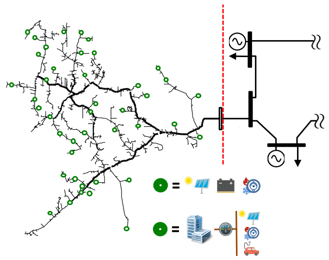
# Learning to optimize grid-edge devices in real time

Emiliano Dall'Anese

University of Colorado Boulder

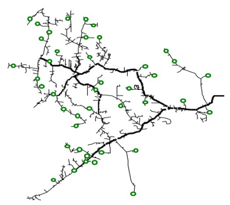
Grid Edge - Devices, Control, Applications, and System Operation

# 10,000-feet view



$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w})$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$$

# Current modus operandi



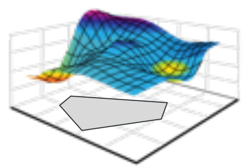
$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w})$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$$

Fast

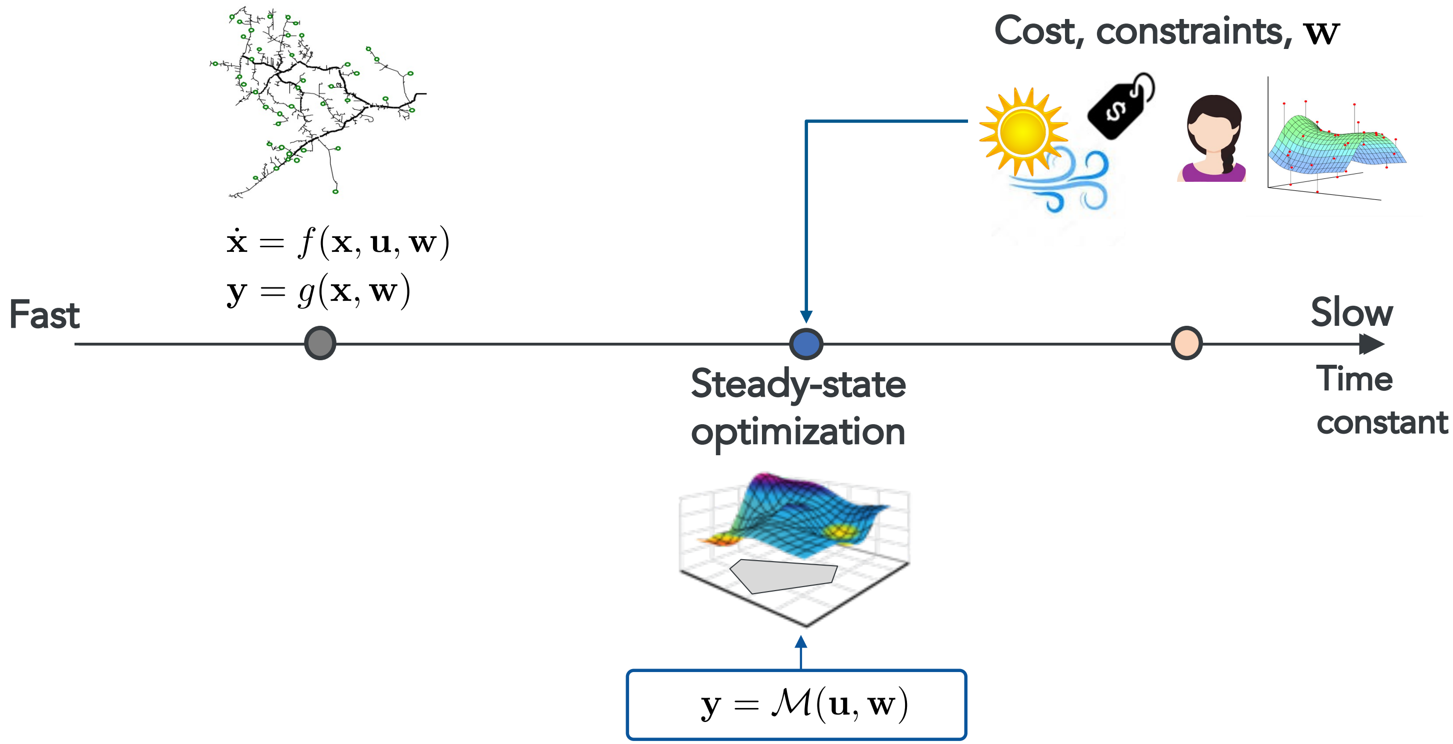
Slow

Time constant

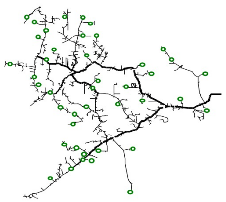
Steady-state optimization



# Current modus operandi

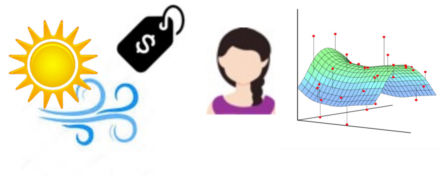


# Current modus operandi



$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}^*, \mathbf{w})$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$$

Cost, constraints, $\mathbf{w}$



Fast

Slow

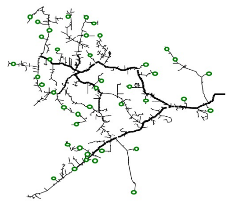
Time constant

Steady-state optimization

repeat
$$\mathbf{u}_{k+1} = \mathcal{T}(\mathbf{u}_k, \mathbf{w})$$
until convergence

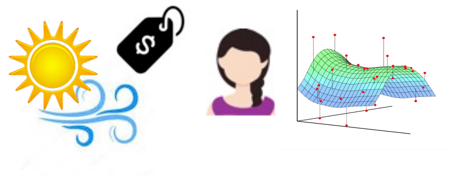
$$\mathbf{y} = \mathcal{M}(\mathbf{u}, \mathbf{w})$$

# Current modus operandi



$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}^*, \mathbf{w})$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$$

Cost, constraints, $\mathbf{w}$



Fast

Slow

Time constant

Steady-state optimization

$\mathbf{u}^\star$

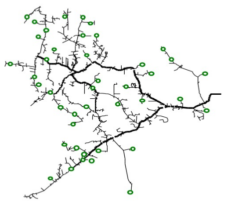
repeat
$$\mathbf{u}_{k+1} = \mathcal{T}(\mathbf{u}_k, \mathbf{w})$$
until convergence

$$\mathbf{y} = \mathcal{M}(\mathbf{u}, \mathbf{w})$$

IEEE PES Power & Energy Society®

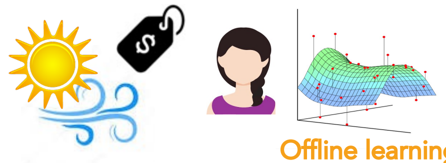
IEEE

# Implicit assumptions



$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}^*, \mathbf{w})$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$$

Cost, constraints, $\mathbf{w}$



Offline learning

Fast

$\infty$-fast

Computation + sensing

Slow

Time constant

Steady-state optimization

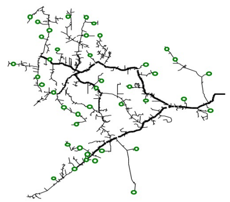
repeat
$\mathbf{u}_{k+1} = \mathcal{T}(\mathbf{u}_k, \mathbf{w})$
until convergence

$\mathbf{y} = \mathcal{M}(\mathbf{u}, \mathbf{w})$

$\mathbf{w}$ known

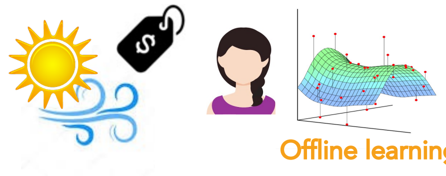
cost is known

# Implicit assumptions



$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}^*, \mathbf{w})$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$$

Cost, constraints, $\mathbf{w}$



Offline learning

Fast

Slow

Time constant

Steady-state optimization

repeat
$\quad \mathbf{u}_{k+1} = \mathcal{T}(\mathbf{u}_k, \mathbf{w})$
until convergence

$$\mathbf{y} = \mathcal{M}(\mathbf{u}, \mathbf{w})$$

Plant $\infty$-fast

Computation

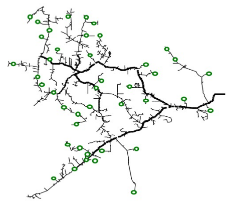
$\mathbf{w}$ known

Cost is known

IEEE PES Power & Energy Society®

IEEE

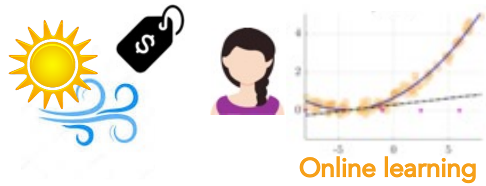
# Stochastic and uncertain operation



$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}^*, \mathbf{w})$

$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$

Cost, constraints, $\mathbf{w}$




Fast

Slow

Steady-state optimization

Time constant

repeat
$\mathbf{u}_{k+1} = \mathcal{T}(\mathbf{u}_k, \mathbf{w})$
until convergence

$\mathbf{y} = \mathcal{M}(\mathbf{u}, \mathbf{w})$

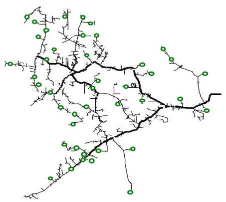
Plant $\infty$-fast

~~Computation~~

~~$\mathbf{w}$ known~~

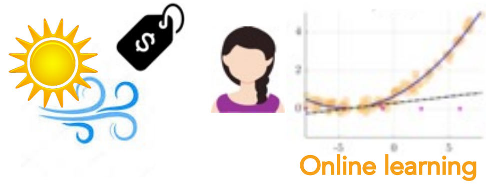
~~Cost is known~~

# Objective



$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w})$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$$

Cost, constraints, $\mathbf{w}$




Fast

Slow

Steady-state optimization

Time constant

online algorithm

$$\mathbf{u}_{t+1} = \hat{\mathcal{T}}_t(\mathbf{u}_t, \mathbf{y}_t)$$

$$\mathbf{y}_t = \mathcal{M}(\mathbf{u}_t, \mathbf{w}_t)$$

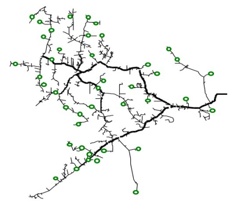
Plant $\infty$-fast

~~Computation~~

~~$\mathbf{w}$ known~~

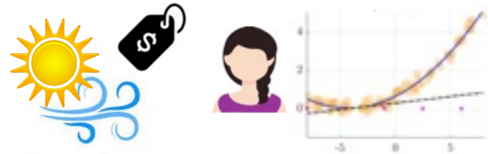
~~Cost is known~~

# Objective



$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w})$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$$

Cost, constraints, $\mathbf{w}$



Online learning

Fast — Slow

Time constant

Steady-state optimization

online algorithm
$$\mathbf{u}_{t+1} = \hat{\mathcal{T}}_t(\mathbf{u}_t, \mathbf{y}_t)$$

$$\mathbf{y}_t = \mathcal{M}(\mathbf{u}_t, \mathbf{w}_t)$$

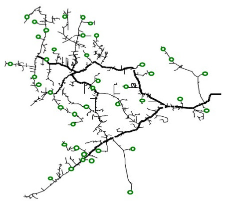
Plant $\infty$-fast

~~Computation~~

~~$\mathbf{w}$ known~~

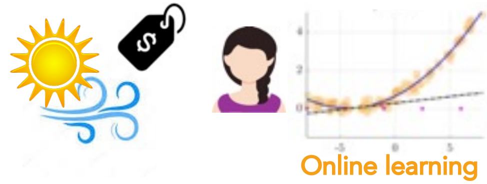
- Algorithmic synthesis
- Rigorous optimality convergence analysis

# Objective



$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w})$$
$$\mathbf{y} = g(\mathbf{x}, \mathbf{w})$$

Cost, constraints, $\mathbf{w}$




Fast — Slow

Steady-state optimization

Time constant

online algorithm
$$\mathbf{u}_{t+1} = \hat{\mathcal{T}}_t(\mathbf{u}_t, \mathbf{y}_t)$$

$$\mathbf{y}_t = \mathcal{M}(\mathbf{u}_t, \mathbf{w}_t)$$

- ❑ Learn users' preferences

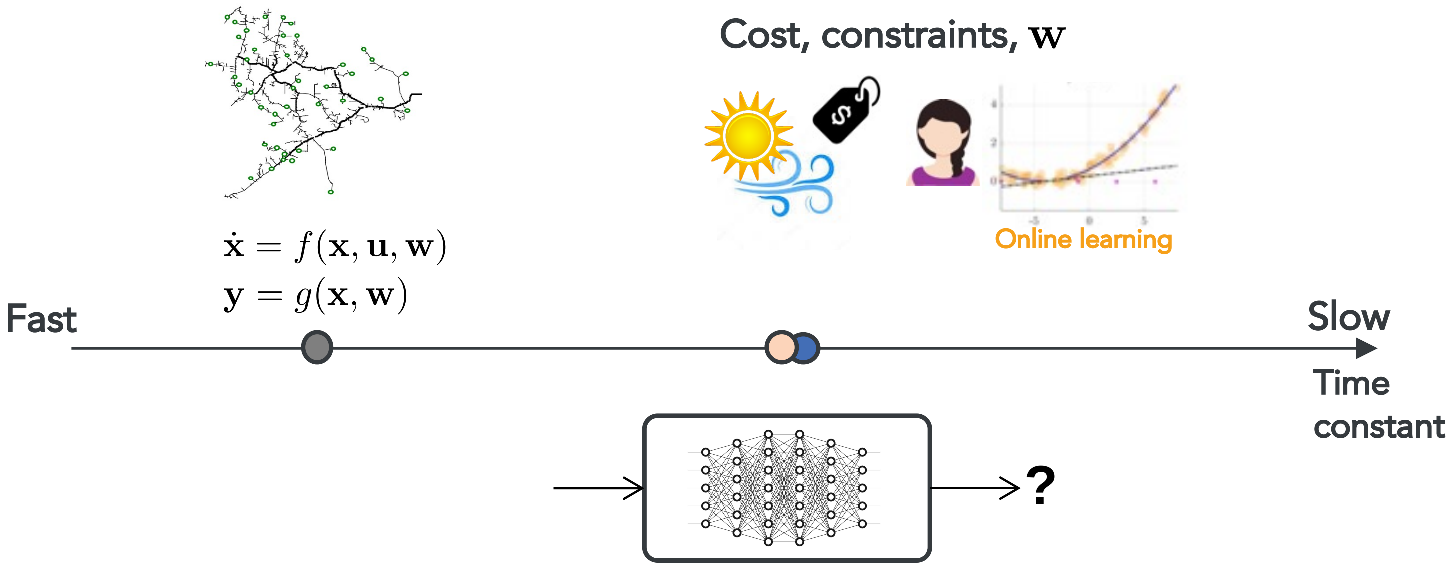
Plant $\infty$-fast

~~Computation~~

~~$\mathbf{w}$ known~~

~~Cost is known~~

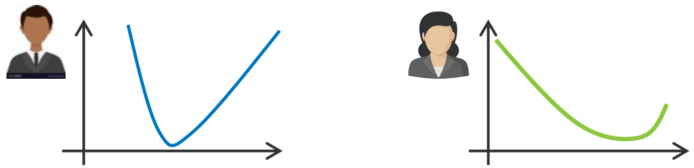
# Aside: not a neural network approach




- Not a “deep learning” approach
  - Require information about non-controllable assets
  - Require massive dataset for training
  - We rely on parsimonious feedback
  - We aim to provide rigorous optimality and feasibility guarantees

# Applications: demand response

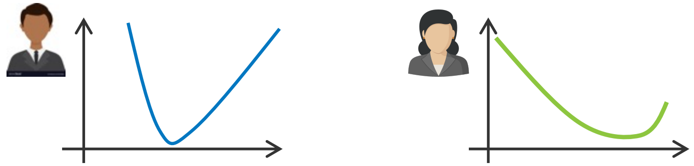
Users' or DERs' function: $J_t(\mathbf{u}) = \sum_{m=1}^{N} J_{m,t}(u_m)$



- HVACs, EVs, energy storage systems, PV systems

# Applications: demand response

Users' or DERs' function: $J_t(\mathbf{u}) = \sum_{m=1}^{N} J_{m,t}(u_m)$



- HVACs, EVs, energy storage systems, PV systems

Total power consumption: $y_t = \sum_m u_m + \mathbf{1}^\top \mathbf{w}_t$

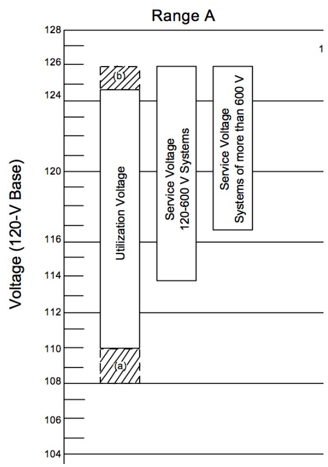
- Task: follow an AGC or DR signal

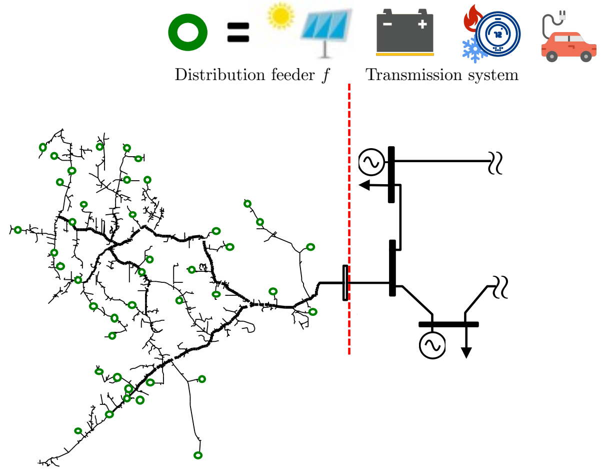
$$C_t(\mathbf{u}) = \frac{a}{2}(y - y_t^{ref})^2 \quad \text{or} \quad \frac{a}{2}(y - y_t^{ref})^2 \le \epsilon$$

# Applications: optimal power flow

$$\min_{\mathbf{u}\in\mathcal{U}_t} \; J_t(\mathbf{u}) + C_t(\mathcal{M}(\mathbf{u},\mathbf{w}_t))$$
$$\text{s. to } \; \mathbf{g}_t(\mathcal{M}(\mathbf{u},\mathbf{w}_t)) \leq \mathbf{0}$$

$$v^{min} \leq |V_{n,t}|^2$$
$$|V_{n,t}|^2 \leq v^{max}$$



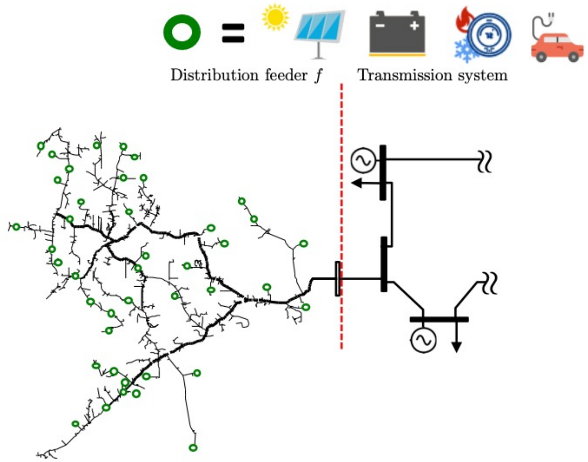



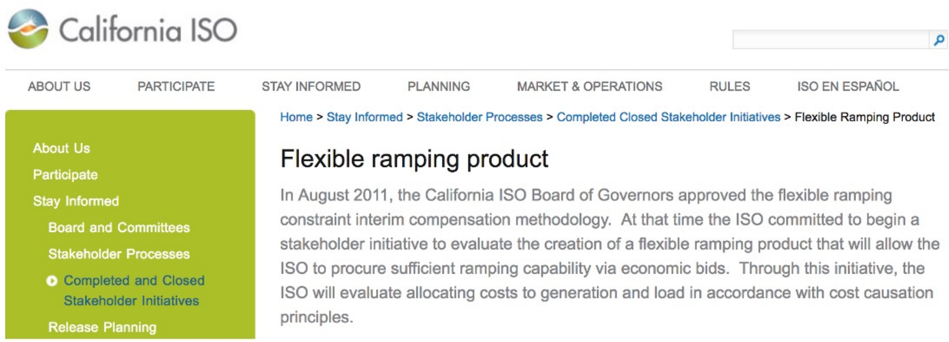
# Applications: optimal power flow

$$\min_{\mathbf{u} \in \mathcal{U}_t} \; J_t(\mathbf{u}) + C_t(\mathcal{M}(\mathbf{u}, \mathbf{w}_t))$$

$$\text{s. to } \; \mathbf{g}_t(\mathcal{M}(\mathbf{u}, \mathbf{w}_t)) \leq \mathbf{0}$$

$$|P_0 - P_{0,t}^{set}| \leq e$$



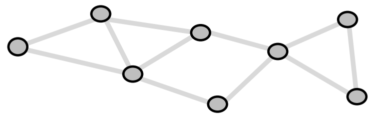
# Model

- Time-varying optimization [Popkov'05]

$$t \in \mathcal{T} := \{k\Delta, k \in \mathbb{N}\}$$

$$\min_{\mathbf{u} \in \mathcal{U}_t} \; J_t(\mathbf{u}) + C_t(\mathcal{M}(\mathbf{u}, \mathbf{w}_t))$$

$$\text{s. to} \;\; \mathbf{g}_t(\mathcal{M}(\mathbf{u}, \mathbf{w}_t)) \leq \mathbf{0}$$



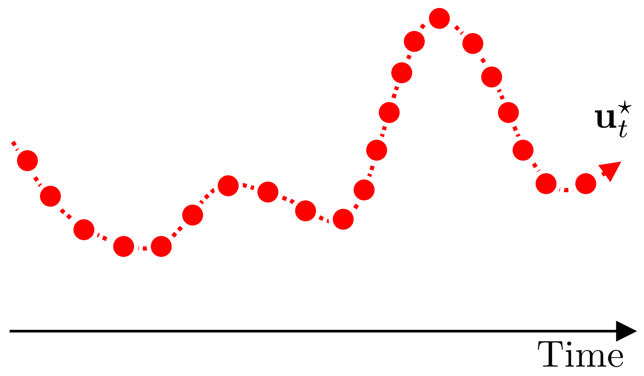
$$\mathbf{y}_t = \mathcal{M}(\mathbf{u}_t, \mathbf{w}_t)$$

# Model

- Time-varying optimization [Popkov'05]

  $t \in \mathcal{T} := \{k\Delta, k \in \mathbb{N}\}$

$$\min_{\mathbf{u} \in \mathcal{U}_t} \; J_t(\mathbf{u}) + C_t(\mathcal{M}(\mathbf{u}, \mathbf{w}_t))$$
$$\text{s. to } \; \mathbf{g}_t(\mathcal{M}(\mathbf{u}, \mathbf{w}_t)) \leq \mathbf{0}$$



# Problem statement revisited

- Time-varying optimization [Popkov'05]

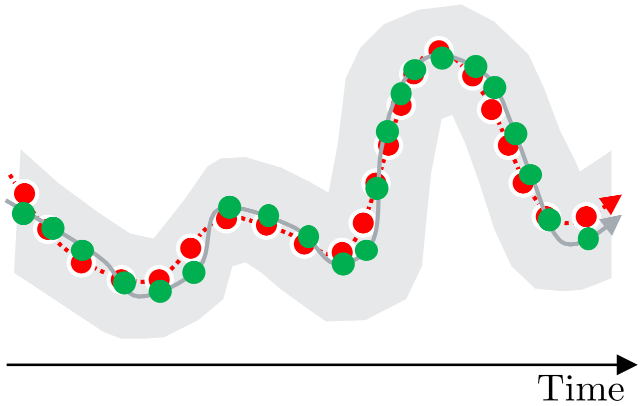
$$t \in \mathcal{T} := \{k\Delta, k \in \mathbb{N}\}$$

$$\min_{\mathbf{u}\in\mathcal{U}_t} \; J_t(\mathbf{u}) + C_t(\mathcal{M}(\mathbf{u}, \mathbf{w}_t))$$
$$\text{s. to} \;\; \mathbf{g}_t(\mathcal{M}(\mathbf{u}, \mathbf{w}_t)) \leq \mathbf{0}$$

- Problem: design an online algorithm

$$\mathbf{u}_{t+1} = \hat{\mathcal{T}}_t(\mathbf{u}_t, \mathbf{y}_t)$$

  - Tracks $\{\mathbf{u}_t^\star, t \in \mathcal{T}\}$
  - Does not require $\{\mathbf{w}_t, t \in \mathcal{T}\}$
  - May rely on approximate costs
    - Learning human preferences [Ospina-Simonetto-Dall'Anese'20], [Menner et al'20]
    - Learning models or barrier functions [Taylor et al'20], [Lindemann et al'20]



# Example of feedback-based optimization

$$\min_{\mathbf{u}\in\mathcal{U}_t} \; J_t(\mathbf{u}) + C_t(\mathbf{G}\mathbf{u} + \mathbf{H}\mathbf{w}_t)$$

IEEE PES Power & Energy Society®

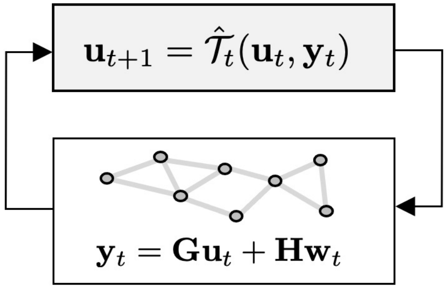
IEEE

# Example of feedback-based optimization

$$\min_{\mathbf{u}\in\mathcal{U}_t} \; J_t(\mathbf{u}) + C_t(\mathbf{G}\mathbf{u} + \mathbf{H}\mathbf{w}_t)$$

- Feedback-based online PGD:

$$\mathbf{u}_{t+1} = \hat{\mathcal{T}}_t(\mathbf{u}_t, \mathbf{y}_t) = \mathrm{proj}_{\mathcal{U}_t}\left\{\mathbf{u}_t - \alpha(\hat{\nabla} J_t + \mathbf{G}^\top \nabla C_t(\mathbf{y}_t))\right\}$$

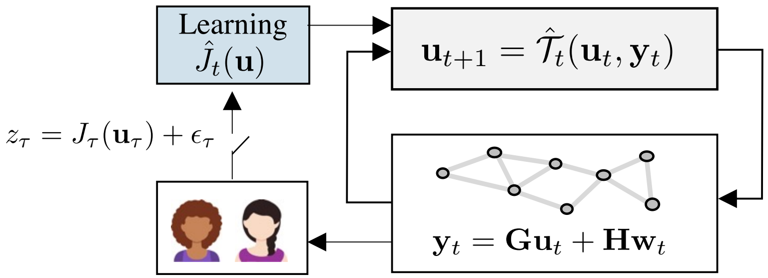


# Example of feedback-based optimization

$$\min_{\mathbf{u}\in\mathcal{U}_t} \; J_t(\mathbf{u}) + C_t(\mathbf{G}\mathbf{u} + \mathbf{H}\mathbf{w}_t)$$

- Feedback-based online PGD:

$$\mathbf{u}_{t+1} = \hat{\mathcal{T}}_t(\mathbf{u}_t, \mathbf{y}_t) = \text{proj}_{\mathcal{U}_t}\left\{\mathbf{u}_t - \alpha(\hat{\nabla} J_t + \mathbf{G}^\top \nabla C_t(\mathbf{y}_t))\right\}$$




- How to obtain $\hat{\nabla} J_t$ or $\hat{J}_t(\mathbf{u})$?
  - Zeroth-order methods [Duchi et al'15], [Hajinezhad-Hong-Garcia'19], [Tang-Ren-Li'20] …
  - Online learning via Gaussian Processes [Amani-Alizadeh-Thrampoulidis'20], [Ospina-Simonetto-Dall'Anese'20], [Srinivas et al'12], …
  - Strongly convex regression [Simonetto'20]

IEEE PES Power & Energy Society®

IEEE

# Example of feedback-based optimization

- Learn function from *functional evaluations* $z_\tau = J_\tau(\mathbf{u}_\tau) + \epsilon_\tau$
- The posterior distribution of $J(\mathbf{u})|\{\mathbf{u}_t, \mathbf{z}_t\}$ is a GP with mean and variance:

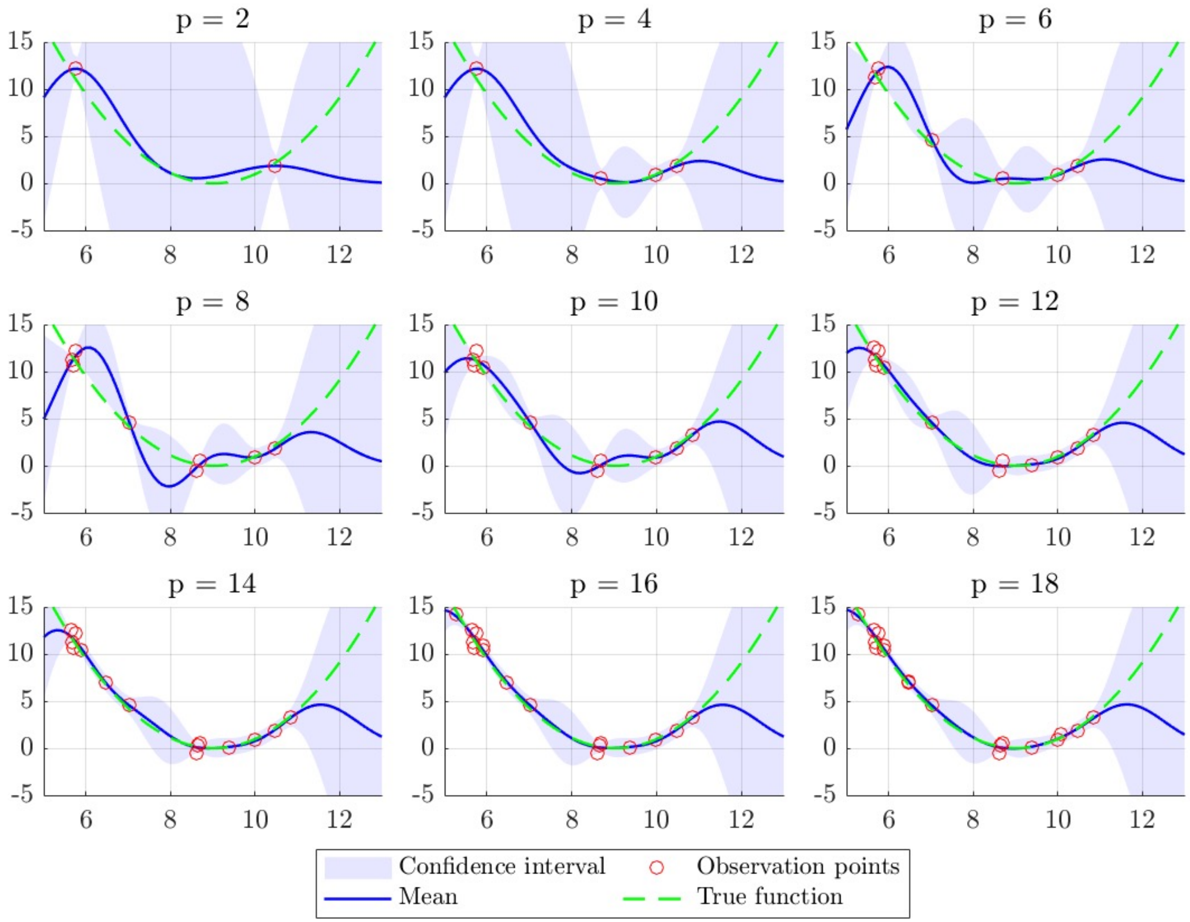
$$
\begin{aligned}
\mu_t(\mathbf{u}) &= \mathbf{k}(\mathbf{u})^\top (\mathbf{K} + \gamma^2 \mathbf{I})^{-1} \mathbf{z}_t \\
k_t(\mathbf{u}, \mathbf{u}') &= k(\mathbf{u}, \mathbf{u}') - \mathbf{k}(\mathbf{u})^\top (\mathbf{K} + \gamma^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{u}') \\
\varsigma_t^2(\mathbf{u}) &= k(\mathbf{u}, \mathbf{u})
\end{aligned}
$$

- Surrogate function:

$$\hat{J}_t(\mathbf{u}) = \mu_t(\mathbf{u})$$ [exploitation]

$$\hat{J}_t(\mathbf{u}) = \mu_t(\mathbf{u}) + \beta_t \varsigma_t(\mathbf{u})$$ [exploitation-exploration]

# Example of feedback-based optimization



# Example of convergence results

$$\min_{\mathbf{u}\in\mathcal{U}_t} \; J_t(\mathbf{u}) + C_t(\mathbf{G}\mathbf{u} + \mathbf{H}\mathbf{w}_t)$$

**Theorem.** Let $\mathbf{u} \mapsto F_t(\mathbf{u})$ be strongly convex and let $\alpha \in (0, 2/L)$. Suppose that $\{\mathbf{u}_t, t \in \mathcal{T}\}$ is generated by the online algorithm $\mathbf{u}_{t+1} = \hat{\mathcal{T}}_t(\mathbf{u}_t, \mathbf{y}_t)$ . Then:

*(i)* $$\mathbb{E}[\|\mathbf{u}_{t+1} - \mathbf{u}^*_{t+1}\|] \leq \zeta_t \|\mathbf{u}_0 - \mathbf{u}^*_0\| + \sum_{i=0}^{t} \beta_i(\alpha\mathbb{E}[\|\mathbf{e}_i\|] + \sigma_i)$$

where

$$\zeta_t := \prod_{i=1}^{t} \rho_i \;, \qquad \beta_i = \begin{cases} 1 & \text{if } i = t, \\ \prod_{k=i+1}^{t} \rho_k & \text{if } i \neq t. \end{cases}$$

*(ii)* With probability $(1-\delta), \delta \in (0,1)$ , we have that:

$$\|\mathbf{u}_{t+1} - \mathbf{u}^*_{t+1}\| \leq \left(\frac{2e}{\theta}\right)^{\theta} \log\left(\frac{2}{\delta}\right)^{\theta} \left(\zeta_t\|\mathbf{u}_0 - \mathbf{u}^*_0\| + \sum_{i=0}^{t}\beta_i(\alpha\nu_i + \sigma_i)\right) .$$

# From theory to the field

## Basalt Vista Affordable Housing Project

- Habitat for Humanity, Pitkin County, Basalt School District
- 27 homes for teachers and local workforces.
- Designed to ZNE building with *all electric* construction
- Adjacent to Basalt High School
- 4 selected for HCE's field deployment

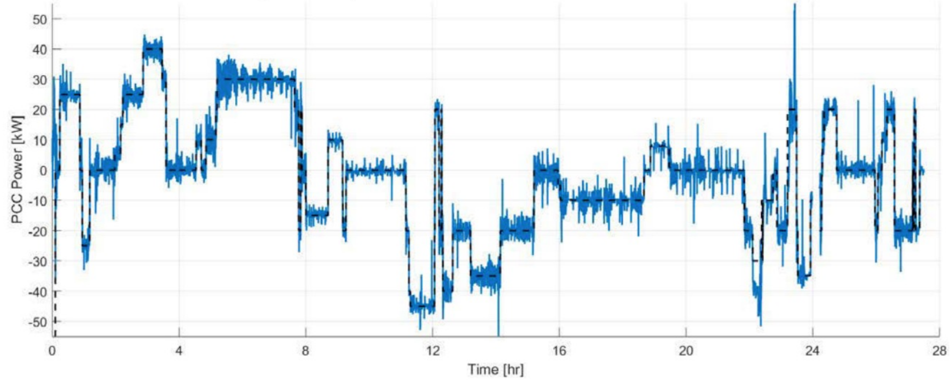
**Home Equipped with Controllable Loads**

- Rooftop solar
- Energy storage
- Mobility charging (EVSE)
- Comfort (Hot Water + HVAC)

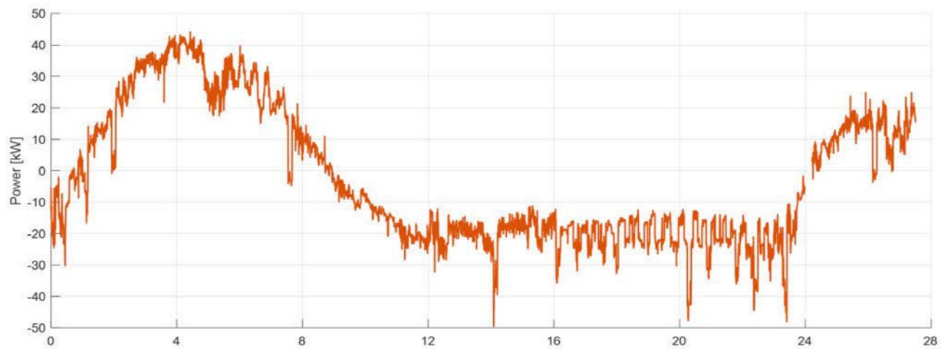
Effort part of ARPA-e NODES project

# From theory to the field

- 24-hour point of common coupling power flow tracking:



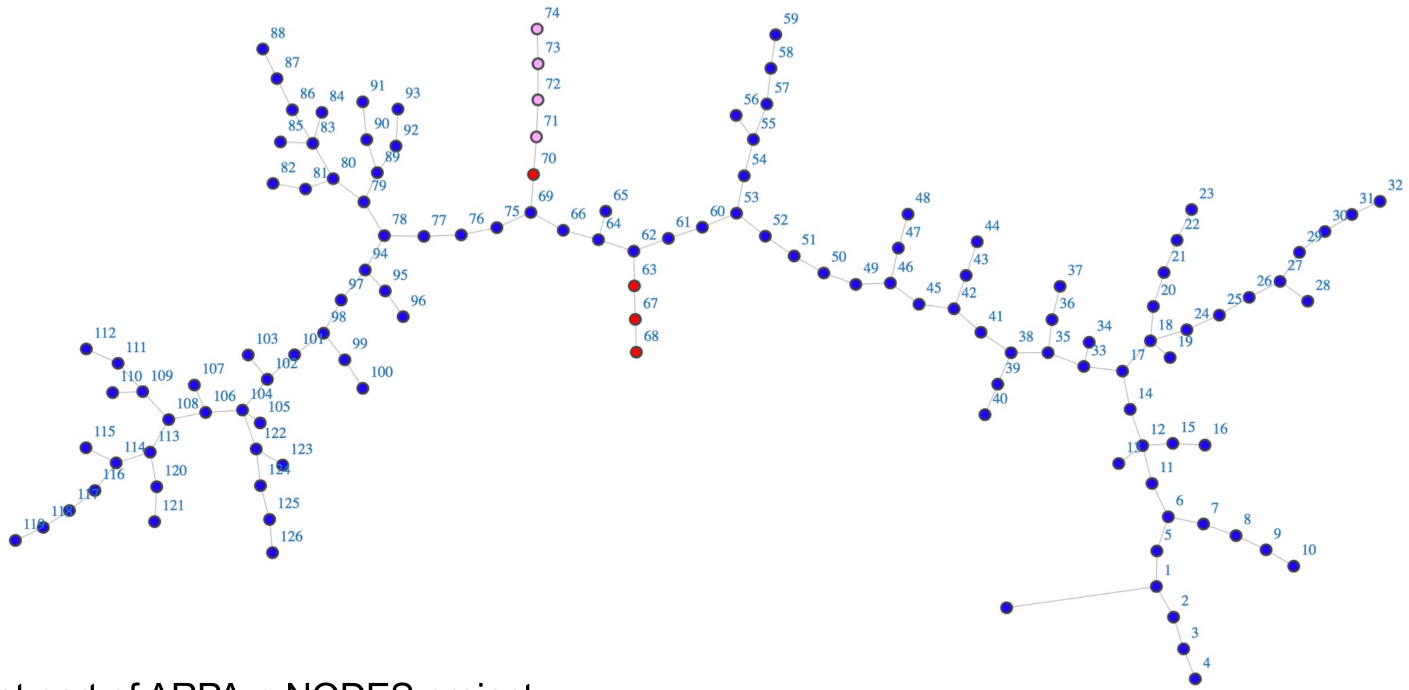

- 24-hour point of common coupling power flow without control:




Effort part of ARPA-e NODES project

# Example: real-time feedback-based OPF

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



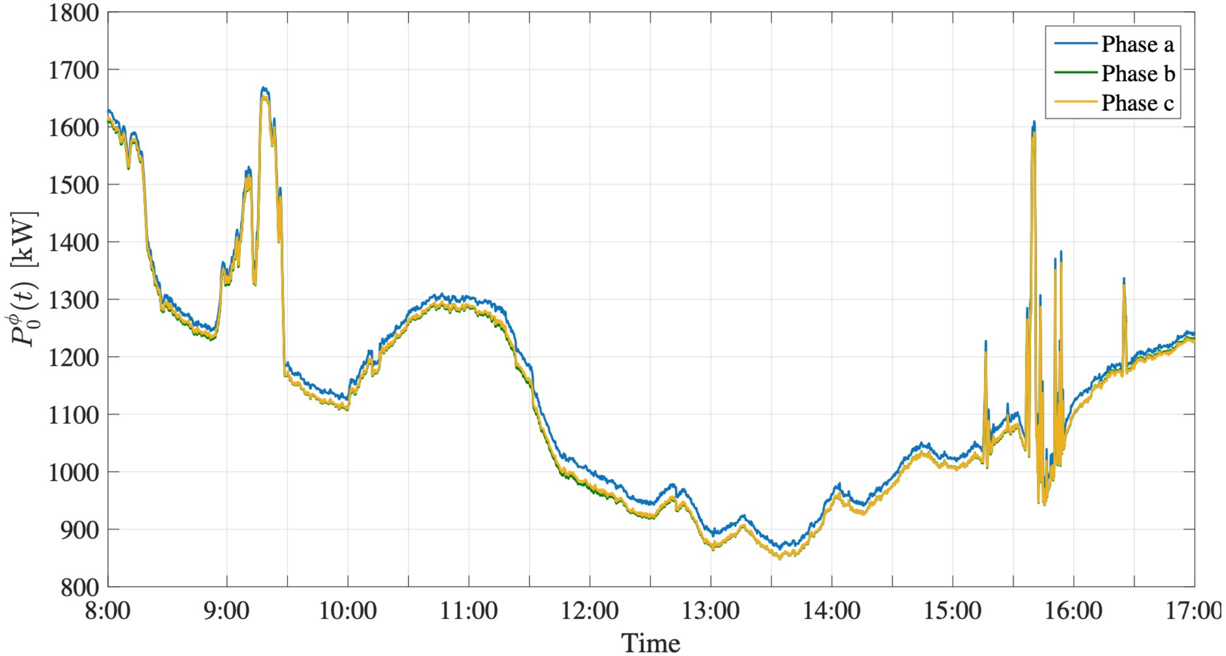
Test part of ARPA-e NODES project

IEEE PES Power & Energy Society®

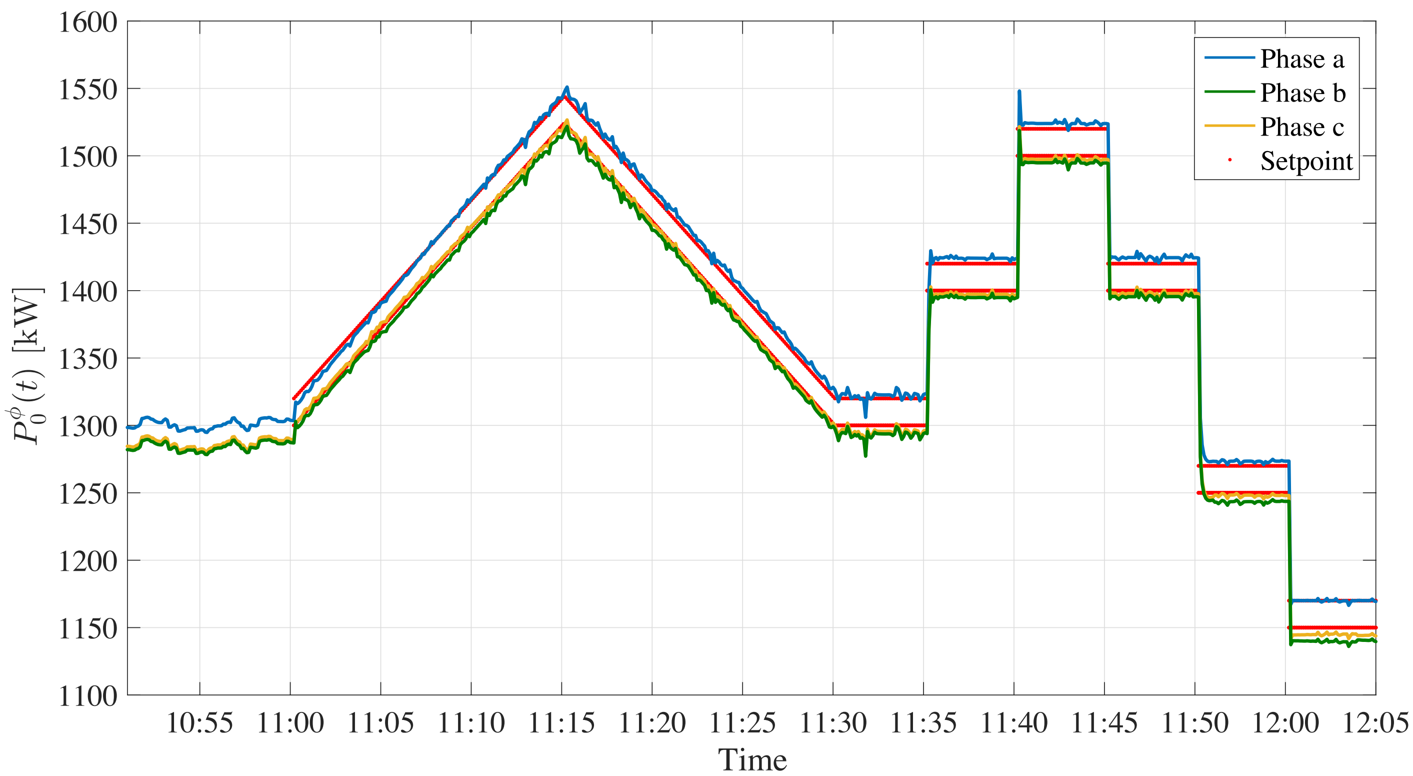
IEEE

# Example: real-time feedback-based OPF

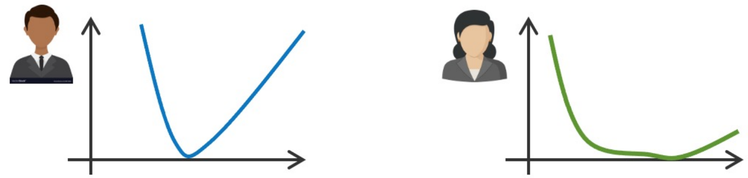
- Real circuit within the Southern California Edison
- PQ of inverters updated every 1s
- Mix of residential, commercial, and industrial customers

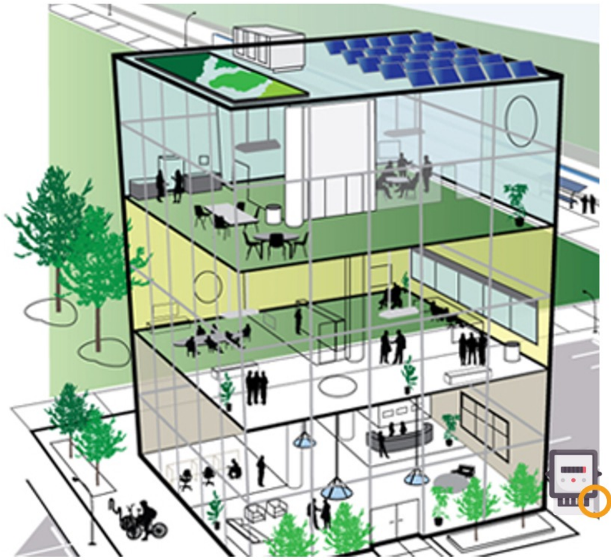


# Example: real-time feedback-based OPF



# Example: real-time feedback-based DM





- 30 DERs: HVACs, EVs, energy storage systems
- Rooms shared by multiple users

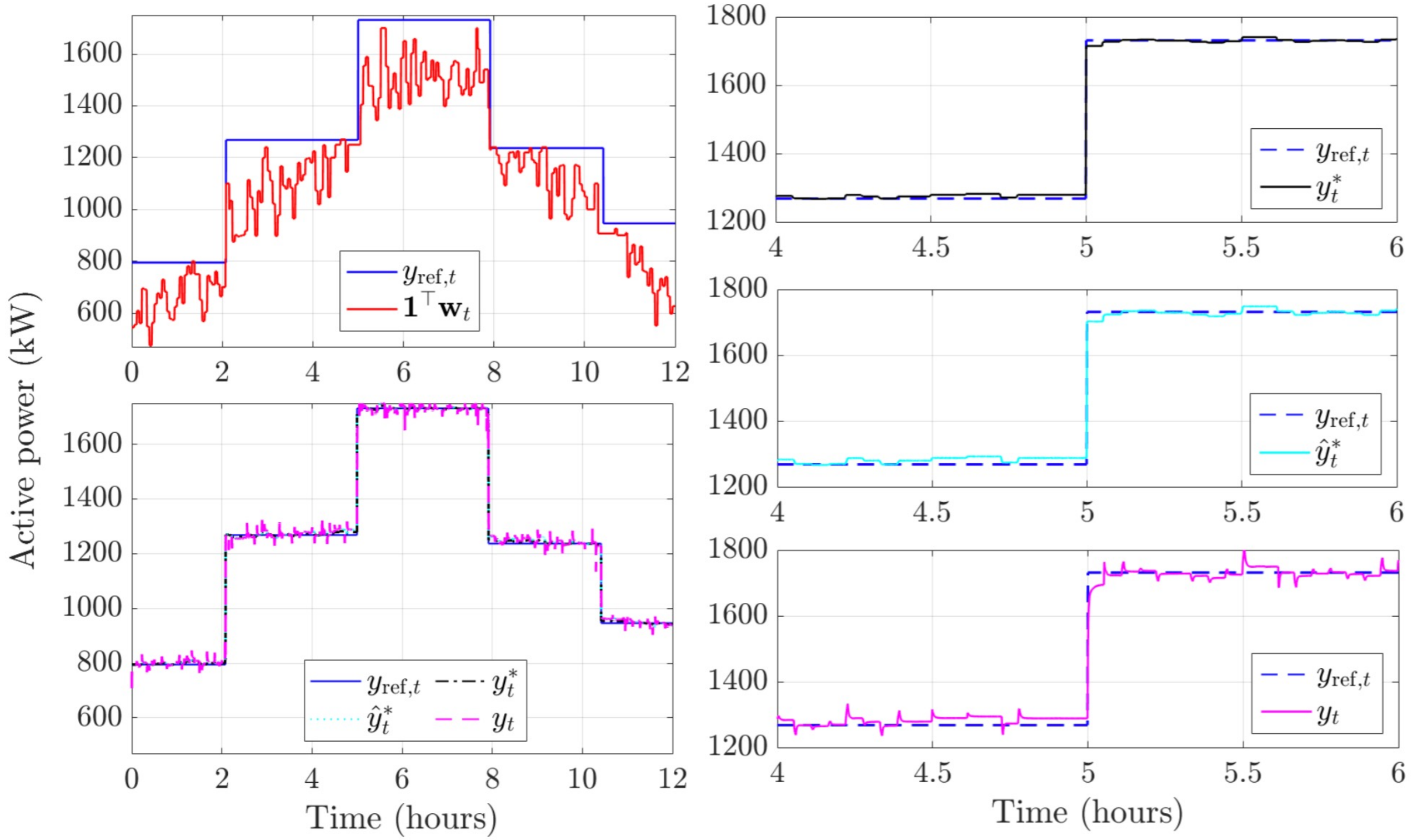
Total power consumption: $y^t = \sum_m x_m^t + \mathbf{1}^\top \mathbf{w}^t$

- **Task:** follow an AGC or DR signal

$$(y^t - y_{\text{ref}}^t)^2 \leq \epsilon$$

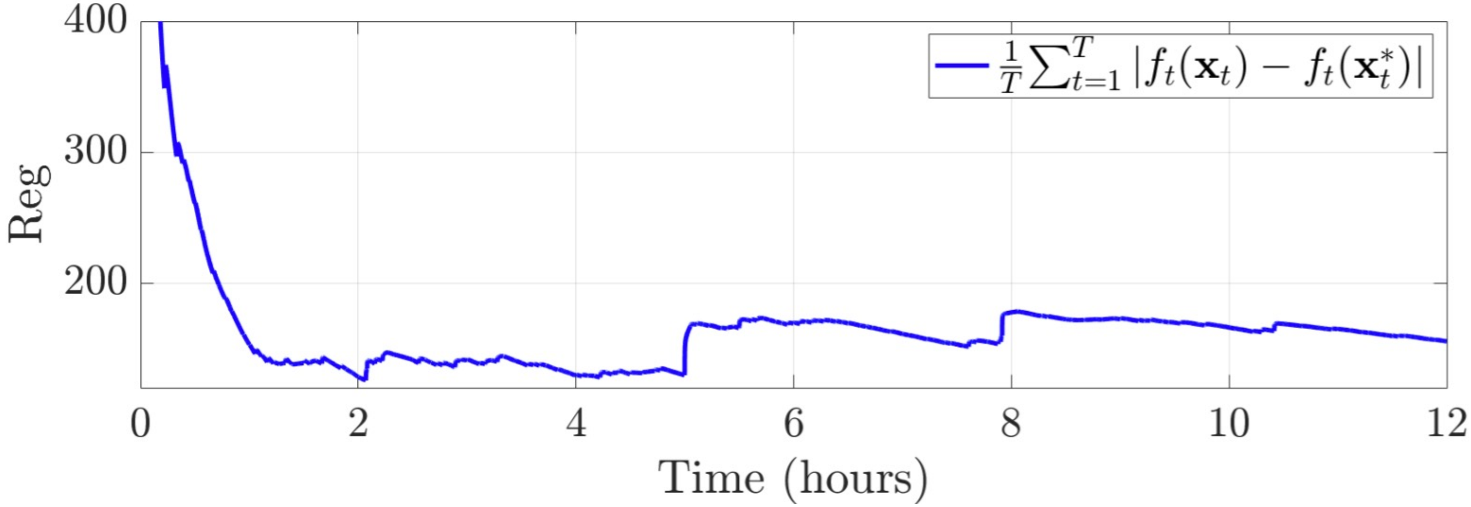
- Real data from NREL, granularity of 1 second

# Example: real-time feedback-based DM



# Example: real-time feedback-based DM

$$f_t(\mathbf{u}) := J_t(\mathbf{u}) + C_t(\mathbf{Gu} + \mathbf{Hw}_t)$$



# Thanks!

emiliano.dallanese@colorado.edu