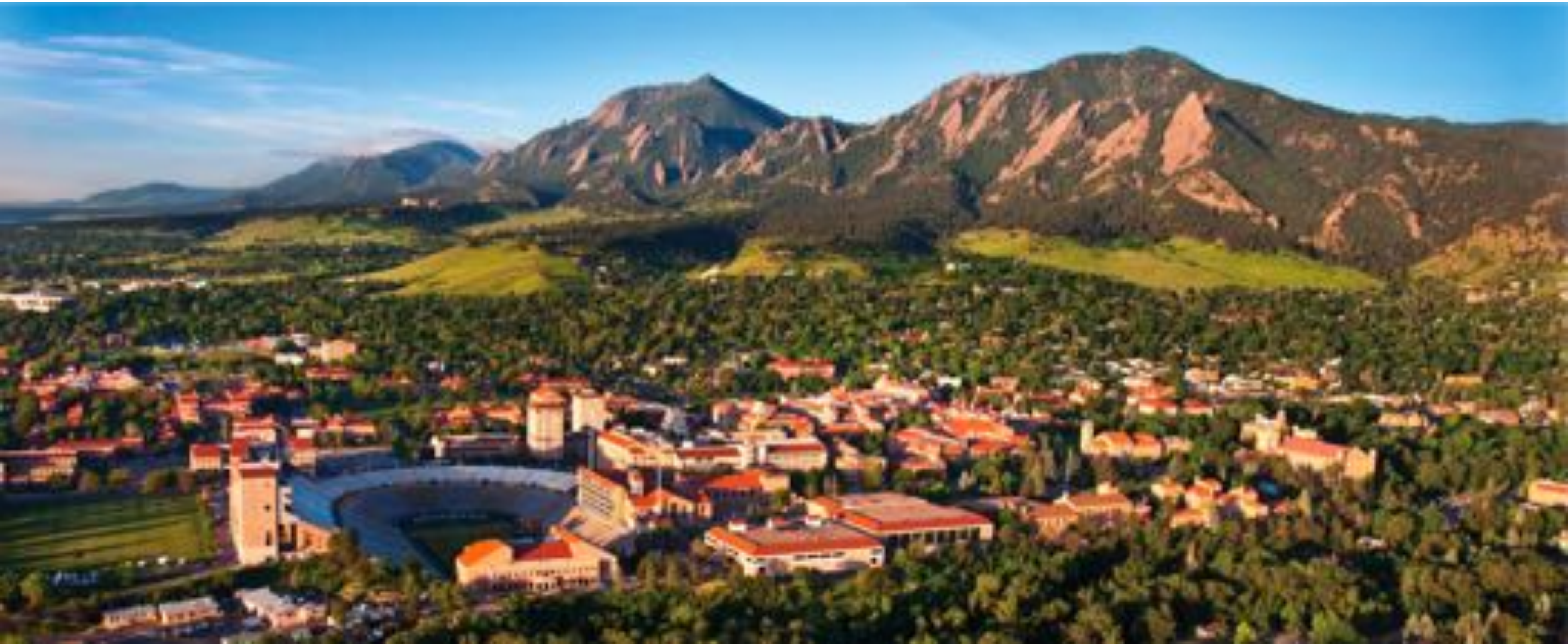
Feedback-based online algorithms for time-varying network optimization

Emiliano Dall'Anese
University of Colorado Boulder

Golden – April 12th, 2019



Decision-making in power systems

City B
Transmission Substation
Biogas
Farm
Network Operating Center
Transmission Substation
Power Plant
Distributed Generation
Wind Turbine
Windfarm
City A
Distribution Substation
Distributed Generation
Network Operating Center
Distribution Substation
Factory
Plug in Hybrid Elec. Vehicles
Smart Appliances
Home Solar
PHEV
Demand Management
Smart Meter

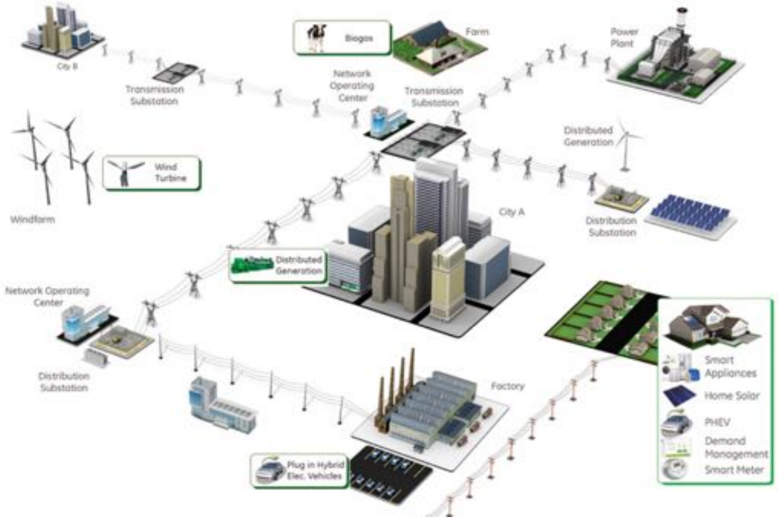
- ❑ Decision-making tasks
  - ❑ Optimal power flow
  - ❑ Demand response
  - ❑ DER aggregation management
  - ❑ *Autonomous energy systems*
  - ❑ …

Decision-making in power systems

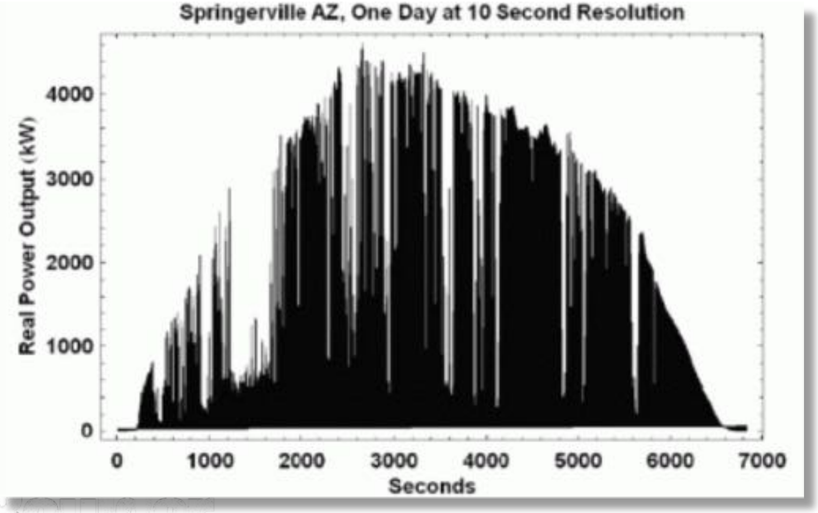
City B
Transmission Substation
Biogas
Farm
Power Plant
Network Operating Center
Transmission Substation
Wind Turbine
Windfarm
Distributed Generation
City A
Distribution Substation
Distributed Generation
Network Operating Center
Distribution Substation
Factory
Plug in Hybrid Elec. Vehicles
Smart Appliances
Home Solar
PHEV
Demand Management
Smart Meter

- ❑ Why are they challenging?

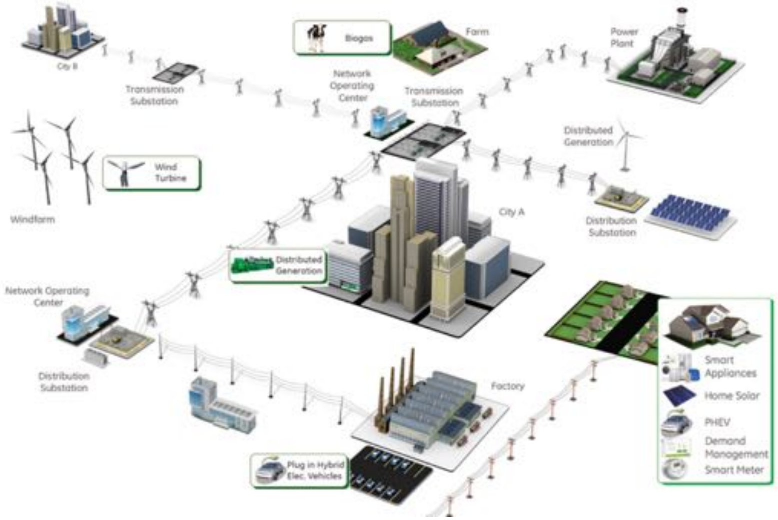
Decision-making in power systems



- ❑ Why are they challenging?
  - ❑ Volatility of operating conditions

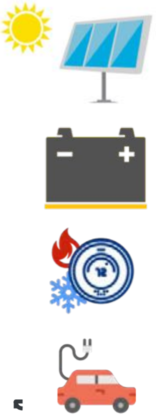


Decision-making in power systems



- ❑ Why are they challenging?
  - ❑ Volatility of operating conditions
  - ❑ Large-scale problems

$46,000$ households ×



$= 184,000$

Decision-making in power systems

City B
Transmission Substation
Biogas
Farm
Network Operating Center
Transmission Substation
Power Plant
Wind Turbine
Windfarm
Distributed Generation
City A
Distribution Substation
Distributed Generation
Network Operating Center
Distribution Substation
Factory
Plug in Hybrid Elec. Vehicles
Smart Appliances
Home Solar
PHEV
Demand Management
Smart Meter

- ❑ Why are they challenging?
  - ❑ Volatility of operating conditions
  - ❑ Large-scale problems
  - ❑ Models difficult to estimate/comp.

$$\mathbf{y}(t) = \mathcal{F}(\underset{\substack{\swarrow \\ 184,000}}{\mathbf{u}(t)}, \underset{\substack{\downarrow \\ 46,000 \text{ households}}}{\mathbf{w}(t)}; t)$$

Real-time decision-making

City B
Transmission Substation
Biogas
Farm
Network Operating Center
Transmission Substation
Power Plant
Wind Turbine
Windfarm
Distributed Generation
City A
Distribution Substation
Distributed Generation
Network Operating Center
Distribution Substation
Factory
Smart Appliances
Home Solar
PHEV
Demand Management
Smart Meter
Plug in Hybrid Elec. Vehicles

- Why are they challenging?
  - Volatility of operating conditions
  - Large-scale problems
  - Models difficult to estimate/comp.

- Desiderata:
  - Computationally-light “optimal” algorithms
  - Scale well with the network size
  - Parallel/distributed implementations
  - Bypass the need for estimating/computing network models

Outline

- Feedback-based online optimization
  - Primal-dual gradient method for time-varying convex problems
  - Time-varying nonconvex problems
- Application to power grids

References

E. Dall'Anese and A. Simonetto, "Optimal power flow pursuit", *IEEE Trans. Smart Grid*, 2016

A Bernstein, E. Dall'Anese, and A. Simonetto, "Online Primal-Dual Methods with Measurement Feedback for Time-Varying Convex Optimization," *IEEE Trans. on Signal Processing*, 2019.

Y. Tang, E. Dall'Anese, A. Bernstein and S. Low. Running primal-dual gradient method for time-varying nonconvex problems, *Mathematical Programming*, 2019 (under review).

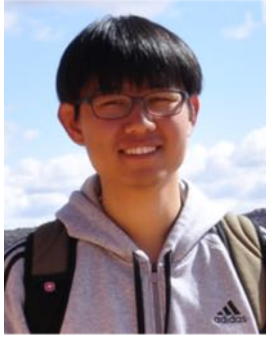
Outline

- Feedback-based online optimization
    - Primal-dual gradient method for time-varying convex problems
    - Time-varying nonconvex problems
- Application to power grids

Acknowledgements

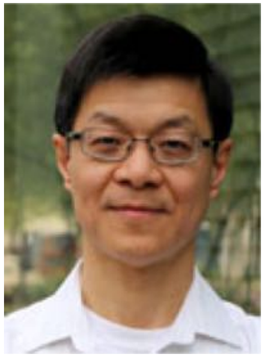
Andrea Simonetto

IBM



Yujie Tang

HARVARD



Steven Low

CALIFORNIA INSTITUTE OF TECHNOLOGY 1891

Andrey Bernstein

NREL NATIONAL RENEWABLE ENERGY LABORATORY

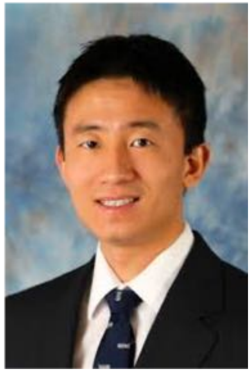
Outline

- ❑ Feedback-based online optimization
  - ❑ Primal-dual gradient method for time-varying convex problems
  - ❑ Time-varying nonconvex problems
- ❑ Application to power grids

Extended acknowledgements

Jorge Cortes

Marcello Colombino



Mingyi Hong

A sample of works in context

- ❑ Time-varying convex optimization:
  - ❑ [Popkov ’05], [Simonetto-Leus ’14], [Simonetto ’17]
  - ❑ [Fazlyab et al ’16], [Rahili-Ren ’17]
- ❑ Nonconvex optimization:
  - ❑ [Dontcev et al’13] Euler-Newton for tracking parametric variational inequalities
  - ❑ [Zavala et al’10] Time-dependent manifold, parametric generalized equation
- ❑ Regret analysis: [Jadbabaie et al ’15], [Mokhtari et al’17], [Chen-Giannakis’17], …
- ❑ Feedback-based *static* optimization:
  - ❑ Communication systems: [Low-Lapsley’99], [Chen-Lau ’12]
  - ❑ Power Systems [next slide]
- ❑ [**this talk**] Feedback-based *online* optimization, linear convergence analysis [Dall’Anese-Simonetto’16], [Bernstein-Dall’Anese-Simonetto’18], [Tang et al’18]
- ❑ Algorithm as feedback controller [Colombino-Dall’Anese-Bernstein’18]

Works in the power systems context

- [Jokic ’09], [Dall’Anese-Dhople-Giannakis ’13]: Static constrained problems, power meas.
- [Zhao et al ’13] Static DC OPF, frequency measurements
- [Bolognani-Zampieri ’13], [Christakou at el ’14]: Static unconstrained problem, voltage meas.
- [Bernstein et al ’15] Online algorithm, power and voltage meas., no analysis
- [Dall’Anese-Simonetto ’16] **Time-varying , voltage meas., linear convergence** [**this talk**]
- [Hauswirth et al ’16] Closed-loop static optimization on power flow manifold
- [Tang et al ’17] Time-varying AC OPF, relaxed, regret analysis
- [Nelson et al’17], [Colombino-Dall’Anese-Bernstein’18] Online, feedback controller
- [Hauswirth et al ’18] Closed-loop time-varying on power flow manifold
- [Tang et al ’19] **Time-varying nonconvex, linear convergence** [**this talk**]

Feedback-based Online Optimization

Modeling optimal trajectories

❑ ***Model***: Continuous-time optimization ➡

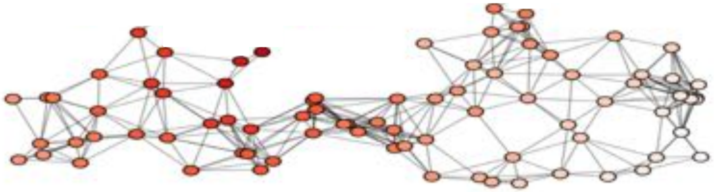
$$\min_{\mathbf{u}} \; c_0(\mathbf{y}(\mathbf{u};t);t) + \sum_i c_i(\mathbf{u}_i;t)$$

$$\text{subject to}: \; \mathbf{u}_i \in \mathcal{U}_i(t), \forall\, i$$

$$\mathbf{g}(\mathbf{y}(\mathbf{u};t);t) \le \mathbf{0}$$

Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$



Modeling optimal trajectories

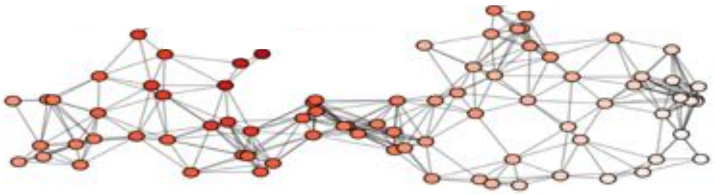
❑ ***Model***: Continuous-time optimization →

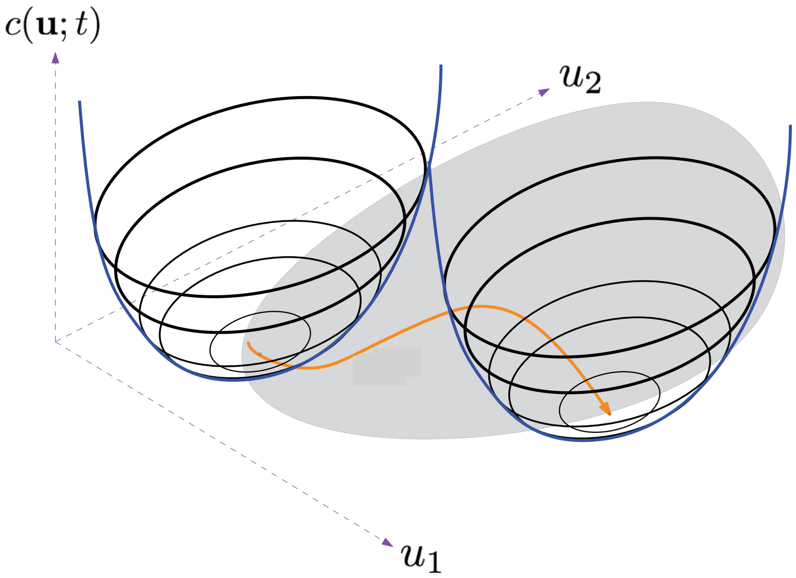
$$\min_{\mathbf{u}} \quad c_0(\mathbf{y}(\mathbf{u};t);t) + \sum_i c_i(\mathbf{u}_i;t)$$

$$\begin{aligned} \text{subject to}: \quad & \mathbf{u}_i \in \mathcal{U}_i(t), \forall\, i \\ & \mathbf{g}(\mathbf{y}(\mathbf{u};t);t) \leq \mathbf{0} \end{aligned}$$

Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$





Modeling optimal trajectories

- ❑ ***Model***: Continuous-time optimization →

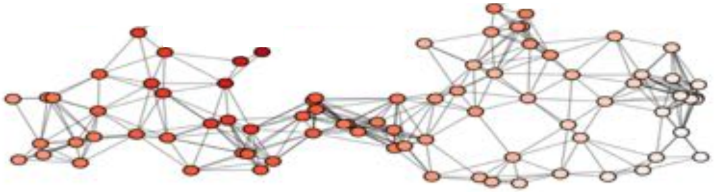
$$\min_{\mathbf{u}} \;\; c_0(\mathbf{y}(\mathbf{u};t);t) + \sum_i c_i(\mathbf{u}_i;t)$$

$$\text{subject to}: \;\; \mathbf{u}_i \in \mathcal{U}_i(t), \forall\, i$$

$$\mathbf{g}(\mathbf{y}(\mathbf{u};t);t) \leq \mathbf{0}$$

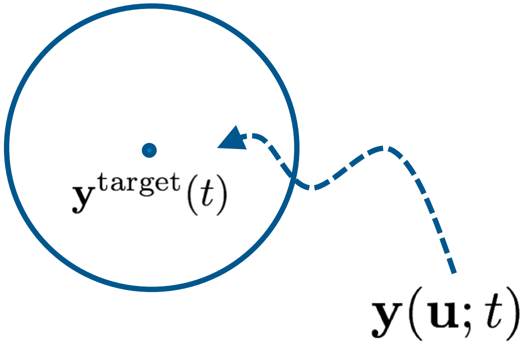
Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$



- ❑ Example:

$$\|\mathbf{y}(\mathbf{u};t) - \mathbf{y}^{\text{target}}(t)\|_2^2 - \nu \leq 0$$



Modeling optimal trajectories

- ❑ ***Model***: Continuous-time optimization ➡

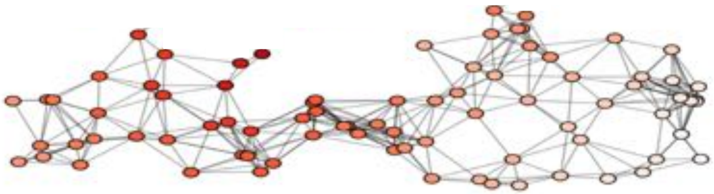
$$\min_{\mathbf{u}} \quad c_0(\mathbf{y}(\mathbf{u};t);t) + \sum_i c_i(\mathbf{u}_i;t)$$

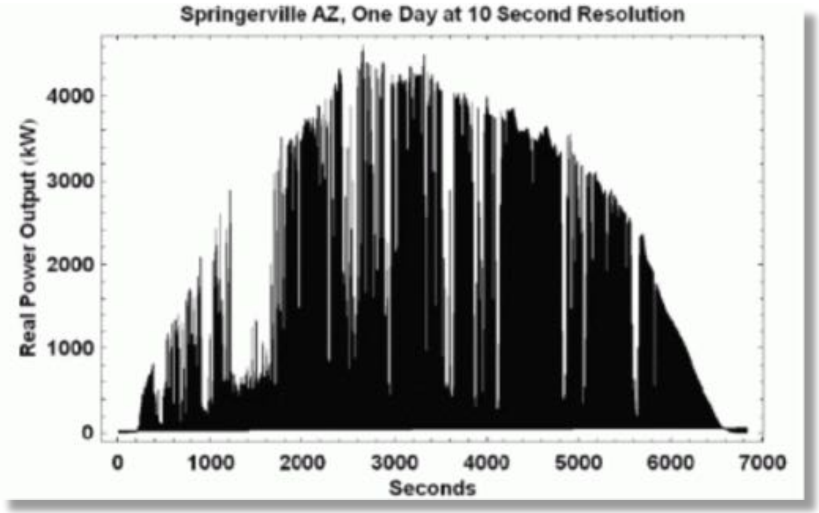
$$\text{subject to}: \quad \mathbf{u}_i \in \mathcal{U}_i(t), \forall\, i$$

$$\mathbf{g}(\mathbf{y}(\mathbf{u};t);t) \leq \mathbf{0}$$

Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$





Modeling optimal trajectories

- ❑ ***Model***: Continuous-time optimization →

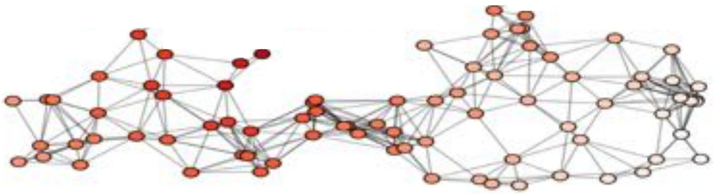
$$\min_{\mathbf{u}} \quad c_0(\mathbf{y}(\mathbf{u};t);t) + \sum_i c_i(\mathbf{u}_i;t)$$

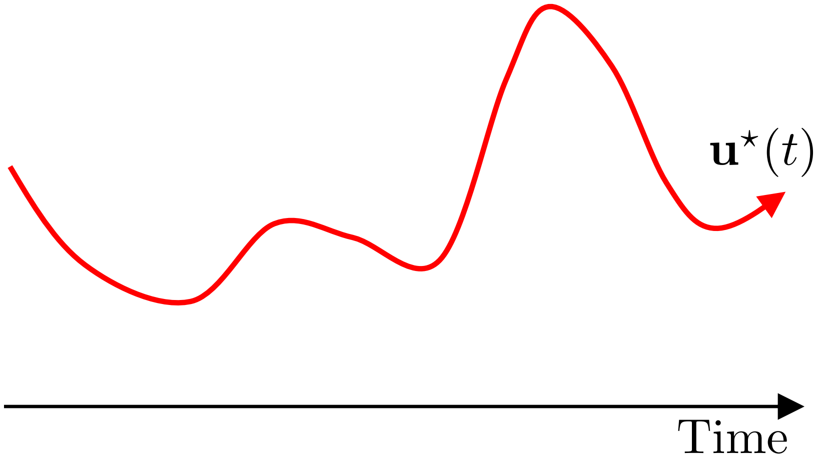
$$\text{subject to}: \quad \mathbf{u}_i \in \mathcal{U}_i(t), \forall\, i$$

$$\mathbf{g}(\mathbf{y}(\mathbf{u};t);t) \leq \mathbf{0}$$

Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$





Modeling optimal trajectories

- ❑ ***Model***: Continuous-time optimization ➡

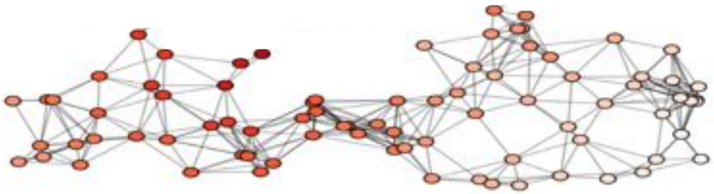
$$\min_{\mathbf{u}} \; c_0(\mathbf{y}(\mathbf{u};t);t) + \sum_i c_i(\mathbf{u}_i;t)$$

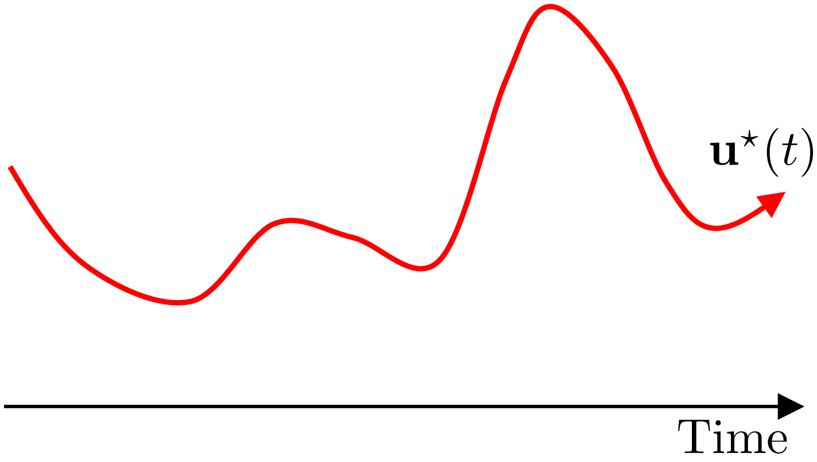
$$\text{subject to}: \; \mathbf{u}_i \in \mathcal{U}_i(t), \forall\, i$$

$$\mathbf{g}(\mathbf{y}(\mathbf{u};t);t) \leq \mathbf{0}$$

Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$





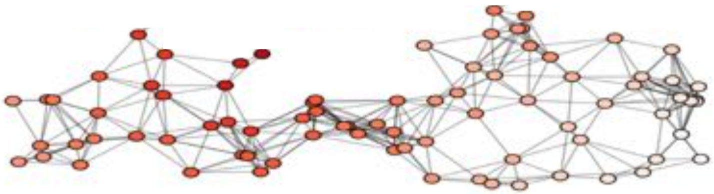
How to drive the time-varying systems towards optimal trajectories?

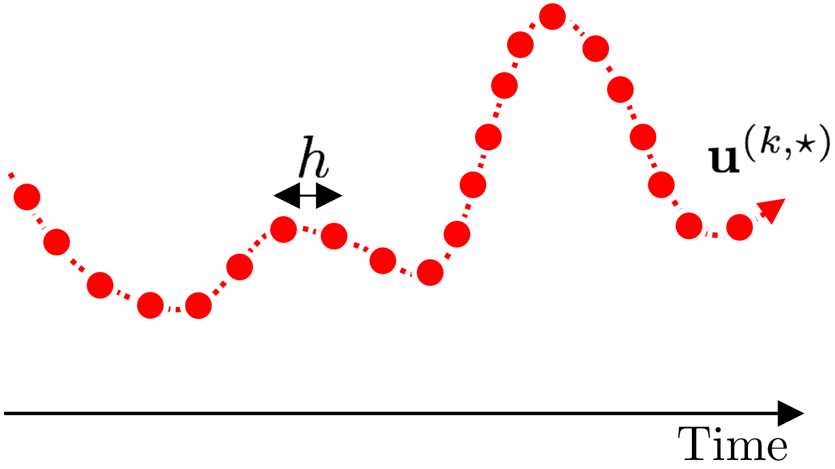
Batch optimization

$$\begin{aligned} \min_{\mathbf{u}} \quad & c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) \\ \text{subject to :} \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i \\ & \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0} \end{aligned}$$

Time-varying systems

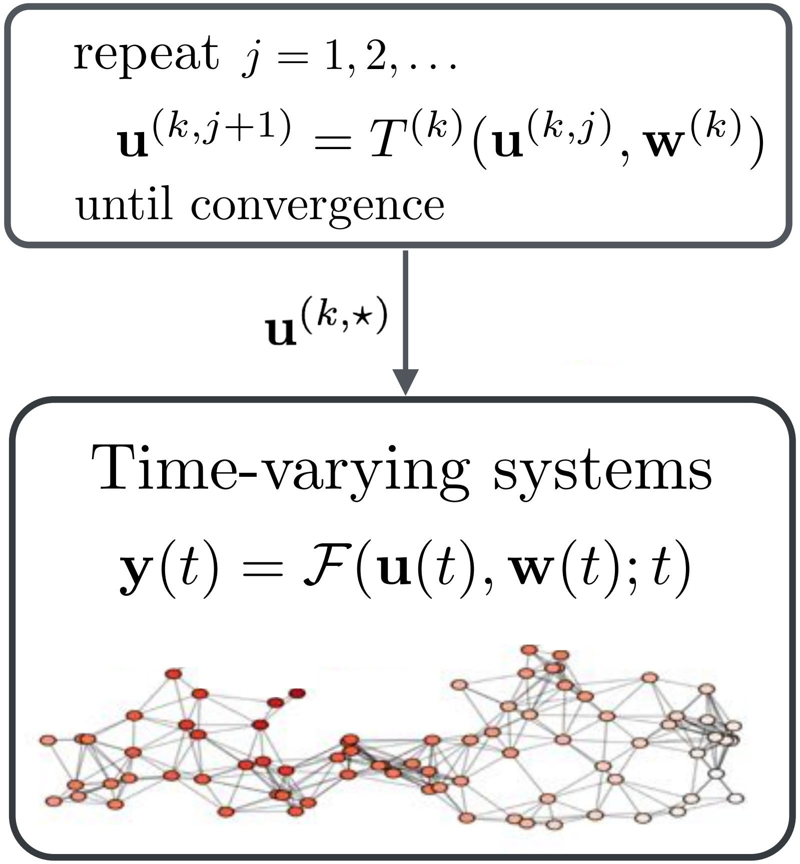
$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$





- ❑ Series of **time-invariant** optimization problems, **sampling intervals** $kh, k \in \mathbb{N}$

Batch optimization



```
repeat $j = 1, 2, \ldots$
    $\mathbf{u}^{(k,j+1)} = T^{(k)}(\mathbf{u}^{(k,j)}, \mathbf{w}^{(k)})$
until convergence
```

$\mathbf{u}^{(k,\star)}$

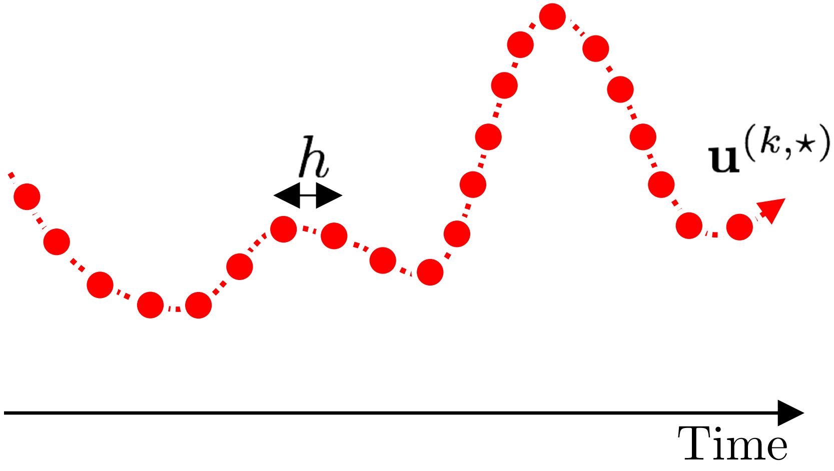
Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$

$$\min_{\mathbf{u}} \quad c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i)$$

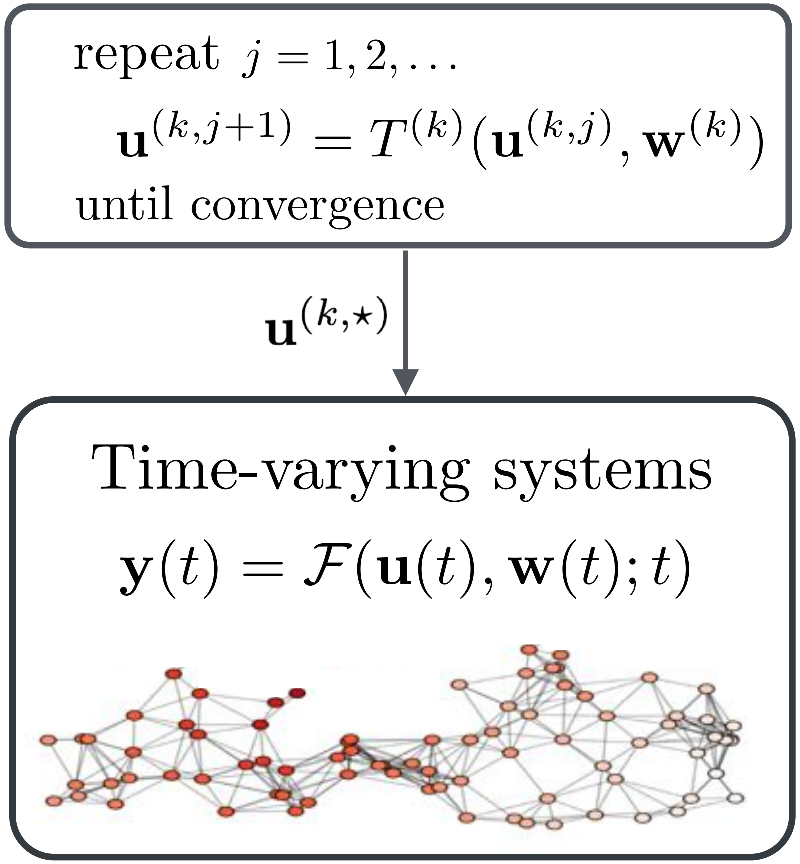
$$\text{subject to}: \quad \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i$$

$$\mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0}$$



- Series of **time-invariant** optimization problems, **sampling intervals** $kh, k \in \mathbb{N}$

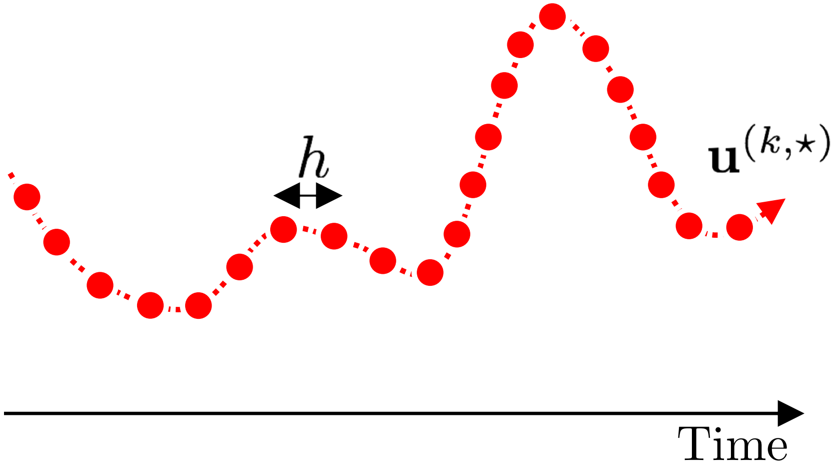
Batch optimization



```
repeat $j = 1, 2, \dots$
    $\mathbf{u}^{(k,j+1)} = T^{(k)}(\mathbf{u}^{(k,j)}, \mathbf{w}^{(k)})$
until convergence
```

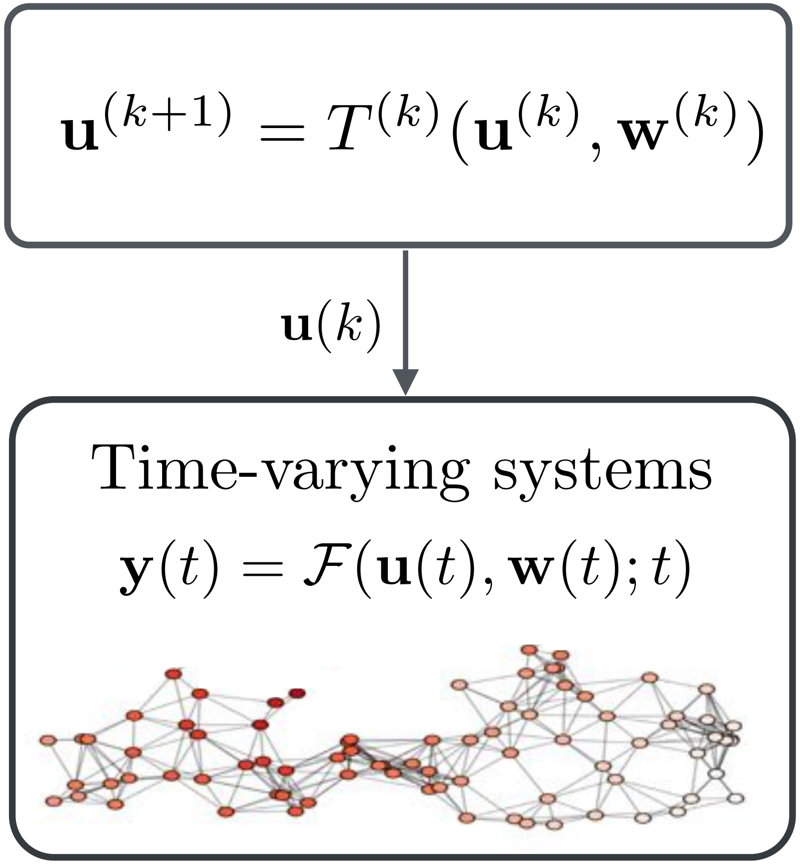
Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$



- **Not practical**: computational limits; convergence time; feed-forward
- What if the convergence time is longer than h ?

Online optimization

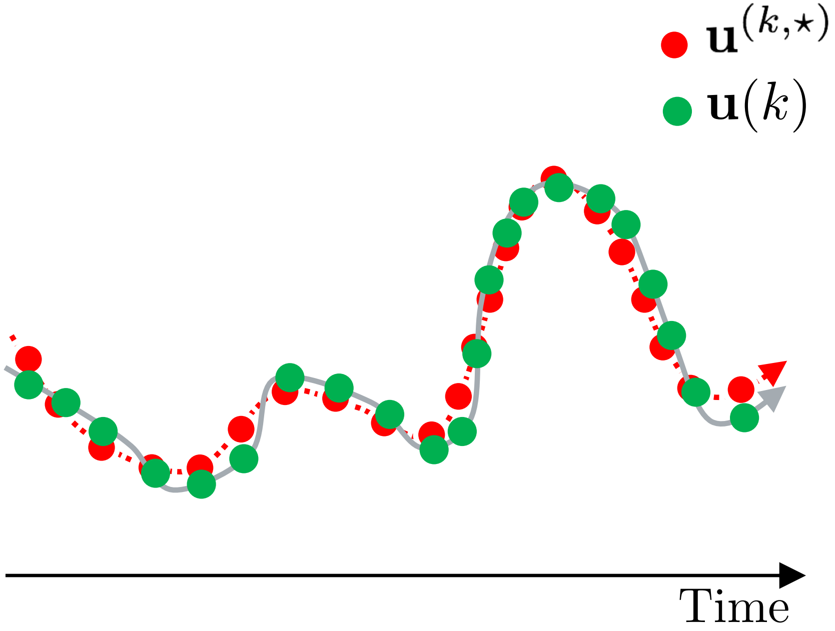


$$\mathbf{u}^{(k+1)} = T^{(k)}(\mathbf{u}^{(k)}, \mathbf{w}^{(k)})$$

$\mathbf{u}(k)$

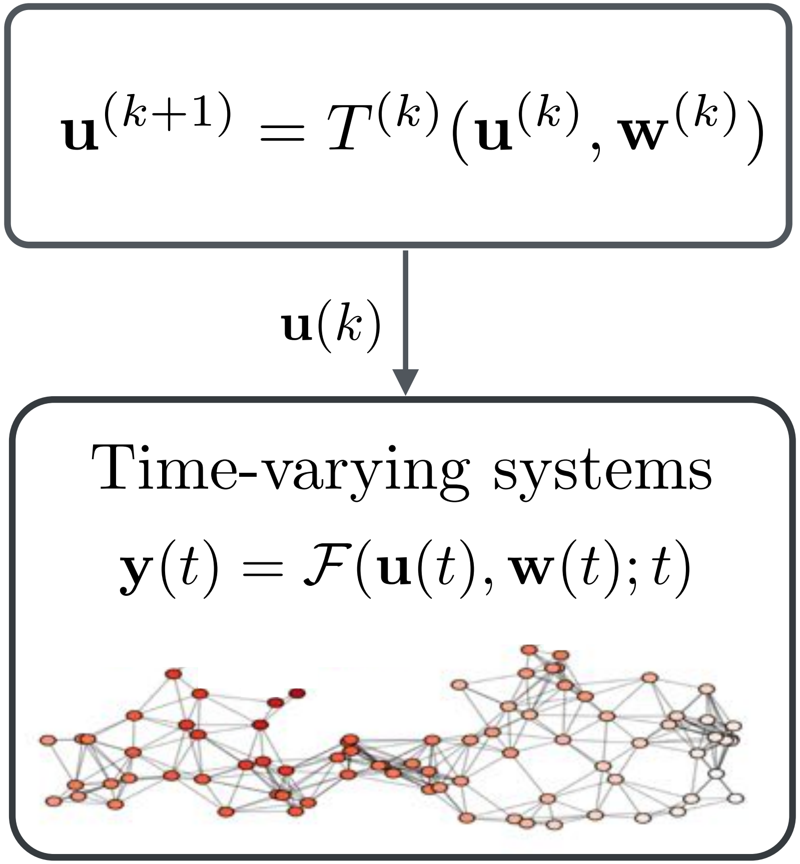
Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$



- **Online** algorithm to track optimal solutions [Dontchev et al'13, Simonetto-Leus'14]

Online optimization

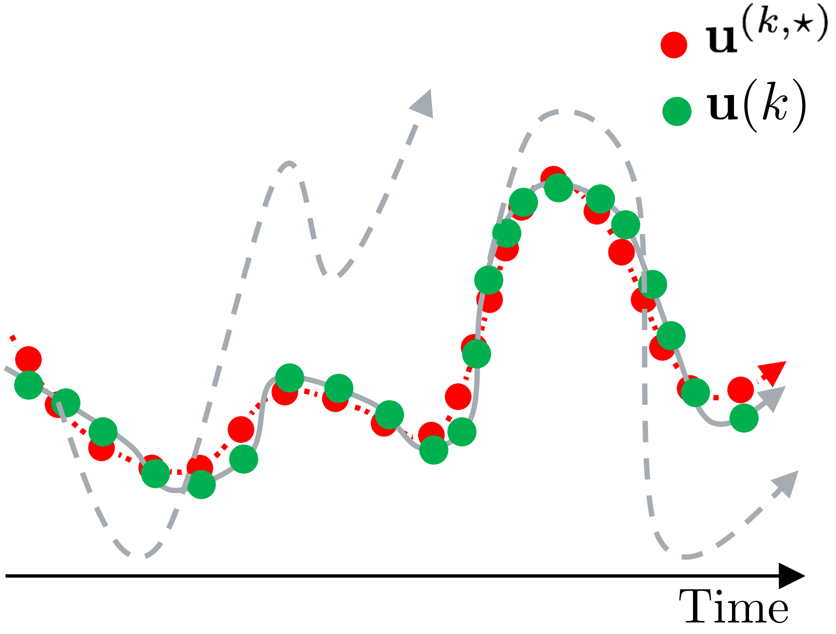


$$\mathbf{u}^{(k+1)} = T^{(k)}(\mathbf{u}^{(k)}, \mathbf{w}^{(k)})$$

$\mathbf{u}(k)$

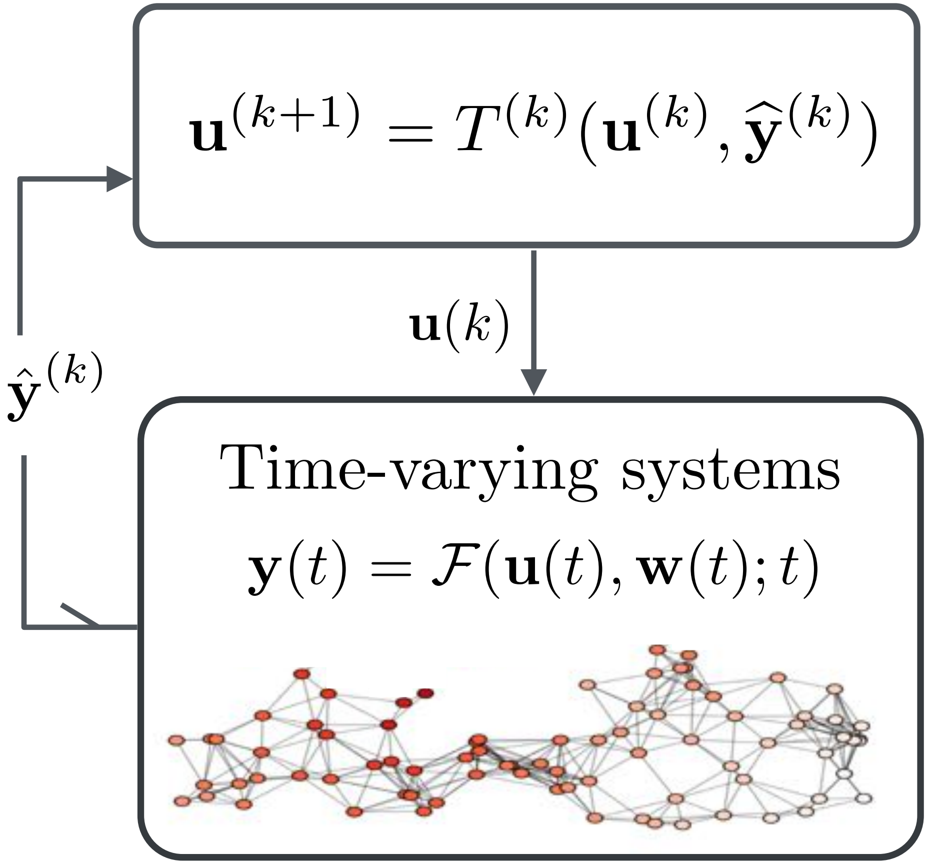
Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$



- **Online** algorithm to track optimal solutions [Dontchev et al'13, Simonetto-Leus'14]
- Feed-forward; needs map $\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$

Feedback-based online optimization



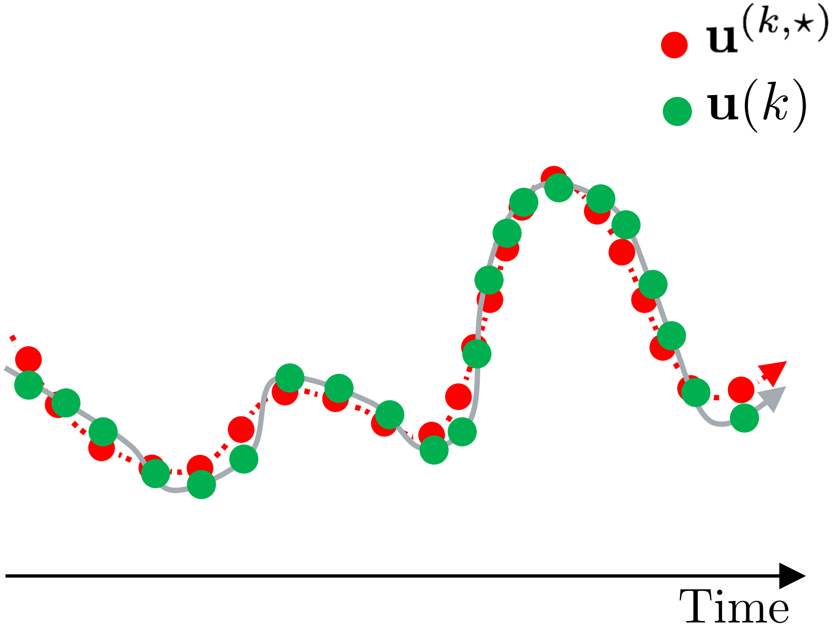
$$\mathbf{u}^{(k+1)} = T^{(k)}(\mathbf{u}^{(k)}, \widehat{\mathbf{y}}^{(k)})$$

$\mathbf{u}(k)$

$\hat{\mathbf{y}}^{(k)}$

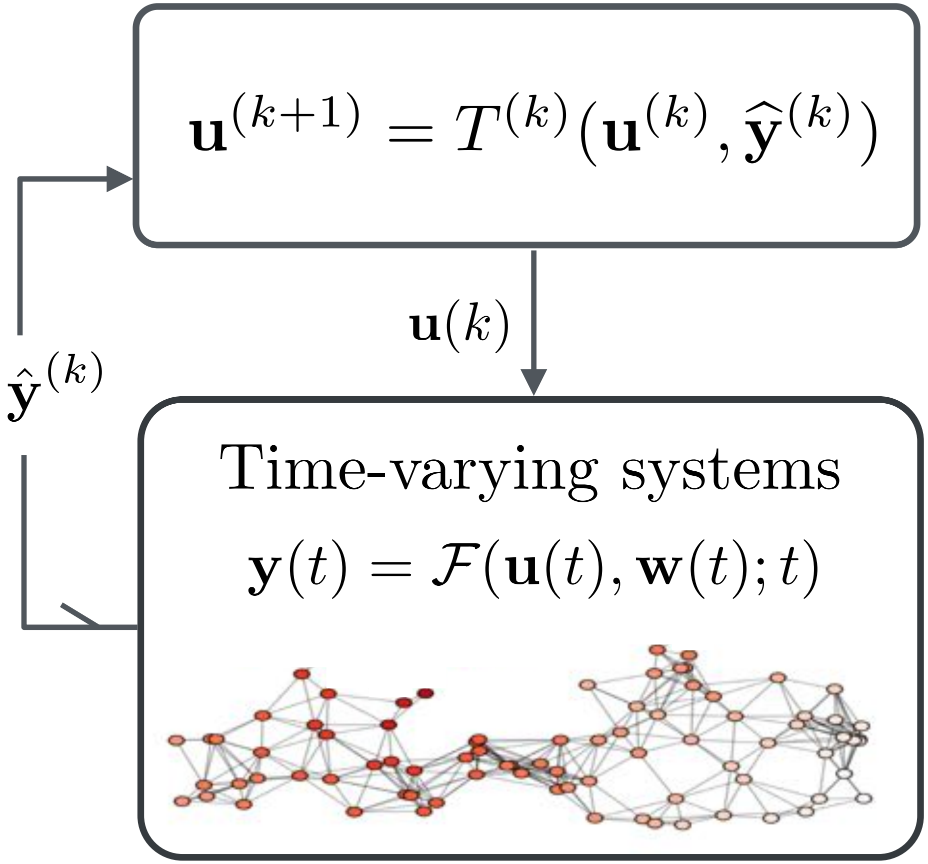
Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$



- ❑ **Feedback-based online:** leverage measurements
- ❑ Measure network output, constraint violation, actuation error

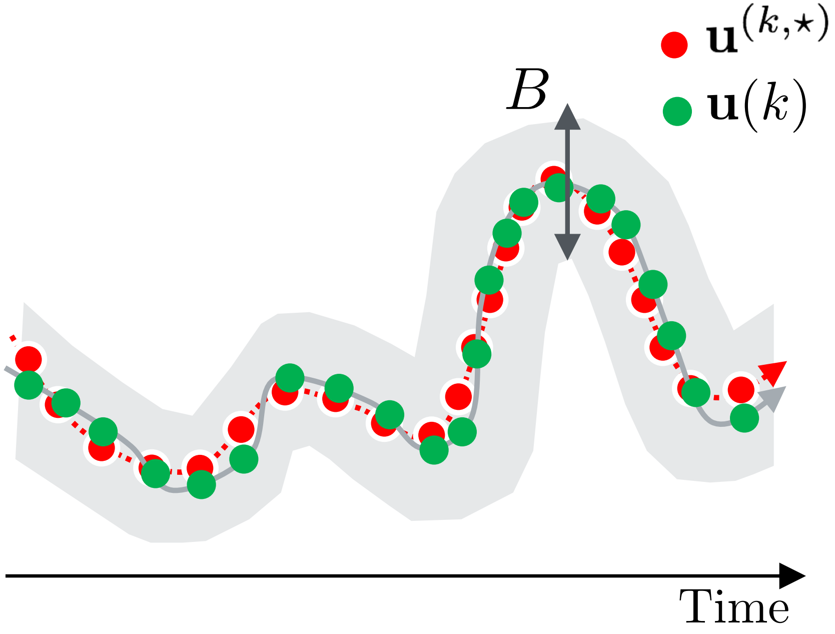
Feedback-based online optimization



$$\mathbf{u}^{(k+1)} = T^{(k)}(\mathbf{u}^{(k)}, \widehat{\mathbf{y}}^{(k)})$$

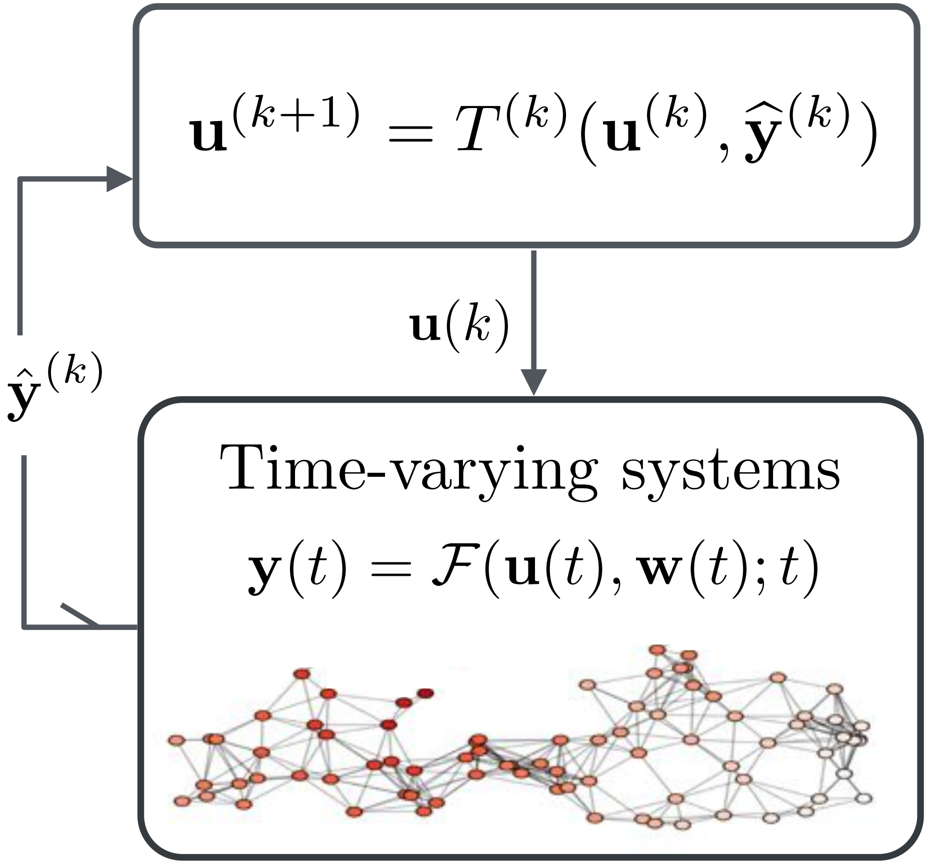
Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$



- ❑ **Feedback-based online:** leverage measurements
- ❑ Derive convergence and tracking results; “stronger” than dynamic regret

Feedback-based online optimization



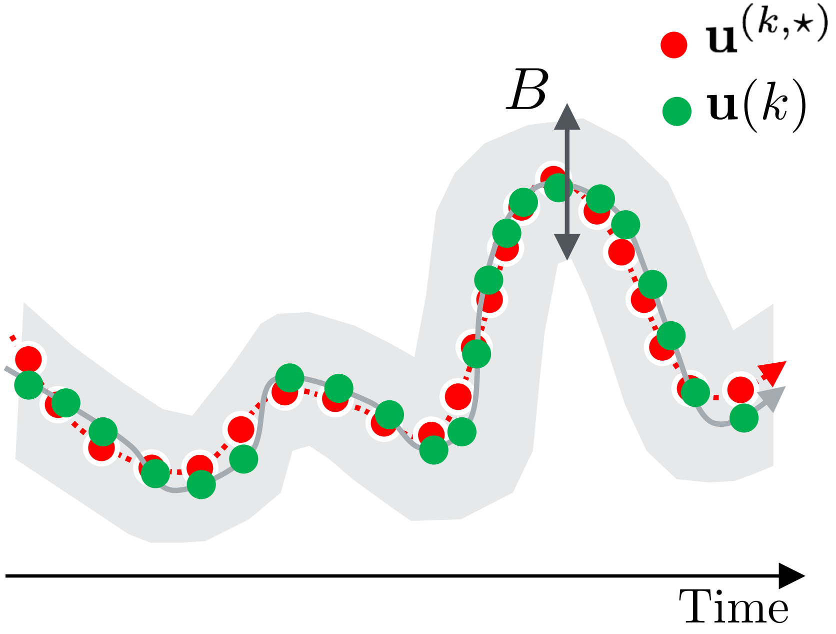
$$\mathbf{u}^{(k+1)} = T^{(k)}(\mathbf{u}^{(k)}, \widehat{\mathbf{y}}^{(k)})$$

$\mathbf{u}(k)$

$\hat{\mathbf{y}}^{(k)}$

Time-varying systems

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$



- Design and analysis of **time-varying** **ϵ-gradient** methods [Bertsekas-Tsitsiklis’00]
- Can model fixed-point arithmetic, finite precision, and inexact maps

Formalizing optimal trajectories

- Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\min_{\mathbf{u}} \quad c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i)$$

$$\text{subject to :} \quad \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i$$

$$\mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0}$$

Classes of problems and systems

- Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$
\begin{aligned}
\min_{\mathbf{u}} \quad & c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) \\
\text{subject to:} \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i \\
& \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0}
\end{aligned}
$$

- **Case 1**: Convex [Dall'Anese-Simonetto'16], [Bernstein-Dall'Anese-Simonetto'19]

  ➡ Convex optimization problem, linear map $\mathbf{y}^{(k)}(\mathbf{u}) = \mathbf{H}\mathbf{u} + \mathbf{D}\mathbf{w}^{(k)}$

Classes of problems and systems

- ❑ Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\begin{aligned} \min_{\mathbf{u}} \quad & c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) \\ \text{subject to :} \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i \\ & \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0} \end{aligned}$$

- ❑ **Case 1**: Convex [Dall'Anese-Simonetto'16], [Bernstein-Dall'Anese-Simonetto'19]

  ➡ Convex optimization problem, linear map $\mathbf{y}^{(k)}(\mathbf{u}) = \mathbf{H}\mathbf{u} + \mathbf{D}\mathbf{w}^{(k)}$

- ❑ **Case 2**: Nonconvex [Tang-Dall'Anese-Bernstein-Low'19]

  ➡ Nonconvex optimization problem, nonlinear map $\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$

Classes of problems and systems

- ❑ Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\begin{aligned} \min_{\mathbf{u}} \quad & c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) \\ \text{subject to}: \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i \\ & \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0} \end{aligned}$$

- ❑ **Case 1**: Convex [Dall'Anese-Simonetto'16], [Bernstein-Dall'Anese-Simonetto'19]

  ➡ Convex optimization problem, linear map $\mathbf{y}^{(k)}(\mathbf{u}) = \mathbf{H}\mathbf{u} + \mathbf{D}\mathbf{w}^{(k)}$

- ❑ **Case 2**: Nonconvex [Tang-Dall'Anese-Bernstein-Low'19]

  ➡ Nonconvex optimization problem, nonlinear map $\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$

- ❑ Common approach: Regularized Lagrangian, primal-dual gradient methods, contraction

Preliminaries

- Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\begin{aligned} \min_{\mathbf{u}} \quad & c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) \\ \text{subject to :} \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i \\ & \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0} \end{aligned}$$

- $L^{(k)}(\mathbf{u}, \boldsymbol{\lambda}) := c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) + \boldsymbol{\lambda}^\mathsf{T} \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u}))$

Preliminaries

- Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\begin{aligned} \min_{\mathbf{u}} \quad & c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) \\ \text{subject to}: \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i \\ & \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0} \end{aligned}$$

- $L^{(k)}(\mathbf{u}, \boldsymbol{\lambda}) := c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) + \boldsymbol{\lambda}^\mathsf{T} \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u}))$

- A Karush-Kuhn-Tucker point: $\mathbf{z}^{(k,\star)} := \{\mathbf{u}^{(k,\star)}, \boldsymbol{\lambda}^{(k,\star)}\}$

Preliminaries

- Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\begin{aligned} \min_{\mathbf{u}} \quad & c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) \\ \text{subject to}: \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i \\ & \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0} \end{aligned}$$

- $L^{(k)}(\mathbf{u}, \boldsymbol{\lambda}) := c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) + \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u}))$

- A Karush-Kuhn-Tucker point: $\mathbf{z}^{(k,\star)} := \{\mathbf{u}^{(k,\star)}, \boldsymbol{\lambda}^{(k,\star)}\}$

- $L_r^{(k)}(\mathbf{u}, \boldsymbol{\lambda}) := L^{(k)}(\mathbf{u}, \boldsymbol{\lambda}) \quad \frac{r}{2}\|\boldsymbol{\lambda}\|_2^2$ [Rockafellar’75], [Koshal-Nedic-Shanbhag’11]

- Critical for linear convergence; but, perturb set of KKT points [Andreani et al’11]

Preliminaries

- ❑ Time-varying problem (intervals $kh, k \in \mathbb{N}$):

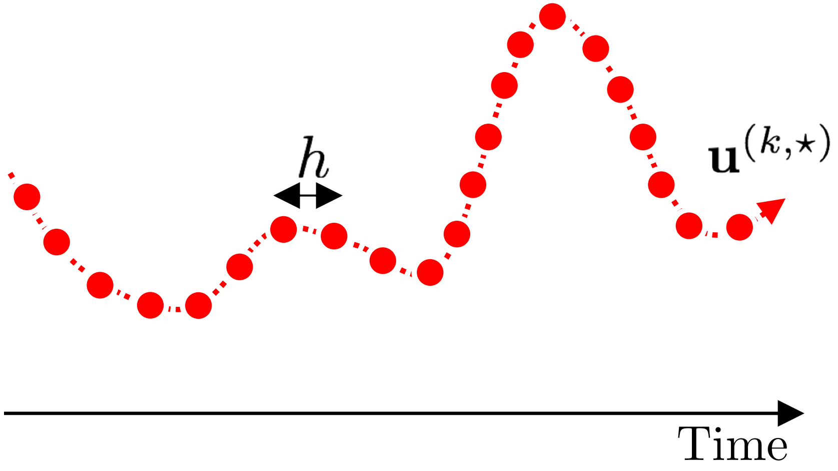
$$\min_{\mathbf{u}} \quad c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i)$$

$$\text{subject to :} \quad \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i$$

$$\mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0}$$

- ❑ $\sigma^{(k)} := h \quad {}^{1}\|\mathbf{z}^{(*,k+1)} \qquad \mathbf{z}^{(*,k)}\|_2$

$\approx$ *Drifting / gradient*



Convex setting

- Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\min_{\mathbf{u}} \quad c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i)$$

$$\text{subject to}: \quad \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i$$

$$\mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0}$$

- **Case 1**: Convex [Dall'Anese-Simonetto'16], [Bernstein-Dall'Anese-Simonetto'19]

  ➡ Convex optimization problem, linear map $\mathbf{y}^{(k)}(\mathbf{u}) = \mathbf{H}\mathbf{u} + \mathbf{D}\mathbf{w}^{(k)}$

- **Case 2**: Nonconvex [Tang-Dall'Anese-Bernstein-Low'19]

  ➡ Nonconvex optimization problem, nonlinear map $\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$

Convex setting

- Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\begin{aligned} \min_{\mathbf{u}} \quad & c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i) \\ \text{subject to}: \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i \\ & \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0} \end{aligned}$$

- *(As.)* The set $\mathcal{U}_i^{(k)}$ is convex and compact for all k .
- *(As.)* Functions are cont. differentiable, Lipschitz gradient, bounded gradient
- *(As.)* Cost is m-strongly convex.
- *(As.)* Problem is feasible and Slater's condition holds.

Online primal-dual method

- ❑ Online implementation of the primal-dual method ↻ k

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}} \left\{ \mathbf{u}^{(k)} - \alpha \nabla_{\mathbf{u}} L_r^{(k)}(\mathbf{u}^{(k)}, \boldsymbol{\lambda}^{(k)}) \right\}$$

\\\\ primal gradient descent

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathcal{D}} \left\{ \boldsymbol{\lambda}^{(k)} + \alpha \nabla_{\boldsymbol{\lambda}} L_r^{(k)}(\mathbf{u}^{(k)}, \boldsymbol{\lambda}^{(k)}) \right\}$$

\\\\ dual gradient ascent

Online primal-dual method

- ❑ Online implementation of the primal-dual method ↺ k

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}}\left\{\mathbf{u}^{(k)} - \alpha \nabla_{\mathbf{u}} L_r^{(k)}(\mathbf{u}^{(k)}, \boldsymbol{\lambda}^{(k)})\right\}$$

\\ primal gradient descent

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathcal{D}}\left\{\boldsymbol{\lambda}^{(k)} + \alpha \nabla_{\boldsymbol{\lambda}} L_r^{(k)}(\mathbf{u}^{(k)}, \boldsymbol{\lambda}^{(k)})\right\}$$

\\ dual gradient ascent

- ❑ $r > 0$: Strongly monotone and L_{map}-Lipschitz map (locally for nonconvex)

→ Contractive arguments → Q-linear convergence

Online primal-dual method

❑ Online implementation of the primal-dual method $\circlearrowleft k$

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}} \left\{ \mathbf{u}^{(k)} - \alpha \nabla_{\mathbf{u}} L_r^{(k)}(\mathbf{u}^{(k)}, \boldsymbol{\lambda}^{(k)}) \right\}$$

\\ primal gradient descent

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathcal{D}} \left\{ \boldsymbol{\lambda}^{(k)} + \alpha \nabla_{\boldsymbol{\lambda}} L_r^{(k)}(\mathbf{u}^{(k)}, \boldsymbol{\lambda}^{(k)}) \right\}$$

\\ dual gradient ascent

❑ $r = 0$: Monotone, Lipschitz map

➡ Dynamic regret analysis [Bernstein-Dall'Anese-Simonetto'18]

Online primal-dual method

- Online implementation of the primal-dual method

$$\mathbf{u}^{(k+1)} = \text{proj}_{\mathcal{U}^{(k)}}\Big\{\mathbf{u}^{(k)} \quad \alpha\Big(\sum_i \nabla c_i^{(k)}(\mathbf{u}^{(k)}) + \mathbf{H}^\top \nabla c_0^{(k)}(\mathbf{Hu}^{(k)} + \mathbf{Dw}^{(k)}) + \mathbf{H}^\top[\mathbf{J}_g(\mathbf{Hu}^{(k)} + \mathbf{Dw}^{(k)})]^\top \boldsymbol{\lambda}^{(k)}\Big)\Big\}$$

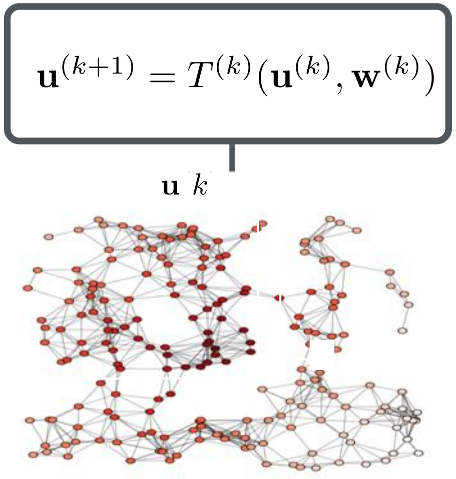
$$\boldsymbol{\lambda}^{(k+1)} = \text{proj}_{\mathcal{D}^{(k)}}\Big\{(1-\alpha r)\boldsymbol{\lambda}^{(k)} + \alpha \mathbf{g}^{(k)}(\mathbf{Hu}^{(k)} + \mathbf{Dw}^{(k)})\Big\}$$

Online primal-dual method

- Online implementation of the primal-dual method

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}}\Big\{\mathbf{u}^{(k)} \quad \alpha\Big(\sum_i \nabla c_i^{(k)}(\mathbf{u}^{(k)}) + \mathbf{H}^\mathsf{T}\nabla c_0^{(k)}(\mathbf{H}\mathbf{u}^{(k)} + \mathbf{D}\mathbf{w}^{(k)}) + \mathbf{H}^\mathsf{T}[\mathbf{J}_g(\mathbf{H}\mathbf{u}^{(k)} + \mathbf{D}\mathbf{w}^{(k)})]^\mathsf{T}\boldsymbol{\lambda}^{(k)}\Big)\Big\}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathcal{D}^{(k)}}\Big\{(1-\alpha r)\boldsymbol{\lambda}^{(k)} + \alpha\mathbf{g}^{(k)}(\mathbf{H}\mathbf{u}^{(k)} + \mathbf{D}\mathbf{w}^{(k)})\Big\}$$



$$\mathbf{u}^{(k+1)} = T^{(k)}(\mathbf{u}^{(k)}, \mathbf{w}^{(k)})$$

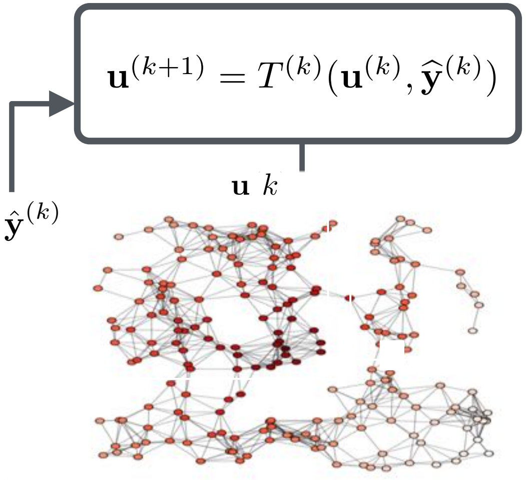
- Feed-forward / autonomous system
- Model-based
- Map evaluated at each iteration

Online primal-dual method

- ❑ Online implementation of the primal-dual method

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}}\Big\{\mathbf{u}^{(k)} - \alpha\Big(\sum_i \nabla c_i^{(k)}(\mathbf{u}^{(k)}) + \mathbf{H}^\top \nabla c_0^{(k)}(\widehat{\mathbf{y}}^{(k)}) + \mathbf{H}^\top [\mathbf{J}_g(\widehat{\mathbf{y}}^{(k)})]^\top \boldsymbol{\lambda}^{(k)}\Big)\Big\}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathcal{D}^{(k)}}\left\{(1-\alpha r)\boldsymbol{\lambda}^{(k)} + \alpha\widehat{\mathbf{g}}^{(k)}\right\}$$



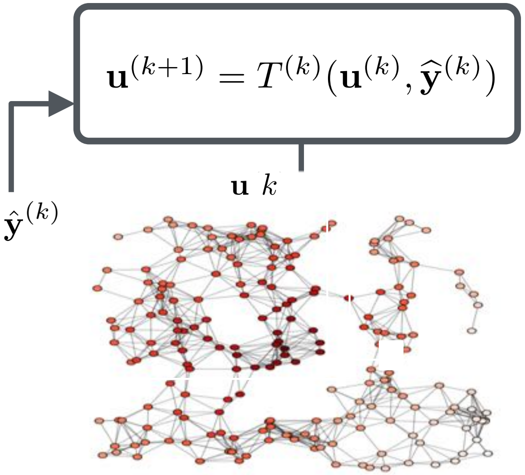
$$\mathbf{u}^{(k+1)} = T^{(k)}(\mathbf{u}^{(k)}, \widehat{\mathbf{y}}^{(k)})$$

Online primal-dual method

- ❑ Online implementation of the primal-dual method

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}}\Big\{\mathbf{u}^{(k)} - \alpha\Big(\sum_i \nabla c_i^{(k)}(\mathbf{u}^{(k)}) + \mathbf{H}^\mathsf{T}\nabla c_0^{(k)}(\widehat{\mathbf{y}}^{(k)}) + \mathbf{H}^\mathsf{T}[\mathbf{J}_g(\widehat{\mathbf{y}}^{(k)})]^\mathsf{T}\boldsymbol{\lambda}^{(k)}\Big)\Big\}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathcal{D}^{(k)}}\Big\{(1-\alpha r)\boldsymbol{\lambda}^{(k)} + \alpha\widehat{\mathbf{g}}^{(k)}\Big\}$$



$$\mathbf{u}^{(k+1)} = T^{(k)}(\mathbf{u}^{(k)}, \hat{\mathbf{y}}^{(k)})$$

Feedback replaces model

$$\mathbf{y}^{(k)}(\mathbf{u}) = \mathbf{H}\mathbf{u} + \mathbf{D}\mathbf{w}^{(k)}$$

or constraint evaluation

- ❑ No need for: $\mathbf{D}$ and $\mathbf{w}^{(k)}$
- ❑ Reduced communication

Bounded error

- *(Assumption)* $\|\hat{\mathbf{y}}^{(k)} - \mathbf{y}^{(k)}(\mathbf{u}^{(k)})\|_2 \leq e_y$ for all k.

- Bounded:
  - Measurement/quantization errors
  - Modeling mismatches
  - Actuation errors
  - Time-scale separation

Bounded error

- *(Assumption)* $\|\hat{\mathbf{y}}^{(k)} - \mathbf{y}^{(k)}(\mathbf{u}^{(k)})\|_2 \leq e_y$ for all k.

- Bounded:
  - Measurement/quantization errors
  - Modeling mismatches
  - Actuation errors
  - Time-scale separation

Lemma. The errors in the primal and dual gradient steps can be bounded, respectively, as:

$$e_p \leq (L_o + M_\lambda M_I L_g)\|\mathbf{H}\|_2 e_y$$
$$e_d \leq M_g e_y \ .$$

From … Lipschitz-continuous gradients, compact sets, # of non-linear inequalities

Convergence

Theorem. If α is chosen such that:

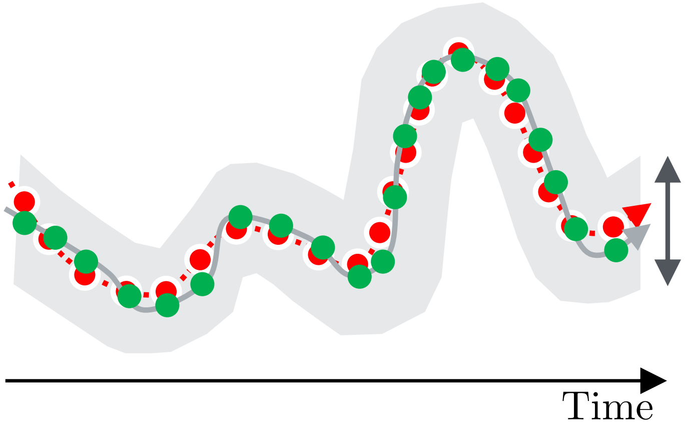
$$\alpha < \frac{\min\{m,r\}}{L_{\text{map}}^2}$$

then the following holds for the algorithm:

$$\lim_{k\to+\infty} \sup \|\mathbf{z}^{(k)} \quad \mathbf{z}^{(k,\star)}\|_2 \leq \frac{1}{1 \quad \rho(\alpha)} \left(\alpha\sqrt{e_p^2 + e_d^2} + \quad + K\right)$$

Where $\sigma := \sup \sigma^{(k)}$, $K := (1+\rho(\alpha)) \sup \sqrt{\frac{r}{2m}\|\boldsymbol{\lambda}^{(k,\star)}\|_2}$, and

$$\rho(\alpha) := \left[1 - 2\alpha \min\{m,r\} + \alpha^2 L_{\text{map}}^2\right]^{\frac{1}{2}}.$$



$$\frac{1}{1-\rho(\alpha)}\left(\alpha\sqrt{e_p^2 + e_d^2} + \quad + K\right)$$

Nonconvex setting

- Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\begin{aligned} \min_{\mathbf{u}} \quad & c^{(k)}(\mathbf{u}) \\ \text{s. to}: \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall i \\ & \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0} \end{aligned}$$

- **Case 1**: Convex [Dall'Anese-Simonetto'16]

  ➡ Convex optimization problem, linear map $\mathbf{y}^{(k)}(\mathbf{u}) = \mathbf{H}\mathbf{u} + \mathbf{D}\mathbf{w}^{(k)}$

- **Case 2**: Nonconvex [Tang-Dall'Anese-Bernstein-Low'19]

  ➡ Nonconvex optimization problem, nonlinear map $\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$

Nonconvex setting

- Time-varying problem (intervals $kh, k \in \mathbb{N}$):

$$\begin{aligned} \min_{\mathbf{u}} \quad & c^{(k)}(\mathbf{u}) \\ \text{s. to}: \quad & \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i \\ & \mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0} \end{aligned}$$

- *(As.)* Functions are twice continuously differentiable

- *(As.)* Problem is feasible and there is no $\boldsymbol{\lambda} \in \mathbb{R}_+^d \backslash \{0\}$ such that:

$$\boldsymbol{\lambda}^\mathsf{T} \mathbf{g}^{(k)}(\mathbf{u}^{(k,\star)}) = 0 \quad \text{and} \quad [\mathbf{J}_g^{(k)}(\mathbf{u}^{(k,\star)})]^\mathsf{T} \boldsymbol{\lambda} \in \mathcal{N}_{\mathcal{U}^{(k)}}(\mathbf{u}^{(k,\star)})$$

[generalization of the Mangasarian-Fromovitz constraint qualification]

Online primal-dual method

- ❑ Online implementation of the primal-dual method

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}}\left\{\mathbf{u}^{(k)} \quad \alpha\Big(\nabla c^{(k)}(\mathbf{u}^{(k)}) + [\mathbf{J}_{\mathcal{F}}(\mathbf{u}^{(k)})]^{\mathsf{T}}[\mathbf{J}_g(\mathbf{y}^{(k)})]^{\mathsf{T}}\boldsymbol{\lambda}^{(k)}\Big)\right\}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathbb{R}^d_+}\left\{(1-\alpha\eta r)\boldsymbol{\lambda}^{(k)} + \eta\alpha\mathbf{g}^{(k)}(\mathbf{y}^{(k)})\right\}$$

where $\mathbf{y}^{(k)} = \mathcal{F}^{(k)}(\mathbf{u}^{(k)}, \mathbf{w}^{(k)})$

Online primal-dual method

- Online implementation of the primal-dual method

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}}\left\{\mathbf{u}^{(k)} \quad \alpha\left(\nabla c^{(k)}(\mathbf{u}^{(k)}) + [\mathbf{J}_{\mathcal{F}}(\mathbf{u}^{(k)})]^{\mathsf{T}}[\mathbf{J}_g(\mathbf{y}^{(k)})]^{\mathsf{T}}\boldsymbol{\lambda}^{(k)}\right)\right\}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathbb{R}^d_+}\left\{(1-\alpha\eta r)\boldsymbol{\lambda}^{(k)} + \eta\alpha\mathbf{g}^{(k)}(\mathbf{y}^{(k)})\right\}$$

where $\mathbf{y}^{(k)} = \mathcal{F}^{(k)}(\mathbf{u}^{(k)}, \mathbf{w}^{(k)})$

- No results for online regularized primal-dual methods for nonconvex problems

- Let's work with this: $\|\mathbf{z}\|_\eta := (\|\mathbf{u}\|_2^2 + \eta^{-1}\|\boldsymbol{\lambda}\|_2^2)^{1/2}$

Some insights

❑ Existence of a set of feasible parameters for locally contractive iterations

$$\Lambda_m \ \delta \ > M_{nc} \ \delta \ \sup_k \|\boldsymbol{\lambda}^{(k,\star)}\|$$

Some insights

- ❑ Existence of a set of feasible parameters for locally contractive iterations

$$\Lambda_m \;\; \delta \;\; > M_{nc} \;\; \delta \;\; \sup_k \|\boldsymbol{\lambda}^{(k,\star)}\|$$

where: $\Lambda_m(\delta) := \inf_t \inf_{\mathbf{x}:\|\mathbf{x}\|\leq\delta} \lambda_{\min}\left(\overline{H}^{nc}_{L_r}(\mathbf{x},t) + \frac{1}{2}\sum_{i=1}^{m} \lambda_i^\star(t)\overline{H}_{g_i^c}(\mathbf{x},t)\right)$

$$\downarrow$$

$$\overline{H}_{L_r^{nc}}(\mathbf{x},t) := \int_0^1 \nabla^2_{uu} L_r^{nc}(\mathbf{u}^\star(t) + \theta\mathbf{x}, \quad {}^\star(t), t)\, d\theta$$

$$M_{nc}(\delta) := \sup_t \sup_{\mathbf{x}:\|\mathbf{x}\|\leq} \left\|D^2_{uu}\mathbf{g}^{nc}(\mathbf{u}^\star(t) + \mathbf{x}, t)\right\|$$

Some insights

❑ Existence of a set of feasible parameters for locally contractive iterations

$$\Lambda_m\ \ \delta\ \ > M_{nc}\ \ \delta\ \ \sup_k \|\boldsymbol{\lambda}^{(k,\star)}\|$$

where: $\Lambda_m(\delta) := \inf_t \inf_{\mathbf{x}:\|\mathbf{x}\|\leq\delta} \lambda_{\min}\left(\overline{H}^{nc}_{L_r}(\mathbf{x},t) + \frac{1}{2}\sum_{i=1}^{m} \lambda_i^\star(t)\overline{H}_{g_i^c}(\mathbf{x},t)\right)$

↓

$$\overline{H}_{L_r^{nc}}(\mathbf{x},t) := \int_0^1 \nabla^2_{uu} L_r^{nc}(\mathbf{u}^\star(t)+\theta\mathbf{x},\ \ ^\star(t),t)\,d\theta$$

$$M_{nc}(\delta) := \sup_t \sup_{\mathbf{x}:\|\mathbf{x}\|\leq} \left\|D^2_{uu}\mathbf{g}^{nc}(\mathbf{u}^\star(t)+\mathbf{x},t)\right\|$$

❑ *“locally strongly convex in a neighborhood of an optimal primal variable”*

❑ Related to the concept of strongly regular point [Dontchev et al’13]

Convergence

Theorem. Assume that $\Lambda_m \;\; \delta \;\; > M_{nc} \;\; \delta \;\; \sup_k \|\boldsymbol{\lambda}^{(k,\star)}\|$ and $\|\mathbf{z}^{(1)} \quad \mathbf{z}^{(\star,1)}\|_\eta < \delta$.

Then the following holds for the algorithm:

$$\lim_{k\to+\infty} \sup \|\mathbf{z}^{(k)} \quad \mathbf{z}^{(k,\star)}\|_2 \leq \frac{\rho(\alpha,\eta)}{1 \quad \rho(\alpha,\eta)} + \frac{K'}{1 \quad \rho(\alpha,\eta)}$$

where $K' := \sqrt{2\eta}\alpha r \sup \|\boldsymbol{\lambda}^{(k,\star)}\|$, and $\rho(\alpha,\eta) < 1$.

Convergence

Theorem. Assume that $\Lambda_m \quad \delta \quad > M_{nc} \quad \delta \quad \sup_k \|\boldsymbol{\lambda}^{(k,\star)}\|$ and $\|\mathbf{z}^{(1)} \qquad \mathbf{z}^{(\star,1)}\|_\eta < \delta$.

Then the following holds for the algorithm:

$$\lim_{k\to+\infty} \sup \|\mathbf{z}^{(k)} \qquad \mathbf{z}^{(k,\star)}\|_2 \leq \frac{\rho(\alpha,\eta)}{1 \quad \rho(\alpha,\eta)} + \frac{K'}{1 \quad \rho(\alpha,\eta)}$$

where $K' := \sqrt{2\eta}\alpha r \sup \|\boldsymbol{\lambda}^{(k,\star)}\|$, and $\rho(\alpha,\eta) < 1$.

- ❑ Sufficient conditions to ensure $\rho(\alpha,\eta) < 1$

How about measurements?

- Online implementation of the primal-dual method

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}}\left\{\mathbf{u}^{(k)} - \alpha\left(\nabla c^{(k)}(\mathbf{u}^{(k)}) + [\mathbf{J}_{\mathcal{F}}(\mathbf{u}^{(k)})]^{\mathsf{T}}[\mathbf{J}_g(\mathbf{y}^{(k)})]^{\mathsf{T}}\boldsymbol{\lambda}^{(k)}\right)\right\}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathbb{R}^d_+}\left\{(1-\alpha\eta r)\boldsymbol{\lambda}^{(k)} + \eta\alpha\mathbf{g}^{(k)}(\mathbf{y}^{(k)})\right\}$$

How about measurements?

- ❑ Online feedback-based implementation of the primal-dual method

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}}\left\{\mathbf{u}^{(k)} - \alpha\left(\nabla c^{(k)}(\mathbf{u}^{(k)}) + [\mathbf{J}_{\mathcal{F}}(\mathbf{u}^{(k)})]^{\mathsf{T}}[\widehat{\mathbf{J}}_g]^{\mathsf{T}}\boldsymbol{\lambda}^{(k)}\right)\right\}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathbb{R}^d_+}\left\{(1-\alpha\eta r)\boldsymbol{\lambda}^{(k)} + \alpha\eta\widehat{\mathbf{g}}^{(k)}\right\}$$

How about measurements?

❑ Online feedback-based implementation of the primal-dual method

$$\mathbf{u}^{(k+1)} = \mathrm{proj}_{\mathcal{U}^{(k)}}\left\{\mathbf{u}^{(k)} - \alpha\left(\nabla c^{(k)}(\mathbf{u}^{(k)}) + [\mathbf{J}_{\mathcal{F}}(\mathbf{u}^{(k)})]^{\mathsf{T}}[\widehat{\mathbf{J}}_g]^{\mathsf{T}}\boldsymbol{\lambda}^{(k)}\right)\right\}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathrm{proj}_{\mathbb{R}^d_+}\left\{(1-\alpha\eta r)\boldsymbol{\lambda}^{(k)} + \alpha\eta\widehat{\mathbf{g}}^{(k)}\right\}$$

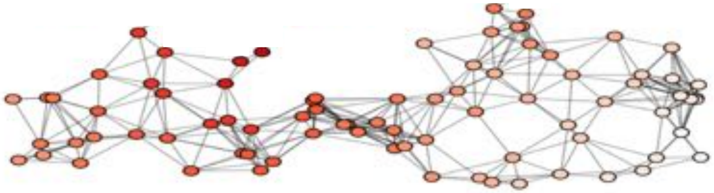
❑ Linear convergence, error bounds for linear constraints and bounded error

❑ *Ongoing work*: general nonlinear constraints

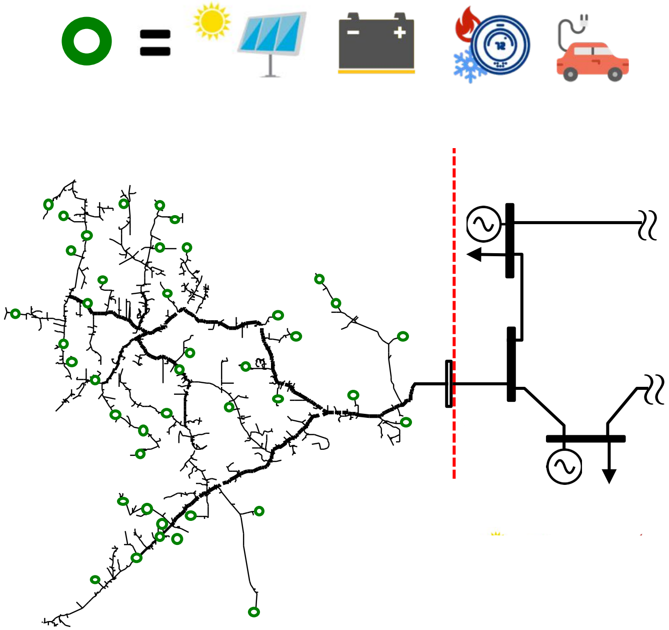
Example: Application to Power Distribution Grids

Problem setup

$$\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)$$



⟷



$\mathbf{u}_i \longleftrightarrow$ Power commands, $\mathcal{U}_i$ hardware constraints

$\mathbf{w} \longleftrightarrow$ Powers of non-controllable assets

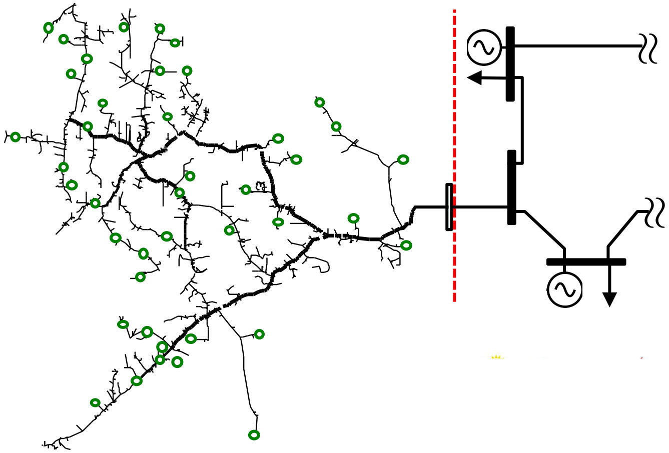
$\mathbf{y} \longleftrightarrow$ Voltage magnitudes, power flows

Problem setup

$$\min_{\mathbf{u}} \quad c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i)$$

$$\text{subject to}: \quad \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i$$

$$\mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0}$$



Problem setup

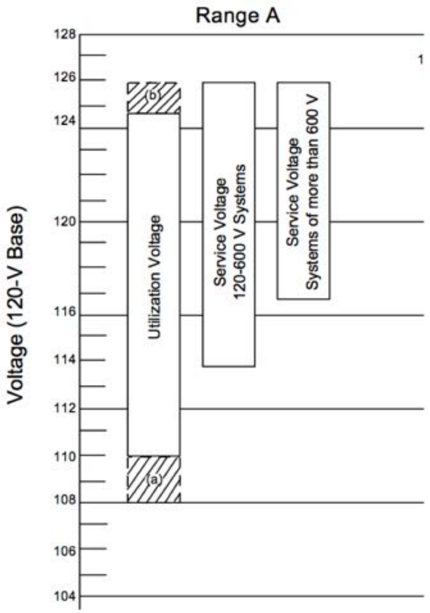
$$\min_{\mathbf{u}} \quad c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i)$$

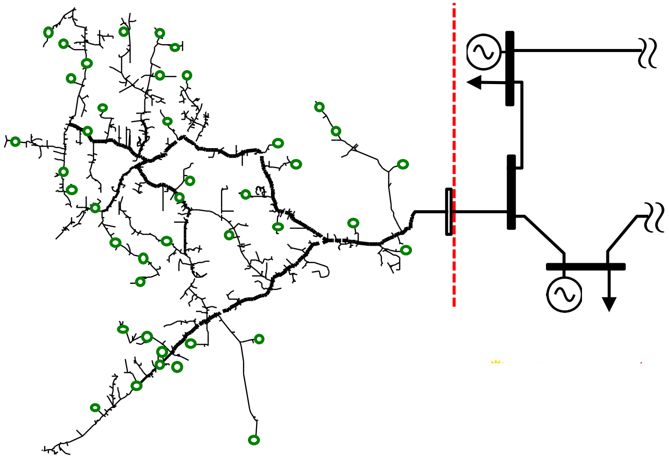
$$\text{subject to}: \quad \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i$$

$$\mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0}$$

$$v^{\min} \leq |V_n^{(k)}|$$

$$|V_n^{(k)}| \leq v^{\max}$$





Voltage Ratings ANSI C84.1

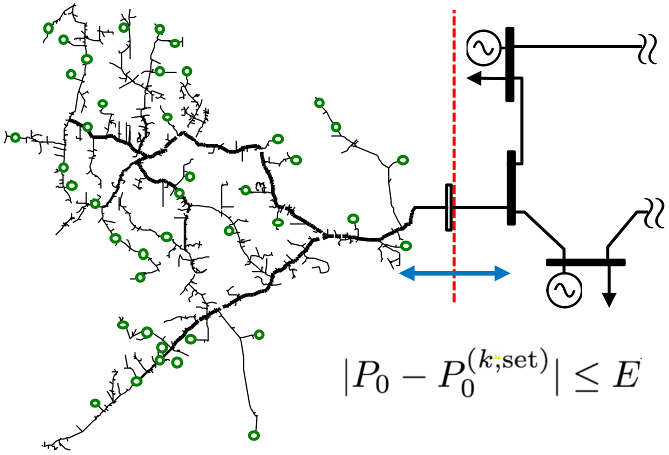
Problem setup

$$\min_{\mathbf{u}} \quad c_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i)$$

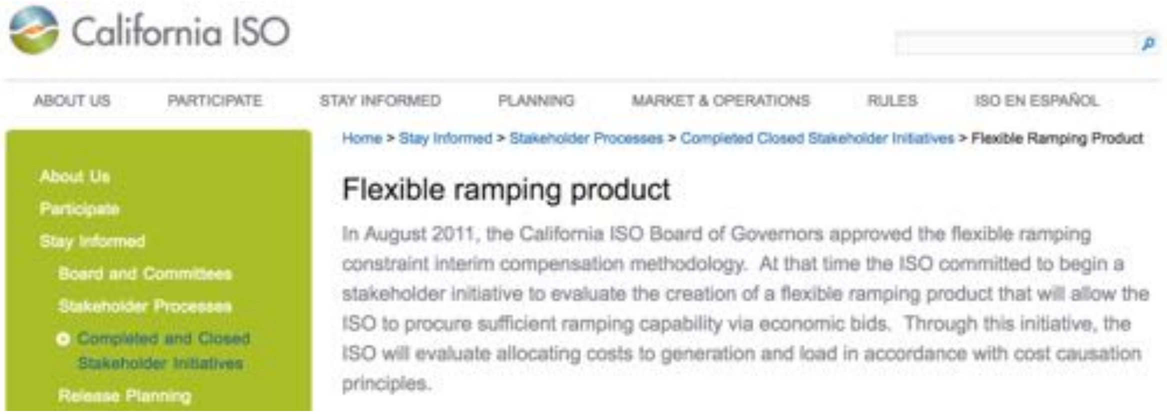
$$\text{subject to}: \quad \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall\, i$$

$$\mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq \mathbf{0}$$

$$|P_0 - P_0^{(k,\text{set})}| \leq E$$



$|P_0 - P_0^{(k,\text{set})}| \leq E$



California ISO

ABOUT US | PARTICIPATE | STAY INFORMED | PLANNING | MARKET & OPERATIONS | RULES | ISO EN ESPAÑOL

Home > Stay Informed > Stakeholder Processes > Completed Closed Stakeholder Initiatives > Flexible Ramping Product

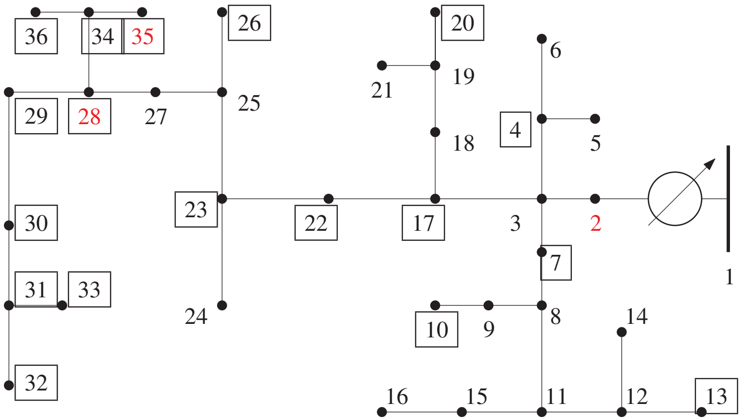
- About Us
- Participate
- Stay Informed
  - Board and Committees
  - Stakeholder Processes
  - Completed and Closed Stakeholder Initiatives
  - Release Planning

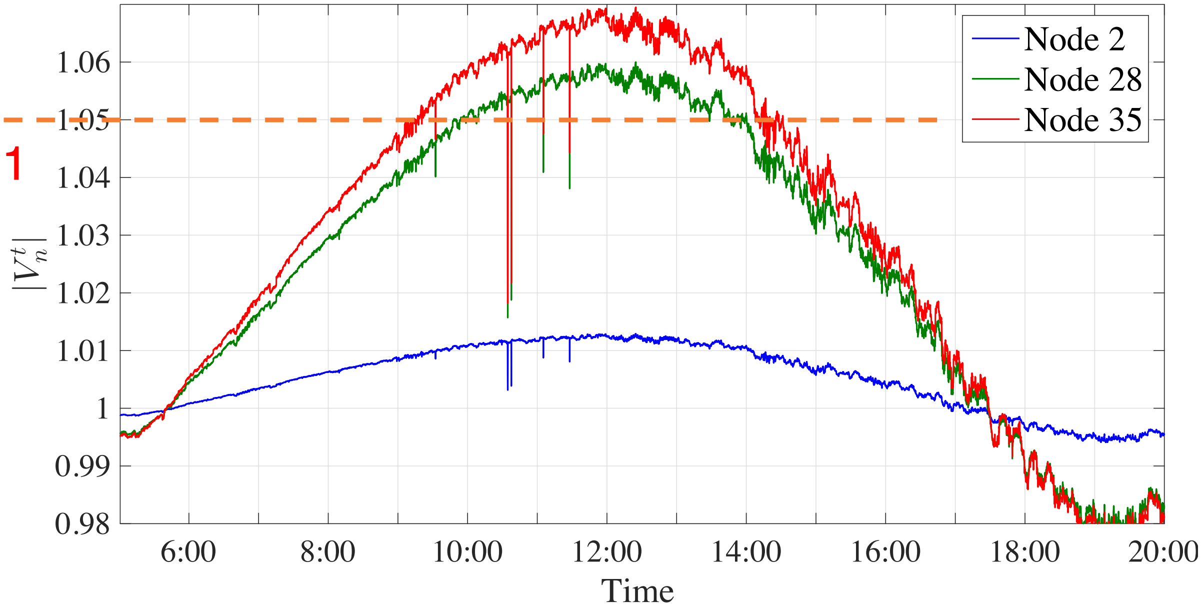
Flexible ramping product

In August 2011, the California ISO Board of Governors approved the flexible ramping constraint interim compensation methodology. At that time the ISO committed to begin a stakeholder initiative to evaluate the creation of a flexible ramping product that will allow the ISO to procure sufficient ramping capability via economic bids. Through this initiative, the ISO will evaluate allocating costs to generation and load in accordance with cost causation principles.

Representative results

- ❑ Real load and solar data from Anatolia, CA
- ❑ PQ of inverters updated every 1s
- ❑ HVAC controlled every 5 min
- ❑ Inverter mimics first-order system
- ❑ Voltage regulation and power tracking



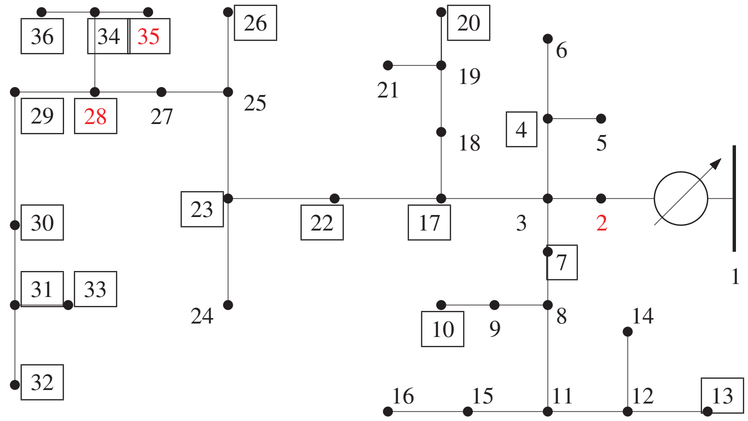


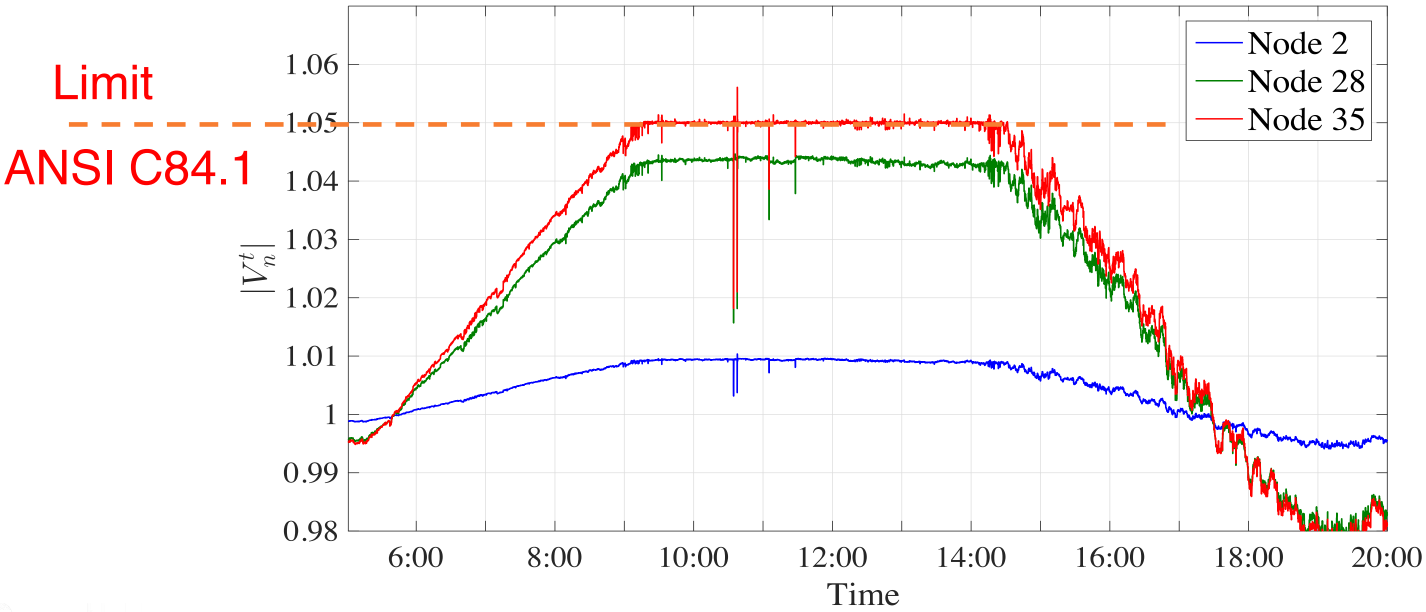
Limit

ANSI C84.1

Representative results

- Real load and solar data from Anatolia, CA
- PQ of inverters updated every 1s
- HVAC controlled every 5 min
- Inverter mimics first-order system
- Voltage regulation and power tracking



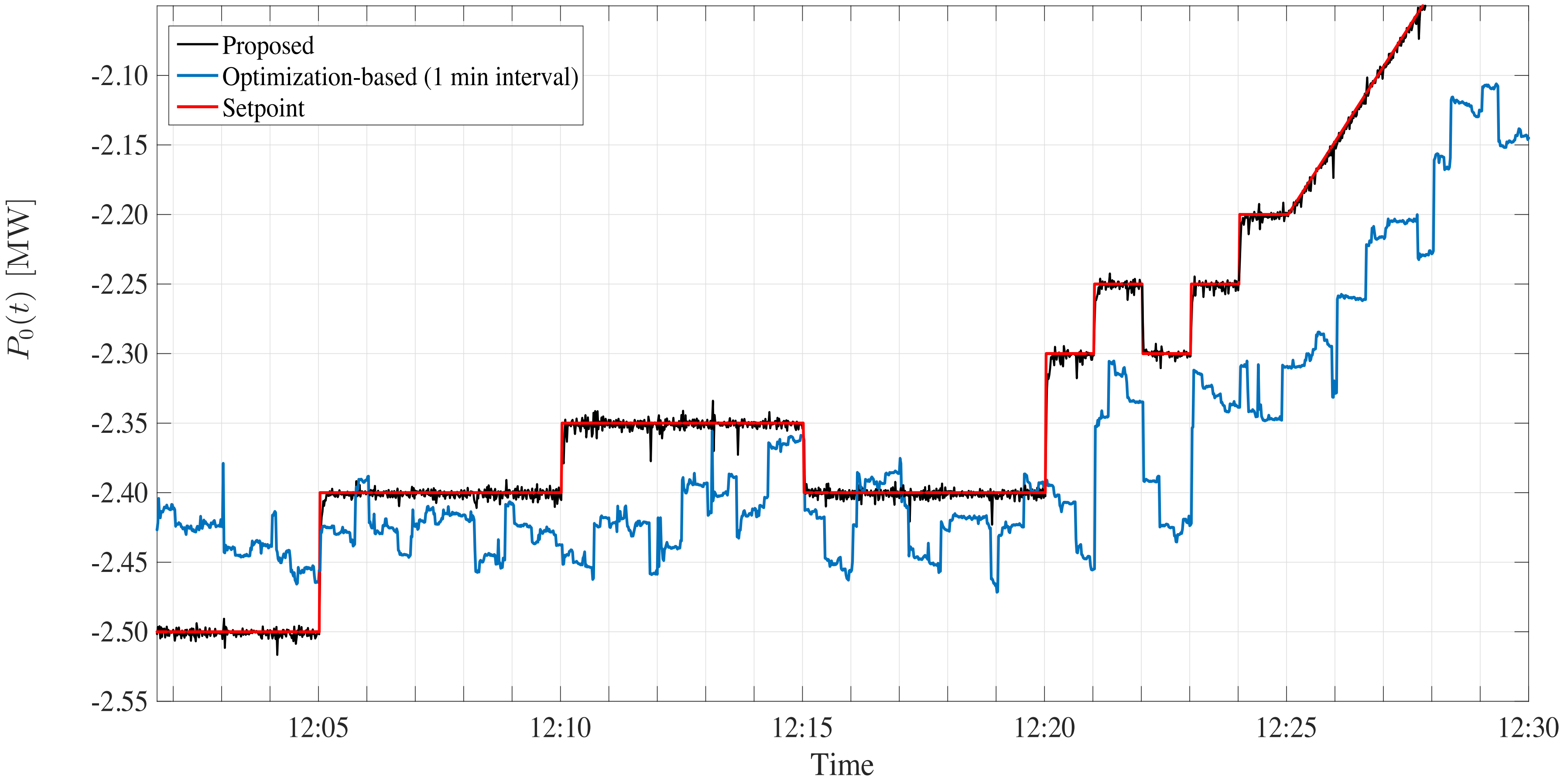


Limit

ANSI C84.1

Representative results

- Online control vs offline optimization

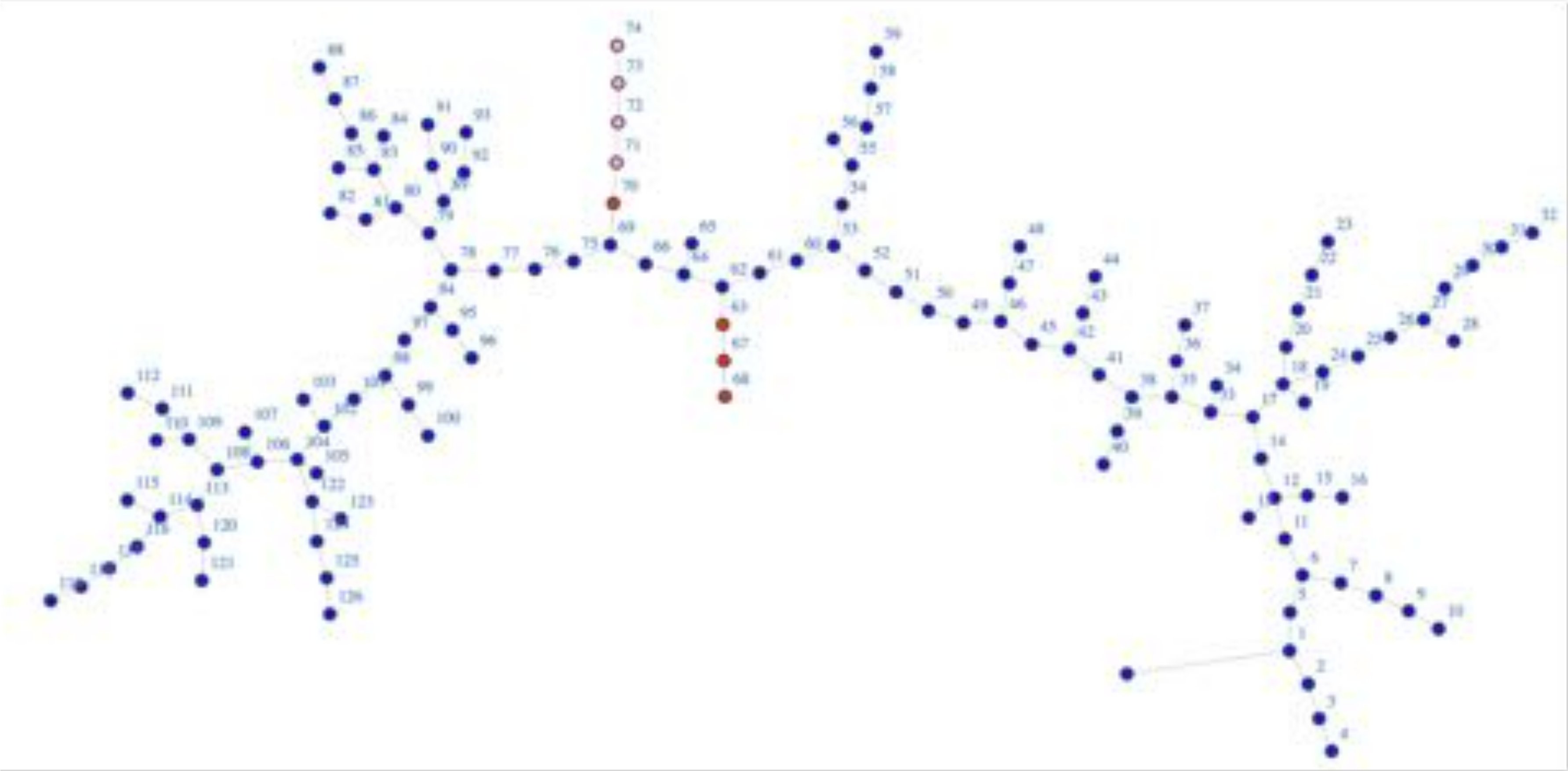


Representative results

❑ Real circuit within the Southern California Edison

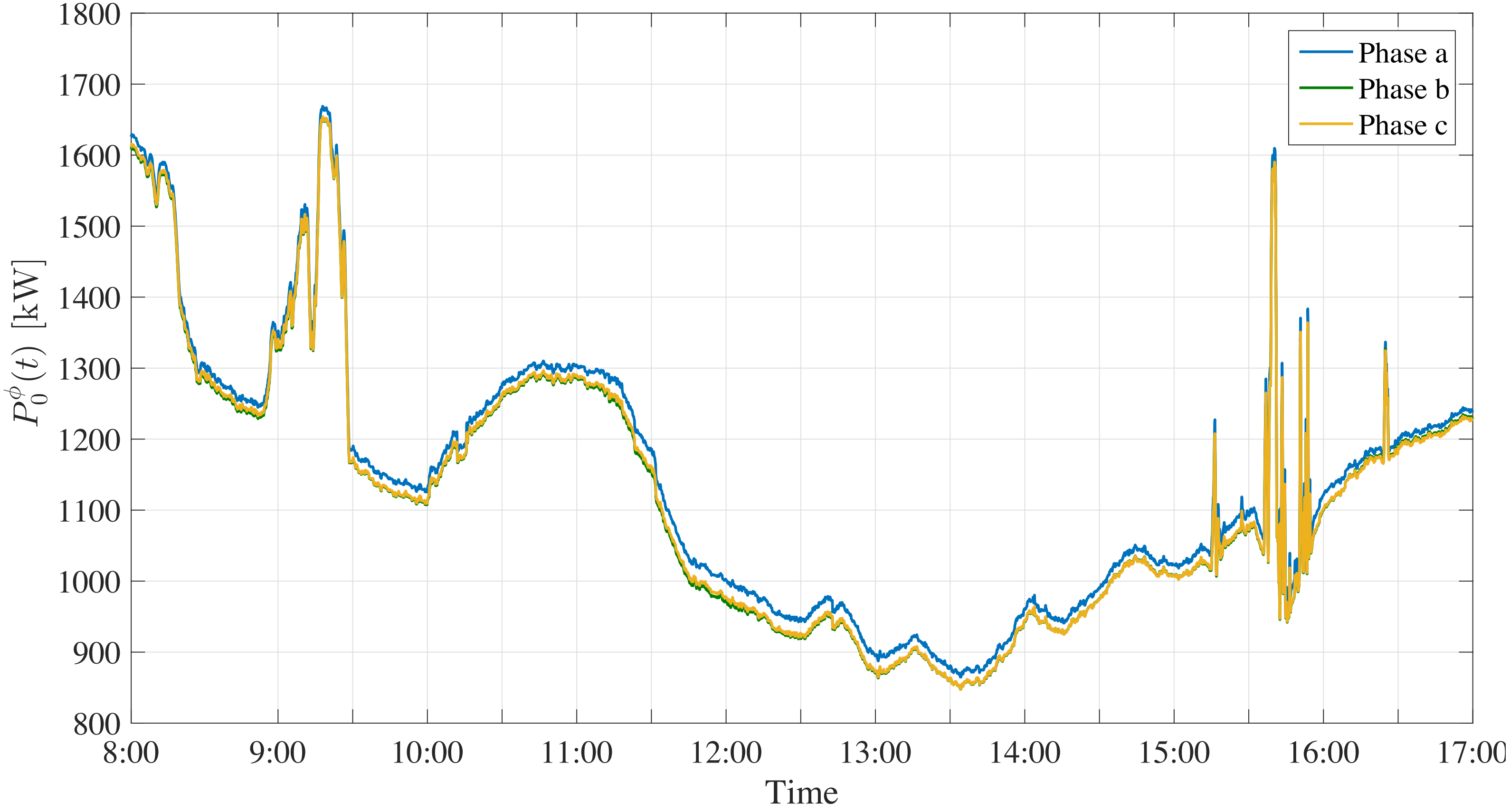
❑ PQ of inverters updated every 1s

❑ Mix of residential, commercial, and industrial customers

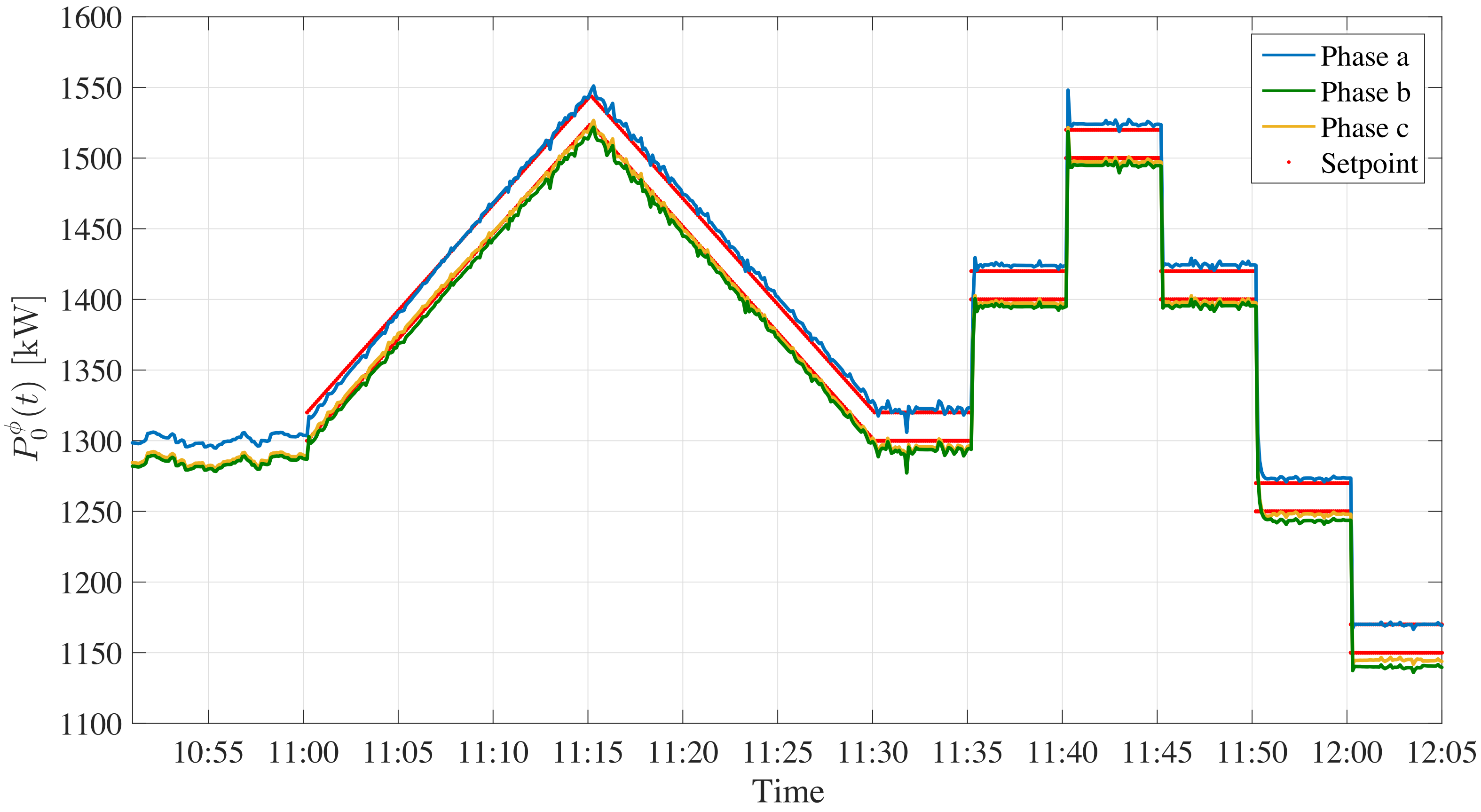


Representative results

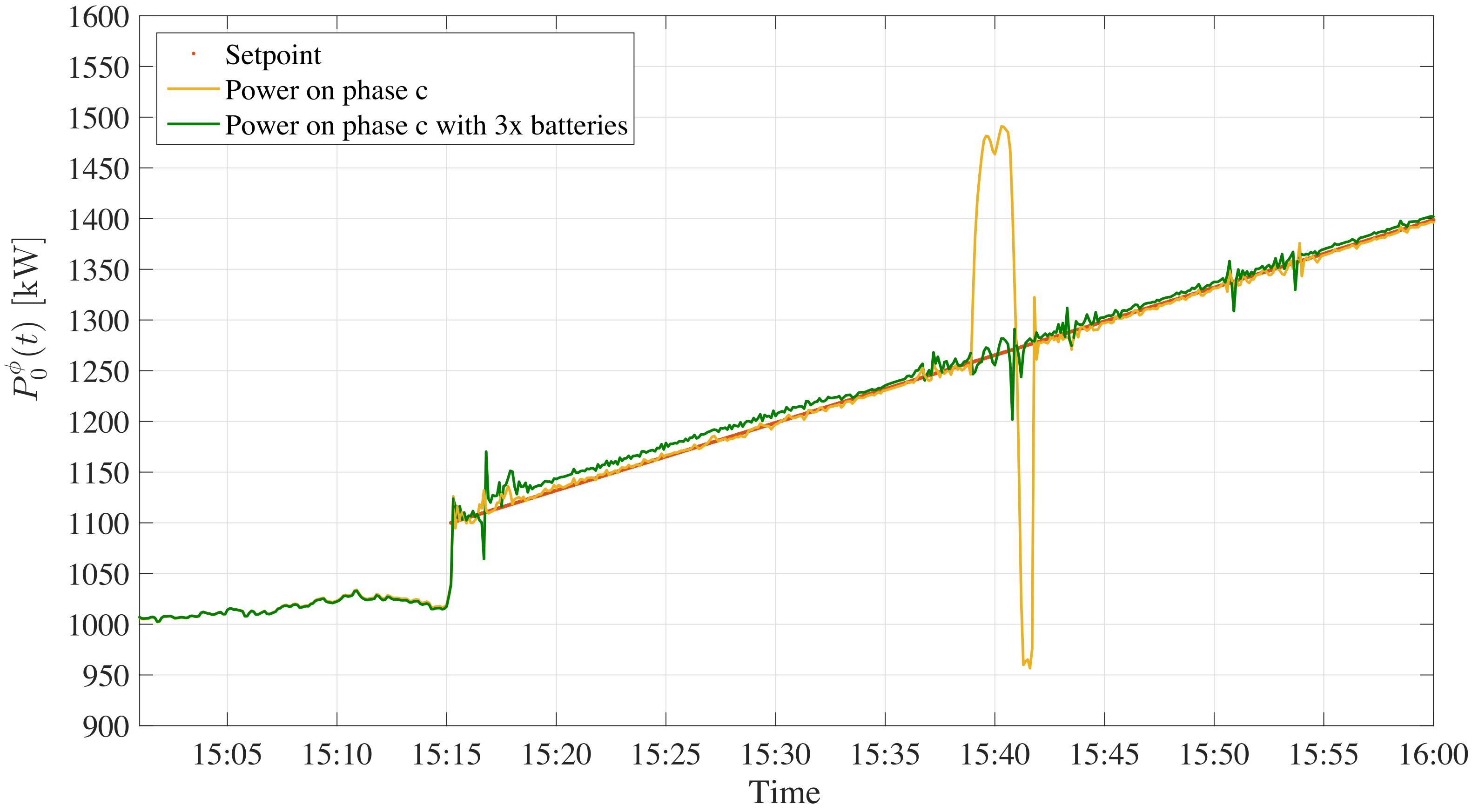
- ❑ Real circuit within the Southern California Edison
- ❑ PQ of inverters updated every 1s
- ❑ Mix of residential, commercial, and industrial customers



Representative results



Representative results



Conclusions

- ❑ *Time-varying optimization* to model optimal operational trajectories
- ❑ *Online optimization with feedback* to track
- ❑ Extended theory of saddle-flow dynamics and gradient methods
- ❑ Application for power grids to integrate DERs at scale

- ❑ Next:
  - ❑ Non-differentiable cost
  - ❑ Distributed architectures
  - ❑ Gradient-free methods

Thanks!

Backup slides

Formalizing optimal trajectories

$$\min_{\{\mathbf{u}_i\}} \ c^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_i c_i^{(k)}(\mathbf{u}_i)$$

$$\text{subject to} \ \ \mathbf{u}_i \in \mathcal{U}_i^{(k)} \qquad \forall i = 1, \dots N$$

$$g_m^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq 0 \ \forall m = 1, \dots M$$

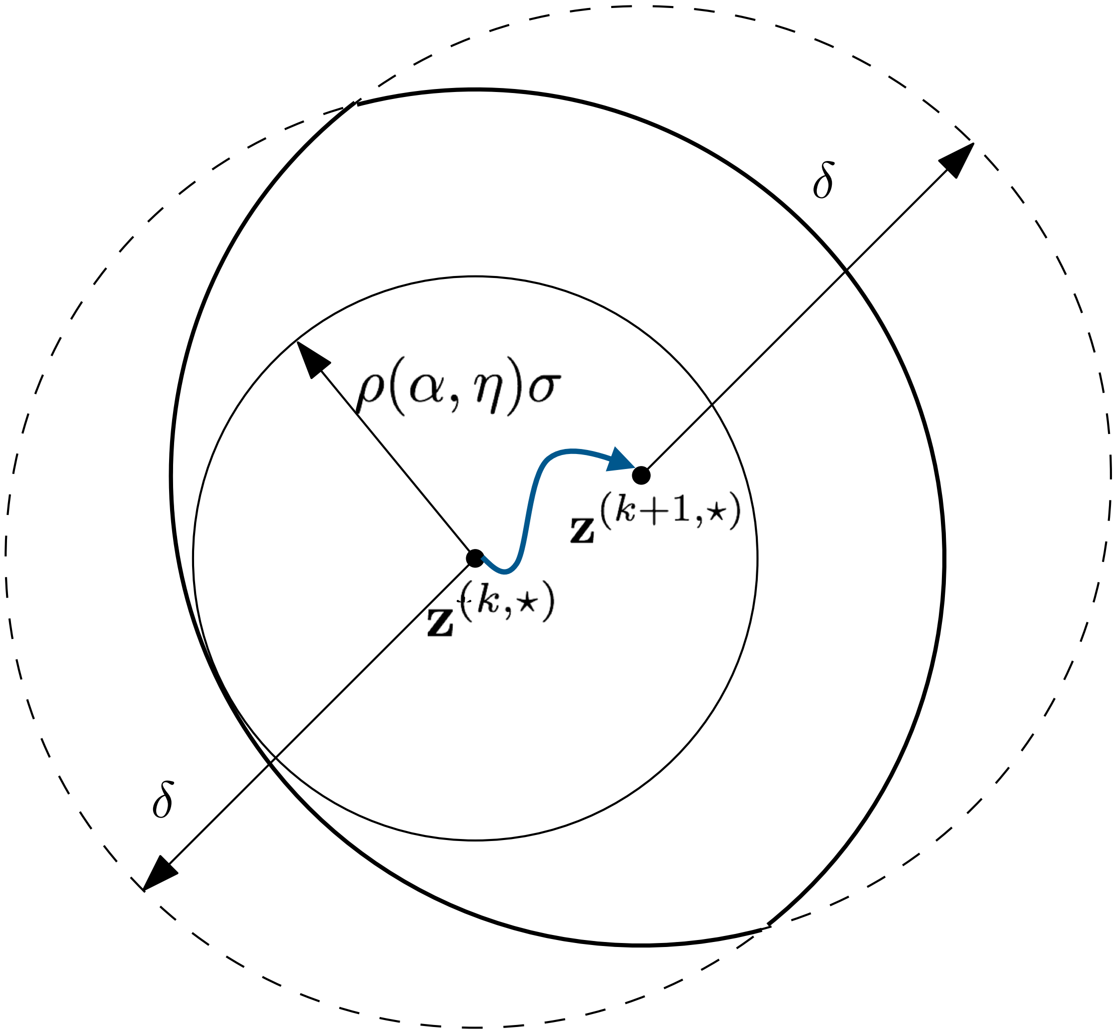
- ❑ *(As. 1)* The set $\mathcal{U}_i^{(k)}$ is convex and compact for all k.

- ❑ *(As. 2)* $c_0^{(k)}$, $\{c_i^{(k)}\}$, are convex, differentiable, with Lipschitz continuous gradient.

- ❑ *(As. 2)* $g_m^{(k)}$ is convex, differentiable, with L_{g_m}- Lipschitz continuous gradient, for all m.

- ❑ *(As. 4)* Slater's condition holds.

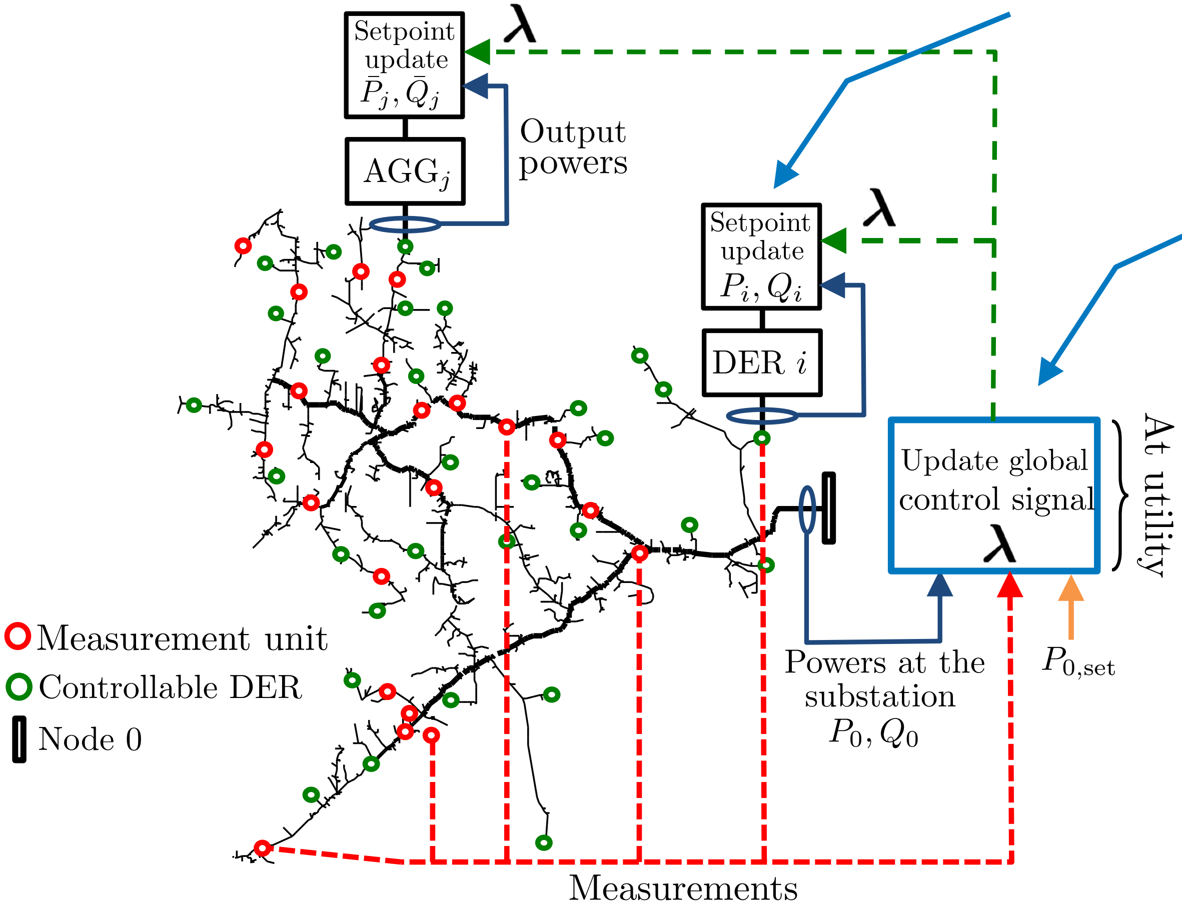
Insight on convergence



Example of distributed implementation

Setpoints computation + command

Dual update


Setpoint update $\bar{P}_j, \bar{Q}_j$
$\boldsymbol{\lambda}$
AGG$_j$
Output powers
Setpoint update P_i, Q_i
$\boldsymbol{\lambda}$
DER i
Update global control signal
$\boldsymbol{\lambda}$
At utility
Powers at the substation P_0, Q_0
$P_{0,\mathrm{set}}$
Measurement unit
Controllable DER
Node 0
Measurements