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

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

- 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

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

\[ y(t) = \mathcal{F}(u(t), w(t); t) \]

- 184,000
- 46,000 households
Real-time decision-making

- 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


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
Yujie Tang
Steven Low
Andrey Bernstein
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

\[
\min_u c_0(y(u; t); t) + \sum_i c_i(u_i; t)
\]

subject to:

\[ u_i \in \mathcal{U}_i(t), \forall i \]
\[ g(y(u; t); t) \leq 0 \]

Time-varying systems

\[ y(t) = \mathcal{F}(u(t), w(t); t) \]
Modeling optimal trajectories

- **Model**: Continuous-time optimization

\[ \min_{u} \quad c_0(y(u; t); t) + \sum_{i} c_i(u_i; t) \]

subject to:

- \( u_i \in \mathcal{U}_i(t), \forall i \)
- \( g(y(u; t); t) \leq 0 \)

Time-varying systems

\[ y(t) = \mathcal{F}(u(t), w(t); t) \]
Modeling optimal trajectories

- **Model**: Continuous-time optimization

\[
\min_{u} \ c_0(y(u; t); t) + \sum_{i} c_i(u_i; t) \\
\text{subject to : } u_i \in U_i(t), \forall i \\
g(y(u; t); t) \leq 0
\]

- Example:

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

Time-varying systems

\[
y(t) = \mathcal{F}(u(t), w(t); t)
\]
Modeling optimal trajectories

- **Model:** Continuous-time optimization

\[
\min_u \quad c_0(y(u; t); t) + \sum_i c_i(u_i; t)
\]

subject to:

\[
u_i \in U_i(t), \forall i
\]

\[
g(y(u; t); t) \leq 0
\]

Time-varying systems

\[y(t) = \mathcal{F}(u(t), w(t); t)\]
Modeling optimal trajectories

- **Model:** Continuous-time optimization

\[
\min_{\mathbf{u}} \quad c_0(\mathbf{y}(\mathbf{u}; t); t) + \sum_i c_i(\mathbf{u}_i; t)
\]
subject to:
\[
\mathbf{u}_i \in \mathcal{U}_i(t), \forall i \\
g(\mathbf{y}(\mathbf{u}; t); t) \leq 0
\]

Time-varying systems
\[
\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)
\]
Modeling optimal trajectories

- **Model**: Continuous-time optimization

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

subject to:

\[ \mathbf{u}_i \in \mathcal{U}_i(t), \forall i \]

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

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

\[
\min_u \ c_0^{(k)}(y^{(k)}(u)) + \sum_i c_i^{(k)}(u_i)
\]
subject to:  \( u_i \in \mathcal{U}_i^{(k)}, \forall i \)
\[
g^{(k)}(y^{(k)}(u)) \leq 0
\]

Time-varying systems
\[
y(t) = \mathcal{F}(u(t), w(t); t)
\]

- 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

\[
\mathbf{u}^{(k,*)}
\]

**Time-varying systems**
\[
y(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)
\]

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

subject to: \( \mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall i \)
\[
g^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq 0
\]

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

---

![Diagram](https://via.placeholder.com/150)
Batch optimization

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

\( \mathbf{u}^{(k,*)} \)

Time-varying systems
\[
y(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)
\]

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

\[ u^{(k+1)} = T^{(k)}(u^{(k)}, w^{(k)}) \]

Time-varying systems
\[ y(t) = F(u(t), w(t); t) \]

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

\[ u^{(k+1)} = T^{(k)}(u^{(k)}, w^{(k)}) \]

**Time-varying systems**
\[ y(t) = \mathcal{F}(u(t), w(t); t) \]

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

\[ u^{(k+1)} = T^{(k)}(u^{(k)}, \hat{y}^{(k)}) \]

\( u^{(k)} \)  
\( \hat{y}^{(k)} \)

Time-varying systems
\[ y(t) = F(u(t), w(t); t) \]

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

\[ u^{(k+1)} = T^{(k)}(u^{(k)}, \hat{y}^{(k)}) \]

Time-varying systems

\[ y(t) = F(u(t), w(t); t) \]

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

\[ u^{(k+1)} = T^{(k)}(u^{(k)}, \hat{y}^{(k)}) \]

Time-varying systems
\[ y(t) = \mathcal{F}(u(t), w(t); t) \]

- Design and analysis of time-varying \( \epsilon \)-gradient methods [Bertsekas-Tsitsiklis’00]
- Can model fixed-point arithmetic, finite precision, and inexact maps
Formalizing optimal trajectories

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

$$\min_u c_0^{(k)}(y^{(k)}(u)) + \sum_i c_i^{(k)}(u_i)$$

subject to: \( u_i \in \mathcal{U}_i^{(k)}, \forall i \)

$$g^{(k)}(y^{(k)}(u)) \leq 0$$
Classes of problems and systems

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

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

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

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

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

  Convex optimization problem, linear map $\mathbf{y}^{(k)}(\mathbf{u}) = \mathbf{H}\mathbf{u} + \mathbf{Dw}^{(k)}$
Classes of problems and systems

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

  \[
  \min_u c^{(k)}_0(y^{(k)}(u)) + \sum_i c^{(k)}_i(u_i)
  \]

  subject to: \(u_i \in \mathcal{U}_i^{(k)}, \forall i\)

  \[g^{(k)}(y^{(k)}(u)) \leq 0\]

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

  ➡️ Convex optimization problem, linear map \(y^{(k)}(u) = Hu + Dw^{(k)}\)

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

  ➡️ Nonconvex optimization problem, nonlinear map \(y(t) = \mathcal{F}(u(t), w(t); t)\)
Classes of problems and systems

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

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

  subject to:  \[\mathbf{u}_i \in \mathcal{U}_i^{(k)}, \forall i\]

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

- **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 \[\mathbf{y}(t) = \mathcal{F}(\mathbf{u}(t), \mathbf{w}(t); t)\]

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

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

$$\min_u c_0^{(k)}(y^{(k)}(u)) + \sum_i c_i^{(k)}(u_i)$$

subject to: $u_i \in U_i^{(k)}, \forall i$

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

- $L^{(k)}(u, \lambda) := c_0^{(k)}(y^{(k)}(u)) + \sum_i c_i^{(k)}(u_i) + \lambda^T g^{(k)}(y^{(k)}(u))$
Preliminaries

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

$$\min_u c^{(k)}_0(y^{(k)}(u)) + \sum_i c^{(k)}_i(u_i)$$

subject to: $u_i \in \mathcal{U}_i^{(k)}, \forall i$

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

- $L^{(k)}(u, \lambda) := c^{(k)}_0(y^{(k)}(u)) + \sum_i c^{(k)}_i(u_i) + \lambda^T g^{(k)}(y^{(k)}(u))$

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

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

$$
\min_u c_0^{(k)}(y^{(k)}(u)) + \sum_i c_i^{(k)}(u_i)
$$

subject to: $u_i \in \mathcal{U}_i^{(k)}$, $\forall i$

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

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

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

- $L_r^{(k)}(u, \lambda) := L^{(k)}(u, \lambda) - \frac{r}{2} \|\lambda\|^2_2$ [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 $kh, k \in \mathbb{N}$):

$$\min_{\mathbf{u}} \ c^{(k)}_0(\mathbf{y}^{(k)}(\mathbf{u})) + \sum_{i} c^{(k)}_i(\mathbf{u}_i)$$

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

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

- $\sigma^{(k)} := h^{-1} \| \mathbf{z}^{(*,k+1)} \|_2 \approx \text{Drifting / gradient}$
Convex setting

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

\[
\min_u c_0^{(k)}(y^{(k)}(u)) + \sum_i c_i^{(k)}(u_i)
\]

subject to : $u_i \in U_i^{(k)}$, $\forall i$

\[
g^{(k)}(y^{(k)}(u)) \leq 0
\]

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

  ➡️ Convex optimization problem, linear map $y^{(k)}(u) = Hu + Dw^{(k)}$

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

  ➡️ Nonconvex optimization problem, nonlinear map $y(t) = F(u(t), w(t); t)$
Convex setting

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

$$\min_{u} c_0^{(k)}(y^{(k)}(u)) + \sum_{i} c_i^{(k)}(u_i)$$

subject to: \[ u_i \in \mathcal{U}_i^{(k)}, \forall i \]

\[ g^{(k)}(y^{(k)}(u)) \leq 0 \]

- (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

\[ u^{(k+1)} = \text{proj}_{U^{(k)}} \left\{ u^{(k)} - \alpha \nabla_u L_r^{(k)}(u^{(k)}, \lambda^{(k)}) \right\} \]\n\nprimal gradient descent

\[ \lambda^{(k+1)} = \text{proj}_D \left\{ \lambda^{(k)} + \alpha \nabla_\lambda L_r^{(k)}(u^{(k)}, \lambda^{(k)}) \right\} \]\ndual gradient ascent
Online primal-dual method

- Online implementation of the primal-dual method

\[
\begin{align*}
  u^{(k+1)} &= \text{proj}_{\mathcal{U}^{(k)}} \left\{ u^{(k)} - \alpha \nabla_u L_r^{(k)}(u^{(k)}, \lambda^{(k)}) \right\} \\
  \lambda^{(k+1)} &= \text{proj}_{\mathcal{D}} \left\{ \lambda^{(k)} + \alpha \nabla_\lambda L_r^{(k)}(u^{(k)}, \lambda^{(k)}) \right\}
\end{align*}
\]

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

- Contractive arguments
- Q-linear convergence
Online primal-dual method

- Online implementation of the primal-dual method

\[
\begin{align*}
    u^{(k+1)} &= \text{proj}_{U^{(k)}} \left\{ u^{(k)} - \alpha \nabla_u L_r^{(k)}(u^{(k)}, \lambda^{(k)}) \right\} \quad \text{primal gradient descent} \\
    \lambda^{(k+1)} &= \text{proj}_{D} \left\{ \lambda^{(k)} + \alpha \nabla_\lambda L_r^{(k)}(u^{(k)}, \lambda^{(k)}) \right\} \quad \text{dual gradient ascent}
\end{align*}
\]

- \( r = 0 \): Monotone, Lipschitz map

Dynamic regret analysis [Bernstein-Dall’Anese-Simonetto’18]
Online primal-dual method

- Online implementation of the primal-dual method

\[ u^{(k+1)} = \text{proj}_{\mathcal{U}^{(k)}} \left\{ u^{(k)} - \alpha \left( \sum_i \nabla c_i^{(k)}(u^{(k)}) + \nabla c_0^{(k)}(Hu^{(k)} + Dw^{(k)}) \right. \right. \]
\[ \left. \left. + \nabla^T [J_g(Hu^{(k)} + Dw^{(k)})] \lambda^{(k)} \right\} \right\} \]

\[ \lambda^{(k+1)} = \text{proj}_{\mathcal{D}^{(k)}} \left\{ (1 - \alpha r)\lambda^{(k)} + \alpha g^{(k)}(Hu^{(k)} + Dw^{(k)}) \right\} \]
Online primal-dual method

- Online implementation of the primal-dual method

\[
\begin{align*}
\mathbf{u}^{(k+1)} &= \text{proj}_{\mathcal{U}^{(k)}} \left\{ \mathbf{u}^{(k)} + \alpha \left( \sum_i \nabla c_i^{(k)}(\mathbf{u}^{(k)}) + \mathbf{H}^T \nabla c_0^{(k)}(\mathbf{H}u^{(k)} + \mathbf{D}w^{(k)}) + \mathbf{H}^T [\mathbf{J}_g(\mathbf{H}u^{(k)} + \mathbf{D}w^{(k)})]\right)^T \lambda^{(k)} \right\} \\
\lambda^{(k+1)} &= \text{proj}_{\mathcal{D}^{(k)}} \left\{ (1 - \alpha r)\lambda^{(k)} + \alpha \mathbf{g}^{(k)}(\mathbf{H}u^{(k)} + \mathbf{D}w^{(k)}) \right\}
\end{align*}
\]

- Feed-forward / autonomous system
- Model-based
- Map evaluated at each iteration
Online primal-dual method

- Online implementation of the primal-dual method

\[ u^{(k+1)} = \text{proj}_{\mathcal{U}^{(k)}} \left\{ u^{(k)} - \alpha \left( \sum_i \nabla c_i^{(k)}(u^{(k)}) + H^T \nabla c_0^{(k)}(\hat{y}^{(k)}) \right. \right. \]
\[ \left. \left. + H^T [J_g(\hat{y}^{(k)})] \lambda^{(k)} \right) \right\} \]

\[ \lambda^{(k+1)} = \text{proj}_{\mathcal{D}^{(k)}} \left\{ (1 - \alpha r)\lambda^{(k)} + \alpha \hat{g}^{(k)} \right\} \]
Online primal-dual method

- Online implementation of the primal-dual method

\[
\begin{align*}
\mathbf{u}^{(k+1)} &= \text{proj}_{\mathcal{U}^{(k)}} \left\{ \mathbf{u}^{(k)} - \alpha \left( \sum_i \nabla c_i^{(k)}(\mathbf{u}^{(k)}) + \mathbf{H}^T \nabla c_0^{(k)}(\hat{\mathbf{y}}^{(k)}) + \mathbf{H}^T [\mathbf{J}_g(\hat{\mathbf{y}}^{(k)})]^T \lambda^{(k)} \right) \right\} \\
\lambda^{(k+1)} &= \text{proj}_{\mathcal{D}^{(k)}} \left\{ (1 - \alpha r) \lambda^{(k)} + \alpha \hat{\mathbf{g}}^{(k)} \right\}
\end{align*}
\]

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

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

- (Assumption) \( \|\hat{y}^{(k)} - y^{(k)}(u^{(k)})\|_2 \leq e_y \) 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:

\[
e_p \leq (L_o + M_\lambda M_I L_g) \|H\|_2 e_y
\]

\[
e_d \leq M_g e_y.
\]

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

**Theorem.** If $\alpha$ is chosen such that:

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

then the following holds for the algorithm:

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

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

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

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

\[
\min_u \ c^{(k)}(u) \ \\
\text{s. to : } \ u_i \in \mathcal{U}^{(k)}_i, \forall i \\
\ g^{(k)}(y^{(k)}(u)) \leq 0
\]

- **Case 1**: Convex [Dall’Anese-Simonetto’16]
  
  Convex optimization problem, linear map $y^{(k)}(u) = Hu + Dw^{(k)}$

- **Case 2**: Nonconvex [Tang-Dall’Anese-Bernstein-Low’19]
  
  Nonconvex optimization problem, nonlinear map $y(t) = \mathcal{F}(u(t), w(t); t)$
Nonconvex setting

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

$$\min_{u} \quad c^{(k)}(u)$$

s. to: $u_i \in \mathcal{U}_i^{(k)}, \forall \ i$

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

- (As.) Functions are twice continuously differentiable

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

$$\lambda^T g^{(k)}(u^{(k,*)}) = 0 \quad \text{and} \quad [J_g^{(k)}(u^{(k,*)})]^T \lambda \in \mathcal{N}_{\mathcal{U}^{(k)}}(u^{(k,*)})$$

[generalization of the Mangasarian-Fromovitz constraint qualification]
Online primal-dual method

- Online implementation of the primal-dual method

\[
\begin{align*}
\mathbf{u}^{(k+1)} &= \text{proj}_{\mathcal{U}^{(k)}} \left\{ \mathbf{u}^{(k)} - \alpha \left( \nabla c^{(k)}(\mathbf{u}^{(k)}) + [\mathbf{J}_f(\mathbf{u}^{(k)})]^T [\mathbf{J}_g(\mathbf{y}^{(k)})]^T \lambda^{(k)} \right) \right\} \\
\lambda^{(k+1)} &= \text{proj}_{\mathbb{R}_+^d} \left\{ (1 - \alpha \eta r) \lambda^{(k)} + \eta \alpha \mathbf{g}^{(k)}(\mathbf{y}^{(k)}) \right\}
\end{align*}
\]

where \( \mathbf{y}^{(k)} = \mathcal{F}^{(k)}(\mathbf{u}^{(k)}, \mathbf{w}^{(k)}) \)
Online primal-dual method

- Online implementation of the primal-dual method

\[ u^{(k+1)} = \text{proj}_{\mathcal{U}(k)} \left\{ u^{(k)} - \alpha \left( \nabla c^{(k)}(u^{(k)}) + [Jf(u^{(k)})]^T [Jg(y^{(k)})]^T \lambda^{(k)} \right) \right\} \]

\[ \lambda^{(k+1)} = \text{proj}_{\mathbb{R}_+^d} \left\{ (1 - \alpha \eta r) \lambda^{(k)} + \eta \alpha g^{(k)}(y^{(k)}) \right\} \]

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

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

- Let’s work with this: \[ \|z\|_\eta := (\|u\|_2^2 + \eta^{-1} \|\lambda\|_2^2)^{1/2} \]
Some insights

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

\[ \Lambda_m \delta > M_{nc} \delta \sup_k \| \lambda^{(k,*)} \| \]
Some insights

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

\[ \Lambda_m \delta > M_{nc} \delta \sup_k \| \lambda^{(k,*)} \| \]

where:

\[ \Lambda_m(\delta) := \inf_t \inf_{x: \|x\| \leq \delta} \lambda_{\min} \left( \overline{H}_{\text{Lr}}^{nc}(x, t) + \frac{1}{2} \sum_{i=1}^{m} \lambda_i^*(t) \overline{H}_{g_i^c}(x, t) \right) \]

\[ \overline{H}_{\text{Lr}}^{nc}(x, t) := \int_0^1 \nabla_{uu}^2 L_{r}^{nc}(u^*(t) + \theta x, *^t(t), t) d\theta \]

\[ M_{nc}(\delta) := \sup_t \sup_{x: \|x\| \leq \delta} \| D_{uu} g^{nc}(u^*(t) + x, t) \| \]
Some insights

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

\[ \Lambda_m \delta > M_{nc} \delta \sup_k \| \lambda^{(k,*)} \| \]

where:

\[ \Lambda_m(\delta) := \inf_t \inf_{x: \|x\| \leq \delta} \lambda_{\min} \left( \bar{H}^{nc}_{L_r}(x, t) + \frac{1}{2} \sum_{i=1}^{m} \lambda^*_i(t) \bar{H} g^c_i(x, t) \right) \]

\[ \bar{H}^{nc}_{L_r}(x, t) := \int_0^1 \nabla_{uu}^2 L^{nc}_{r}(u^*(t) + \theta x, \ast(t), t) \, d\theta \]

\[ M_{nc}(\delta) := \sup_t \sup_{x: \|x\| \leq} \| D_{uu}^2 g^{nc}(u^*(t) + x, t) \| \]

- “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 \( \Lambda_m \delta > M_{nc} \delta \sup_k \|\lambda^{(k,\ast)}\| \) and \( \|z^{(1)}\|_{\eta} < \delta \).

Then the following holds for the algorithm:

\[
\lim_{k \to +\infty} \sup \|z^{(k)} - z^{(k,\ast)}\|_2 \leq \frac{\rho(\alpha,\eta)}{1} + \frac{K'}{\rho(\alpha,\eta)}
\]

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

**Theorem.** Assume that \( \Lambda_m \delta > M_{nc} \delta \sup_k \| \lambda^{(k,*)} \| \) and \( \| z^{(1)} \| z^{(*,1)} \| \eta < \delta \).

Then the following holds for the algorithm:

\[
\lim_{k \to +\infty} \sup \| z^{(k)} \| z^{(k,*)} \| 2 \leq \frac{\rho(\alpha, \eta)}{1 + \frac{K'}{\rho(\alpha, \eta)}}
\]

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

- Sufficient conditions to ensure \( \rho(\alpha, \eta) < 1 \)
How about measurements?

- Online implementation of the primal-dual method

\[
\begin{align*}
\mathbf{u}^{(k+1)} &= \text{proj}_{\mathcal{U}^{(k)}} \left\{ \mathbf{u}^{(k)} - \alpha \left( \nabla c^{(k)}(\mathbf{u}^{(k)}) + [\mathbf{J}_\mathcal{F}(\mathbf{u}^{(k)})]^T [\mathbf{J}_g(\mathbf{y}^{(k)})]^T \mathbf{\lambda}^{(k)} \right) \right\} \\
\mathbf{\lambda}^{(k+1)} &= \text{proj}_{\mathbb{R}_+^d} \left\{ (1 - \alpha \eta r) \mathbf{\lambda}^{(k)} + \eta \alpha \mathbf{g}^{(k)}(\mathbf{y}^{(k)}) \right\}
\end{align*}
\]
How about measurements?

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

\[
\begin{align*}
\mathbf{u}^{(k+1)} &= \text{proj}_{\mathcal{U}^{(k)}} \left\{ \mathbf{u}^{(k)} - \alpha \left( \nabla c^{(k)}(\mathbf{u}^{(k)}) + [\mathbf{J}_\mathcal{F}(\mathbf{u}^{(k)})]^{T} [\hat{\mathbf{J}}_g]^{T} \lambda^{(k)} \right) \right\} \\
\lambda^{(k+1)} &= \text{proj}_{\mathbb{R}^d_+} \left\{ (1 - \alpha \eta r) \lambda^{(k)} + \alpha \eta \hat{\mathbf{g}}^{(k)} \right\}
\end{align*}
\]
How about measurements?

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

\[
\begin{align*}
\mathbf{u}^{(k+1)} &= \text{proj}_{\mathcal{U}(k)} \left\{ \mathbf{u}^{(k)} - \alpha \left( \nabla c^{(k)}(\mathbf{u}^{(k)}) + [\mathbf{J}_F(\mathbf{u}^{(k)})]^T \hat{\mathbf{g}}^T \lambda^{(k)} \right) \right\} \\
\lambda^{(k+1)} &= \text{proj}_{\mathbb{R}_+^d} \left\{ (1 - \alpha \eta r) \lambda^{(k)} + \alpha \eta \hat{g}^{(k)} \right\}
\end{align*}
\]

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

- *Ongoing work*: general nonlinear constraints
Example: Application to Power Distribution Grids
Problem setup

\[ y(t) = \mathcal{F}(u(t), w(t); t) \]

- \( u_i \) \(\rightarrow\) Power commands, \( U_i \) hardware constraints
- \( w \) \(\rightarrow\) Powers of non-controllable assets
- \( y \) \(\rightarrow\) Voltage magnitudes, power flows
Problem setup

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

subject to:

\[
\mathbf{u}_i \in \mathcal{U}_i^{(k)}, \quad \forall i
\]

\[
\mathbf{g}^{(k)}(\mathbf{y}^{(k)}(\mathbf{u})) \leq 0
\]
Problem setup

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

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

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

\[
v_{\min} \leq |V_n^{(k)}| \leq v_{\max}
\]

Voltage Ratings ANSI C84.1
Problem setup

\[
\min_{u} \ c_{0}^{(k)}(y^{(k)}(u)) + \sum_{i} \ c_{i}^{(k)}(u_{i})
\]

subject to:  \( u_{i} \in \mathcal{U}_{i}^{(k)}, \forall i \)

\[
g^{(k)}(y^{(k)}(u)) \leq 0
\]

\[
|P_{0} - P_{0}^{(k, \text{set})}| \leq E
\]
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
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

<table>
<thead>
<tr>
<th>Time</th>
<th>-2.55</th>
<th>-2.50</th>
<th>-2.45</th>
<th>-2.40</th>
<th>-2.35</th>
<th>-2.30</th>
<th>-2.25</th>
<th>-2.20</th>
<th>-2.15</th>
<th>-2.10</th>
</tr>
</thead>
</table>

- $P_0(t)$ [MW] vs Time

- Proposed
- Optimization-based (1 min interval)
- Setpoint
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

![Graph showing power output over time for different phases](image)
Representative results

\[ P_0^\phi(t) \text{ [kW]} \]

Time


Phase a
Phase b
Phase c
Setpoint
Representative results

\[ P_0^\phi(t) \, [\text{kW}] \]

- Setpoint
- Power on phase c
- Power on phase c with 3x batteries
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

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

subject to $u_i \in U_i^{(k)} \quad \forall i = 1, \ldots, N$

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

- (As. 1) The set $U_i^{(k)}$ 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

- **Controllable DER**
- **Measurement unit**
- **Node 0**

Setpoints computation + command

Dual update

**AGG**

Setpoint update $P_j, Q_j$

Output powers

Setpoint update $P_i, Q_i$

**DER**

Update global control signal

Powers at the substation $P_0, Q_0$

Measurements

At utility $P_{0, set}$