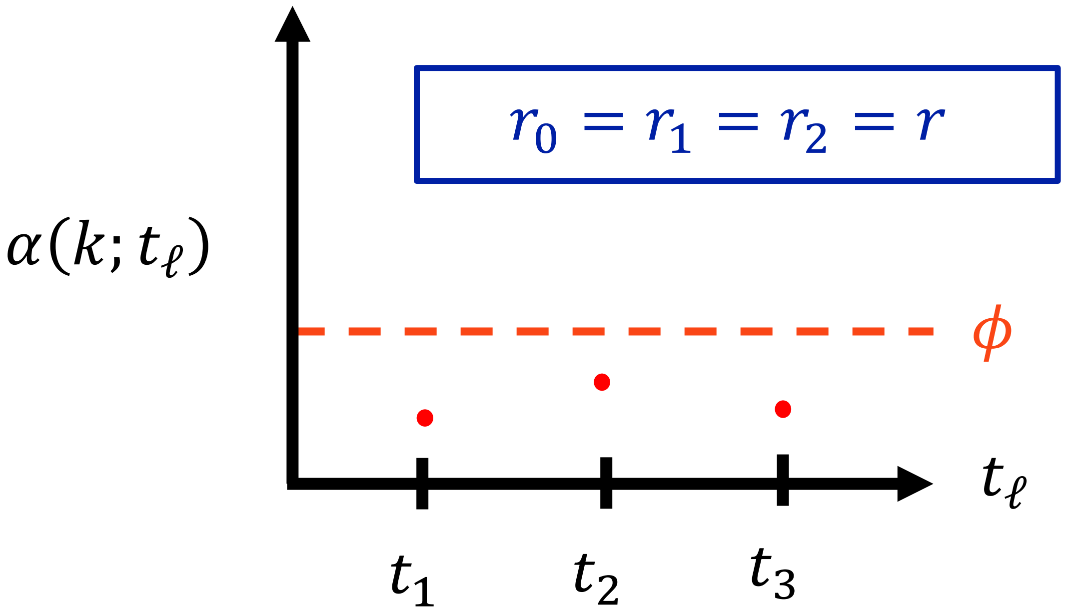


Performance Requirement Result

Theorem 4. Let $\phi > 0$. For a fixed $T \in \mathbb{N}$ and fix $\mathcal{T} = \{t_0, \dots, t_T\}$. Suppose N agents are executing Algorithm 1 with $r_\ell \equiv r$ for all $t_\ell \in \mathcal{T}$ and $r \in \mathbb{N}$ with $r \geq 2$. Let $V_{\max} = \left\{a_0, \max_{t_\ell \in \mathcal{T}} V_\ell\right\} > 0$ and $\rho_{\max} \coloneqq \max_{t_\ell \in \mathcal{T}} \rho_\ell \in (0,1)$. If

$$r \geq 1 + \frac{\ln\left(\frac{V_{\max}\rho_{\max}^{(T+2)(r-1)} + \phi}{V_{\max} + \phi}\right)}{\ln(\rho_{\max})}$$

then $\alpha(\eta_\ell; t_\ell) \leq \phi$ for all $t_\ell \in \mathcal{T}$.



Takeaway: If agents complete enough operations they can track the minimizer within an error bound of $\phi > 0$

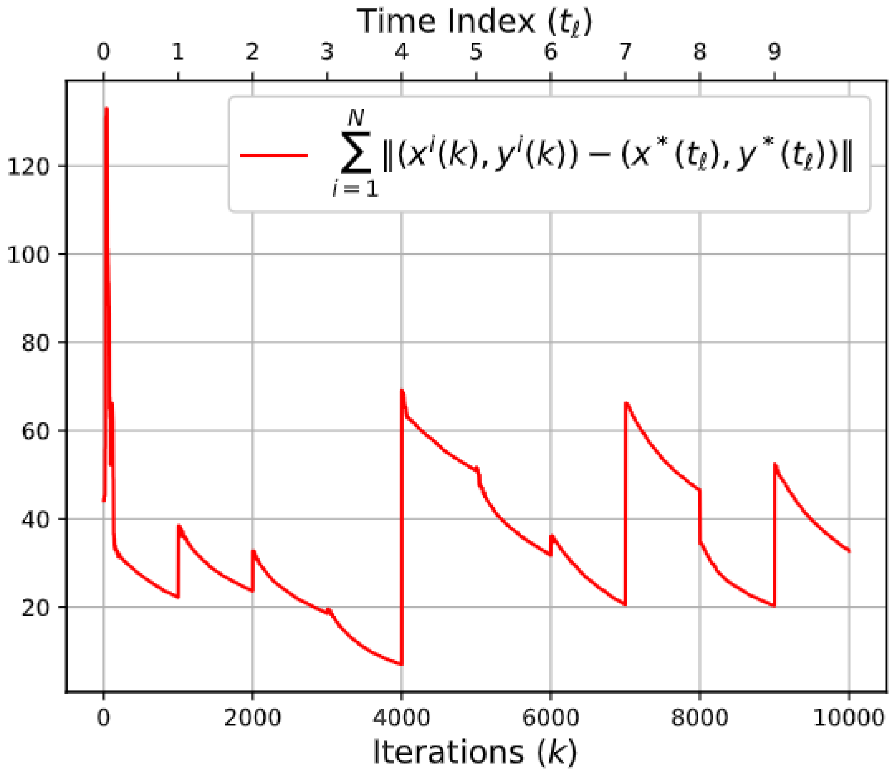
Simulation #1: Time-Varying Quadratic Programs with Feedback

- 10 agents track the solution of a time-varying quadratic program

$$\begin{aligned} \underset{x \in \mathcal{X}}{\text{minimize}} \quad & \frac{1}{2} x^T Q(t_\ell) x + q(t_\ell)^T x + \frac{1}{2} y^T P(t_\ell) y + p(t_\ell)^T y \\ \text{subject to} \quad & y = Cx \end{aligned}$$

- The probability of agent i computing an update, measuring its output, or communicating is 0.01
- Maximum operation delay $B = 5$
- The objective function changes every 1000 iterations
- The sudden increases in error are due to the change of objective function which changes the minimizer from

$$\left(x^*(t_\ell), y^*(t_\ell)\right) \rightarrow \left(x^*(t_{\ell+1}), y^*(t_{\ell+1})\right)$$



Simulation #1: Theorem 2 Values

Sub-optimality Gap

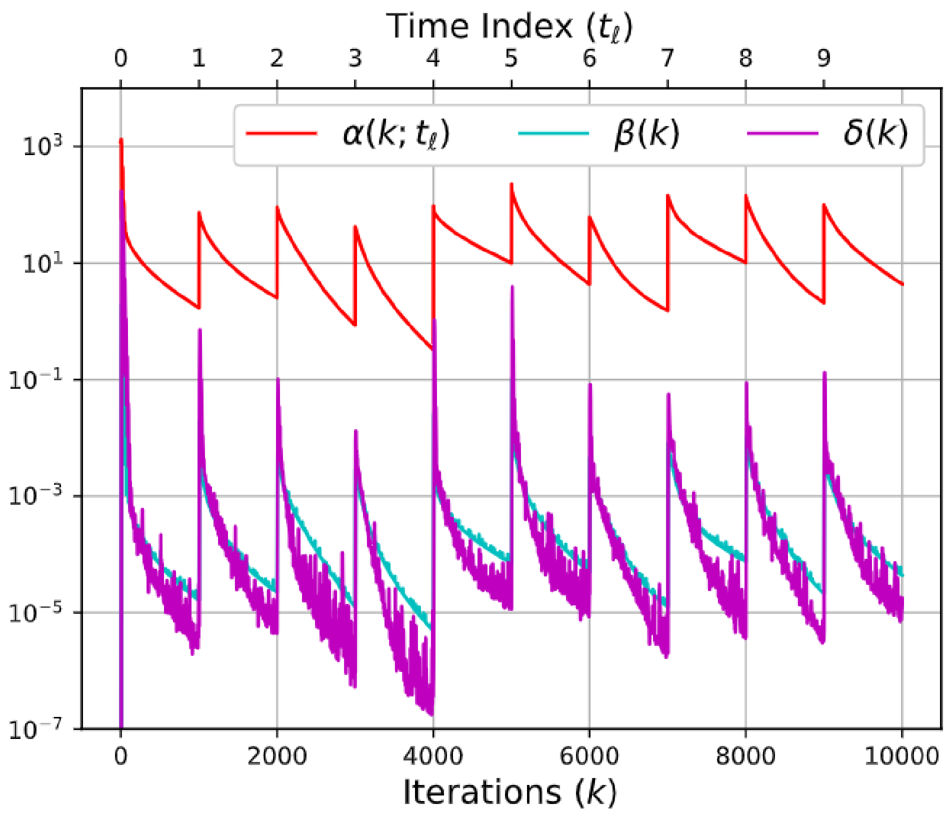
$$\alpha(k;t_\ell) := J(x(k),y(k);t_\ell) - J(x^*(t_\ell),y^*(t_\ell);t_\ell)$$

Difference of Agent gradient updates

$$\beta(k) := \sum_{\tau=k-B_\ell}^{k-1} \|s(\tau)\|^2$$

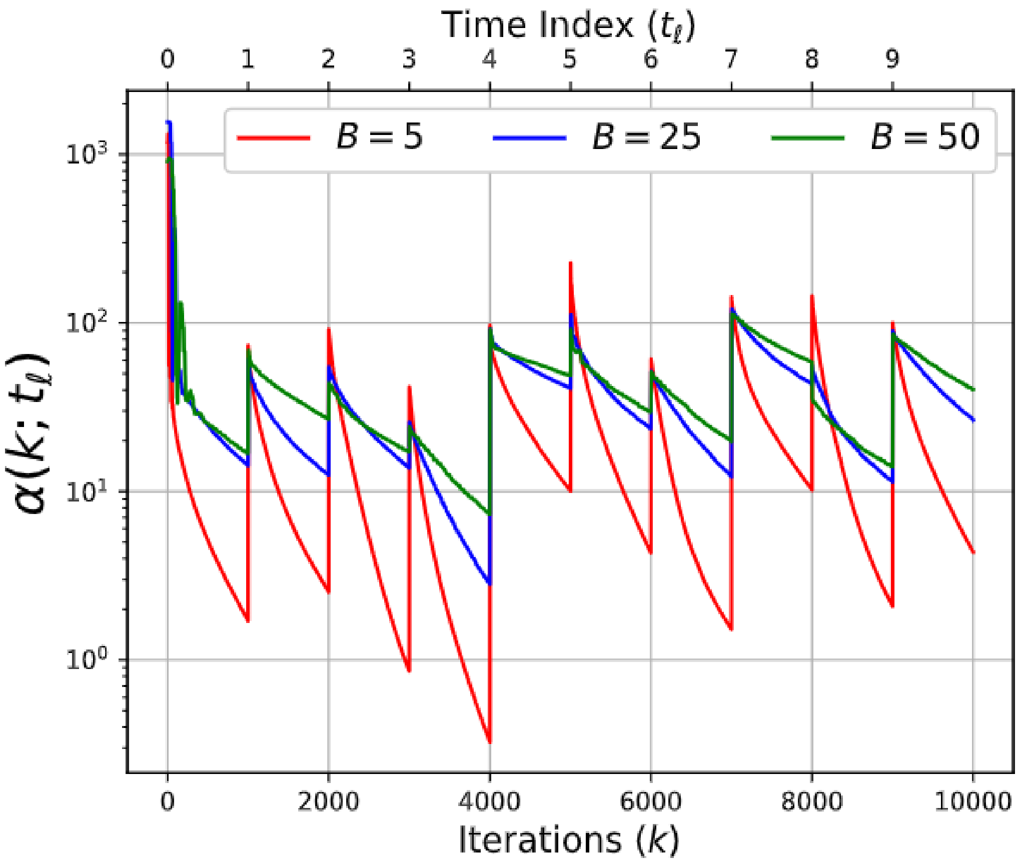
Difference of Agent measurements

$$\delta(k) := \sum_{\tau=k-B_\ell}^{k-1} \|q(\tau)\|^2$$



Effect of Maximum Operations Delay B

$$\alpha(k;t_\ell) \coloneqq J(x(k),y(k);t_\ell) - J(x^*(t_\ell),y^*(t_\ell);t_\ell)$$



Simulation #2: Aircraft Altitude Tracking

- We consider the longitudinal dynamics of 8 F16-XL aircraft
- Agent input and output vectors

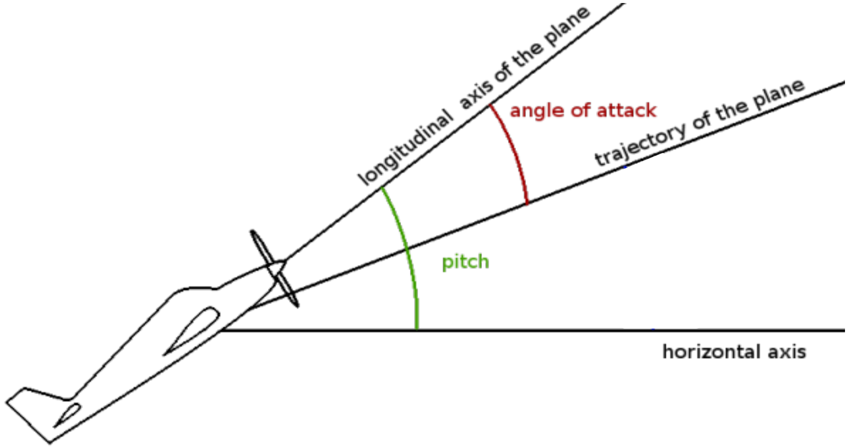
$$x_i = [v_i, \vartheta_i, \varphi_i, \dot{\varphi}_i, \xi_i] \in \mathbb{R}^5$$
$$y_i = [\dot{v}_i, \xi_i] \in \mathbb{R}^2$$

  - $v_i \rightarrow$ velocity
  - $\dot{v}_i \rightarrow$ acceleration
  - $\vartheta_i \rightarrow$ angle of attack
  - $\varphi_i \rightarrow$ pitch angle
  - $\dot{\varphi}_i \rightarrow$ pitch rate
  - $\xi_i \rightarrow$ altitude

- Linearized Input-Output Map

$$y_i = C_i x_i$$

$$C_i = \begin{bmatrix} -0.0133 & -7.53269 & -3.17 & -1.1965 & 0.0001 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Simulation #2: Problem Formulation

- The aircraft are tasked with the following goals
  1. Track a time-varying desired altitude, $\Phi(t_\ell) \in \mathbb{R}$ ★
  2. Track a time-varying desired acceleration, $\Psi_i(t_\ell) \in \mathbb{R}, \forall i = 1, \ldots, 8$ ★
  3. Maintain a desired altitude separation $\omega_i \in \mathbb{R}, \forall i = 1, \ldots, 8$ ★
- Problem Formulation

$$\Theta(t_\ell) = \begin{bmatrix} \Phi(t_\ell) \\ \Psi_1(t_\ell) \\ \vdots \\ \Phi(t_\ell) \\ \Psi_8(t_\ell) \end{bmatrix} \in \mathbb{R}^{16}$$

$$\underset{x \in \mathcal{X}}{\text{minimize}} \ \boxed{\frac{1}{2} x^T Q x} + \boxed{\frac{1}{2}\left(y - \Theta(t_\ell)\right)^T P \left(y - \Theta(t_\ell)\right)} + \boxed{\frac{1}{2}(\tilde{\xi} - \omega)^T P (\tilde{\xi} - \omega)}$$

Input Cost

Altitude + Acceleration Tracking Cost

Altitude Separation Cost

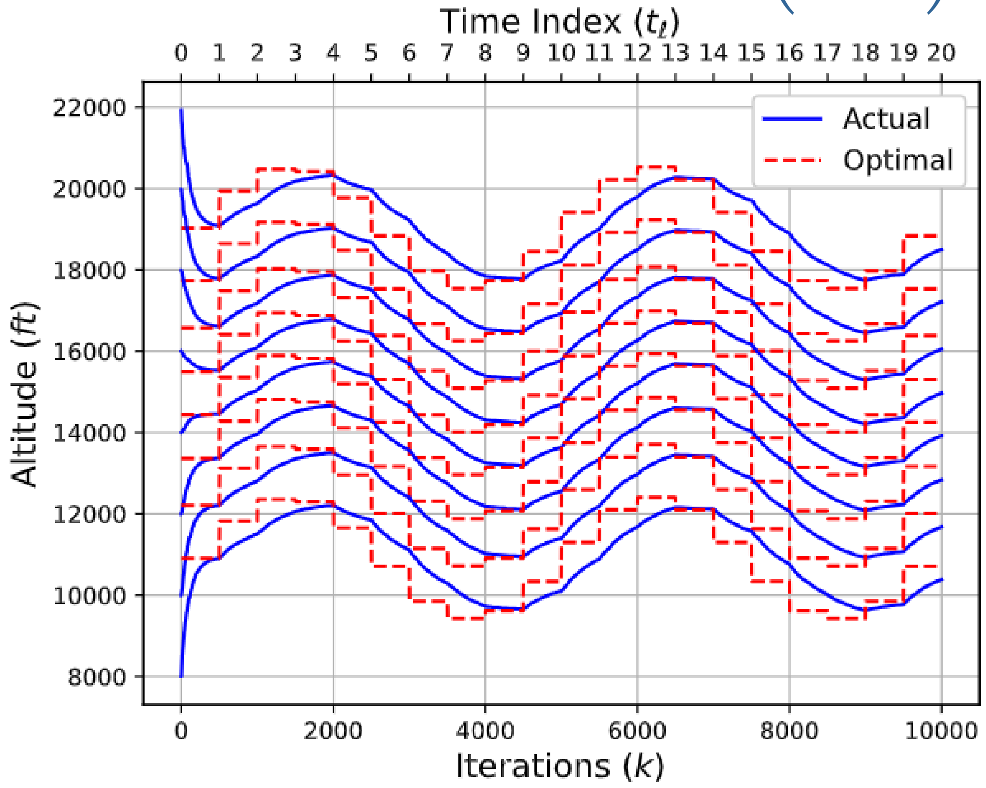
$$\text{subject to } \ y = Cx$$

$$C = \begin{bmatrix} C_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_8 \end{bmatrix}$$

Simulation #2: Tracking Results

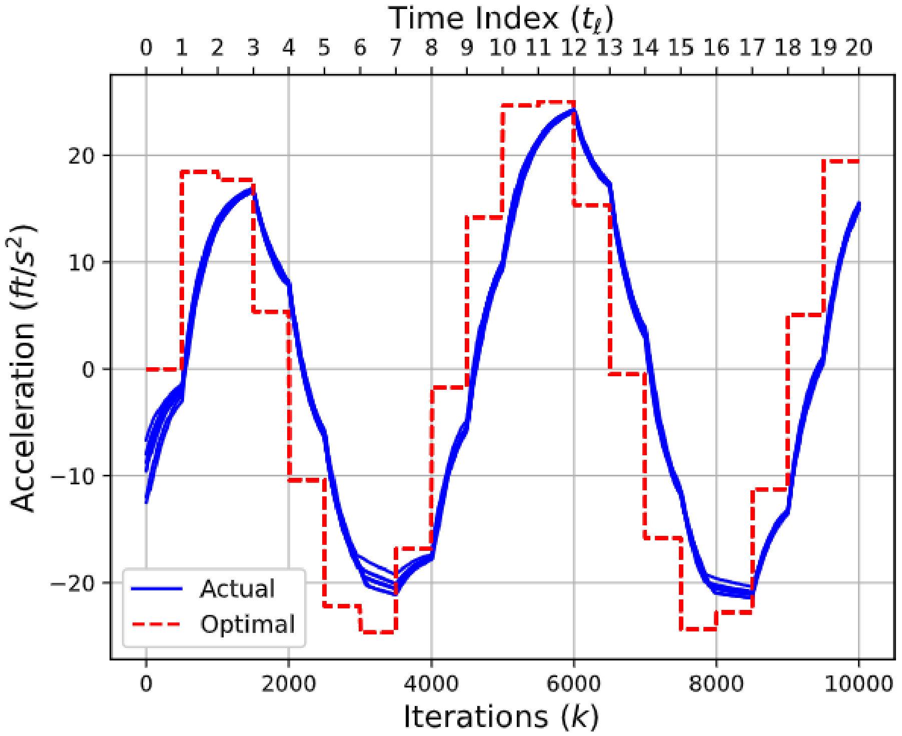
- Altitude Tracking

$$\Phi(t_\ell) = 15{,}000 + 1500 \sin\left(\frac{t_\ell t_s \pi}{24}\right)$$



- Acceleration Tracking

$$\Psi_i(t_\ell) = \frac{0.1}{t_s}\left(\Phi(t_\ell) - \boxed{\frac{1}{N}\sum_{j=1}^{N} \xi_j^i(\eta_\ell)}\right)$$

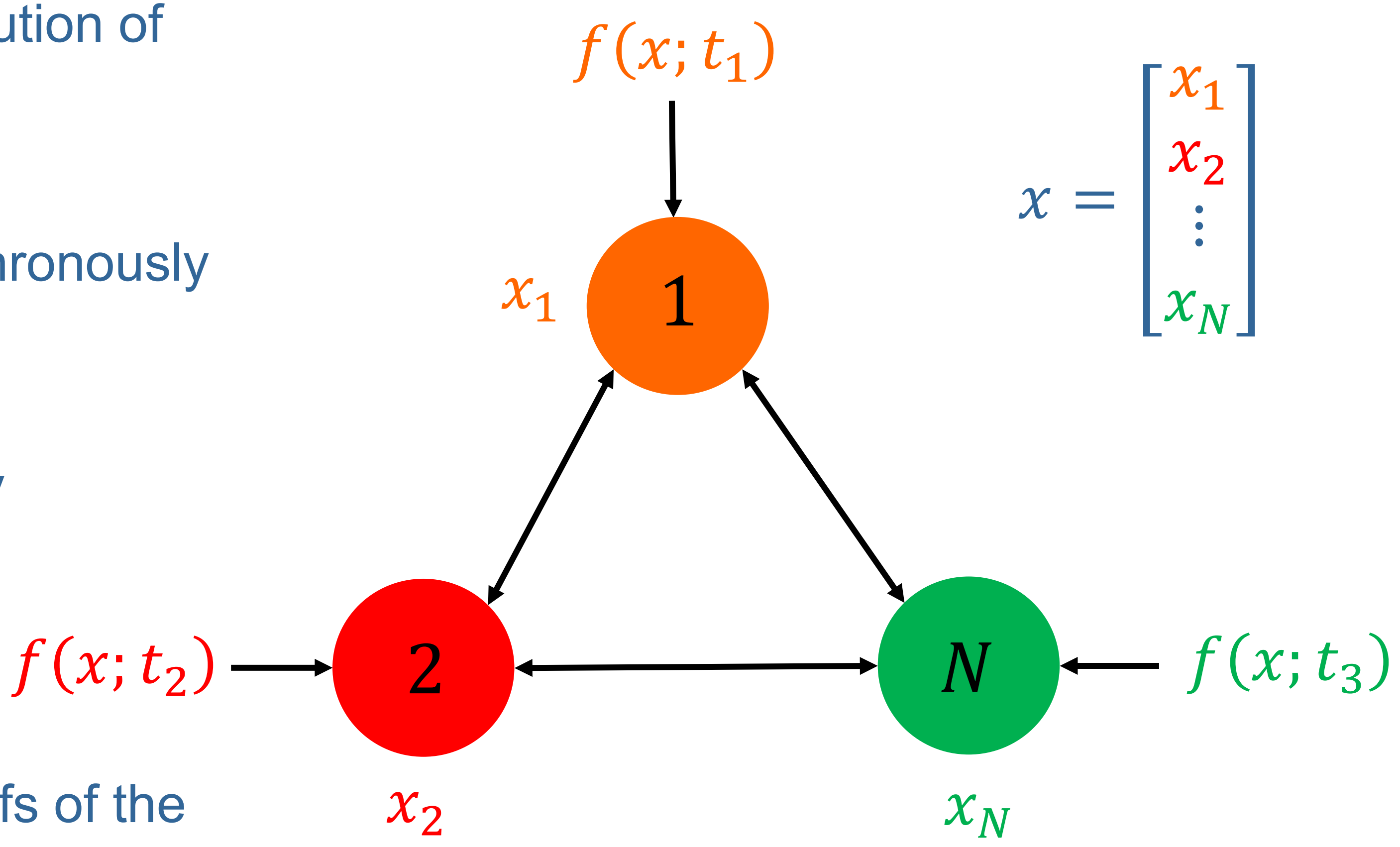


Future Work: Asynchronous Objective Sampling

Over a network of N agents, track the solution of

$$\underset{x \in \mathcal{X}}{\text{minimize}} := f(x; t_\ell)$$

- Agent i computes updates to x_i asynchronously
- Agent i communicates updates to x_i asynchronously
- Agent i samples $f(\cdot; t)$ asynchronously
    - Agent 1 minimizes $f(x; t_1)$
    - Agent 2 minimizes $f(x; t_2)$
    - Agent 3 minimizes $f(x; t_3)$
- Application: Agents have differing beliefs of the target position



Thank you

Simulation #2: Aircraft Altitude Tracking