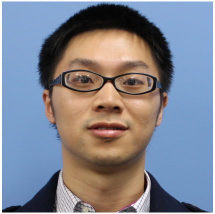
Primal-Dual Methods for Real-Time System Optimization

Andrey Bernstein
National Renewable Energy Laboratory

Theory of Reinforcement Learning Boot Camp, Sep 4 2020

Acknowledgments



Yue Chen

Adithya Devraj

Sean Meyn

NREL
NATIONAL RENEWABLE ENERGY LABORATORY

UF UNIVERSITY of FLORIDA

Real-Time System Optimization

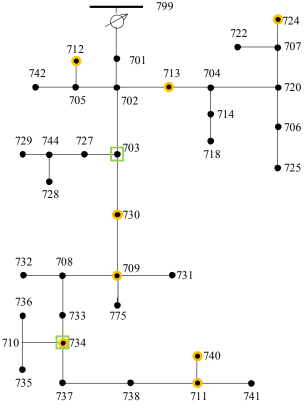
Consider a system described at time t by

$$\mathbf{y}(t) = \mathbf{h}_t(\mathbf{x}(t))$$

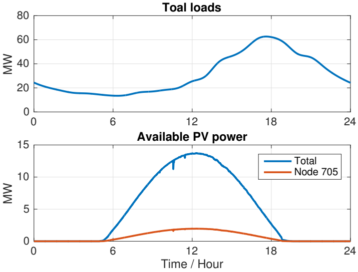
- $\mathbf{x}(t) \in \mathbb{R}^n$ is a vector of controllable inputs
- $\mathbf{y}(t) \in \mathbb{R}^m$ collects the system outputs
- $\mathbf{h}_t(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ is a time-varying map representing the algebraic system model

Example: Power Systems

- Power system



- $\mathbf{x}(t)$ – power injections of controllable devices
- $\mathbf{y}(t)$ – system voltages
- $\mathbf{h}_t(\cdot)$ – power-flow equations (Ohm + Kirchhoff)
- Time-varying: load, solar, topology changes



Real-Time System Optimization

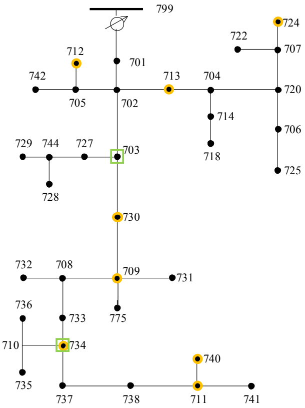
The desired behaviour of the system is defined via:

$$\min_{\mathbf{x} \in \mathcal{X}(t), \mathbf{y} = \mathbf{h}_t(\mathbf{x})} f_t(\mathbf{y})$$

- $\mathcal{X}(t)$ is a convex set of engineering constraints
- $f_t : \mathbb{R}^m \to \mathbb{R}$ is a convex function representing performance goals

Example: Optimal Power Flow (OPF)

- Power system



- Optimize generation cost and customer satisfaction
- Subject to device constraints and physics (power-flow equations)

Model-Based Feedforward Optimization

The desired behaviour of the system is defined via:

$$\min_{\mathbf{x}\in\mathcal{X}(t),\mathbf{y}=\mathbf{h}_t(\mathbf{x})} f_t(\mathbf{y}) \tag{1}$$

1. Obtain system model h_t and its Jacobian $\mathbf{J}_{\mathbf{h}_t}$.
2. Solve (1). E.g., projected-gradient method:

$$\mathbf{x}^{(k+1)} = \mathrm{proj}_{\mathcal{X}(t)}\left\{\mathbf{x}^{(k)} - \alpha(\mathbf{J}_t^{(k)})^\mathsf{T}\nabla_\mathbf{y} f_t(h_t(\mathbf{x}^{(k)}))\right\},\ k = 1, 2, \dots$$

$$\mathbf{J}_t^{(k)} := \mathbf{J}_{\mathbf{h}_t}(\mathbf{x}^{(k)})$$

Model-Based Feedforward Optimization

The desired behaviour of the system is defined via:

$$\min_{\mathbf{x} \in \mathcal{X}(t), \mathbf{y} = \mathbf{h}_t(\mathbf{x})} f_t(\mathbf{y}) \qquad (1)$$

1. Obtain system model h_t and its Jacobian $\mathbf{J}_{\mathbf{h}_t}$.
2. Solve (1). E.g., projected-gradient method:

$$\mathbf{x}^{(k+1)} = \mathrm{proj}_{\mathcal{X}(t)} \left\{ \mathbf{x}^{(k)} - \alpha (\mathbf{J}_t^{(k)})^\mathsf{T} \nabla_{\mathbf{y}} f_t(h_t(\mathbf{x}^{(k)})) \right\}, \; k = 1, 2, \ldots$$

$$\mathbf{J}_t^{(k)} := \mathbf{J}_{\mathbf{h}_t}(\mathbf{x}^{(k)})$$

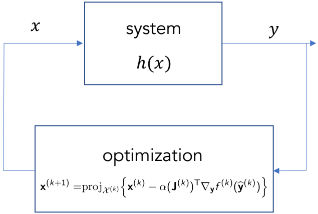
- Stringent real-time requirements... Can we run the above to convergence?
- Do we have model information in real time? E.g., forecasting uncontrollable inputs, topology information, etc.

Model-Based Feedback Optimization

At each (discrete) time step t_k:

1. Obtain a measurement $\widehat{\mathbf{y}}^{(k)}$ of the system output
2. Run a single optimization iteration:

$$\mathbf{x}^{(k+1)} = \mathrm{proj}_{\mathcal{X}^{(k)}}\left\{\mathbf{x}^{(k)} - \alpha(\mathbf{J}^{(k)})^\mathsf{T}\nabla_{\mathbf{y}} f^{(k)}(\widehat{\mathbf{y}}^{(k)})\right\}, \tag{2}$$

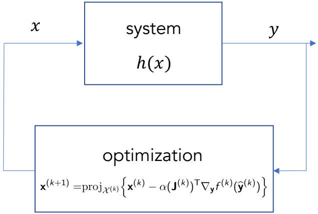


Model-Based Feedback Optimization

At each (discrete) time step t_k:

1. Obtain a measurement $\widehat{\mathbf{y}}^{(k)}$ of the system output
2. Run a single optimization iteration:

$$\mathbf{x}^{(k+1)} = \mathrm{proj}_{\mathcal{X}^{(k)}}\left\{\mathbf{x}^{(k)} - \alpha(\mathbf{J}^{(k)})^{\mathsf{T}}\nabla_{\mathbf{y}} f^{(k)}(\widehat{\mathbf{y}}^{(k)})\right\}, \tag{2}$$



Still requires model information in the form of $\mathbf{J}^{(k)}$!

Model-Free Feedback Optimization

Replace the gradient of $F^{(k)}(\mathbf{x}) := f^{(k)}(\mathbf{h}_{t_k}(\mathbf{x}))$

$$\nabla F^{(k)}(\mathbf{x}) = (\mathbf{J}_{\mathbf{h}_{t_k}}(\mathbf{x}))^{\mathsf{T}} \nabla_{\mathbf{y}} f^{(k)}(\mathbf{h}_{t_k}(\mathbf{x}))$$

with the zero-order approximation.

Model-Free Feedback Optimization

Replace the gradient of $F^{(k)}(\mathbf{x}) := f^{(k)}(\mathbf{h}_{t_k}(\mathbf{x}))$

$$\nabla F^{(k)}(\mathbf{x}) = (\mathbf{J}_{\mathbf{h}_{t_k}}(\mathbf{x}))^{\mathsf{T}} \nabla_{\mathbf{y}} f^{(k)}(\mathbf{h}_{t_k}(\mathbf{x}))$$

with the zero-order approximation.

- Single function evaluation:

$$\widehat{\nabla} F^{(k)}(\mathbf{x}; \boldsymbol{\xi}, \epsilon) := \frac{1}{\epsilon} \boldsymbol{\xi} F^{(k)}(\mathbf{x} + \epsilon \boldsymbol{\xi})$$

- $\boldsymbol{\xi} \in \mathbb{R}^n$ is an exploration vector
- $\epsilon > 0$ is a (small) scalar

- Two function evaluations:

$$\widehat{\nabla} F^{(k)}(\mathbf{x}; \boldsymbol{\xi}, \epsilon) := \frac{1}{2\epsilon} \boldsymbol{\xi} \left[F^{(k)}(\mathbf{x} + \epsilon \boldsymbol{\xi}) - F^{(k)}(\mathbf{x} - \epsilon \boldsymbol{\xi}) \right]$$

- Multiple evaluations...

Model-Free Feedback Optimization

This talk focuses on two function evaluation approximation:

$$\widehat{\nabla} F^{(k)}(\mathbf{x}; \boldsymbol{\xi}, \epsilon) := \frac{1}{2\epsilon} \boldsymbol{\xi} \left[F^{(k)}(\mathbf{x} + \epsilon \boldsymbol{\xi}) - F^{(k)}(\mathbf{x} - \epsilon \boldsymbol{\xi}) \right]$$

Motivation:

- Admits approximation:

  $$\widehat{\nabla} F(\mathbf{x}; \boldsymbol{\xi}, \epsilon) = \boldsymbol{\xi} \boldsymbol{\xi}^{\mathsf{T}} \nabla F(\mathbf{x}) + O(\epsilon^2)$$

  with $O(\epsilon^2) = 0$ for quadratic functions.
- Has nicer properties than single-evaluation: smaller variance, Lipschitz, etc

Related Work

- Le Blanc, 1922 - origin of Extremum Seeking? Kiefer and Wolfowitz, 1952. One-dimensional algorithm, no constraints.
- Spall, 1992. Stochastic perturbations, two function evaluations.
- Bhatnagar et al, 2003; Prashanth et al, 2019. Deterministic perturbations, static problem.
- Duchi et al, 2015; Nesterov and Spokoiny, 2017. Stochastic exploration, constrained problems.
- Bandit optimization literature (Awerbuch and Kleinberg, 2004, Bubeck and Cesa-Bianchi, 2012, etc): stochastic exploration, regret analysis.
- Extremum seeking literature (Ariyur and Krstic, 2003, etc): deterministic exploration, single evaluation
- Hajinezhad et al, 2019. Network optimization with stochastic exploration.

Our Focus

- Constrained time-varying networked systems optimization
- Using deterministic exploration signals – see Sean Meyn's talk for "Why?"
- Online distributed (light) primal-dual methods for real-time implementation
- Application to real-time optimal power flow in power networks

Networked Systems Optimization

Consider N systems interconnected via a network.

Desired behaviour of the network is defined via a time-varying convex optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad f_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{x})) + \sum_{i=1}^{N} f_i^{(k)}(\mathbf{x}_i) \tag{3a}$$

$$\text{subject to}: \ \mathbf{x}_i \in \mathcal{X}_i^{(k)},\ i = 1, \ldots, N \tag{3b}$$

$$g_j^{(k)}(\mathbf{y}^{(k)}(\mathbf{x})) \leq 0,\ j = 1, \ldots, M \tag{3c}$$

Desired Trajectory Formulation

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad f_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{x})) + \sum_{i=1}^{N} f_i^{(k)}(\mathbf{x}_i) \tag{4a}$$

$$\text{subject to}: \; \mathbf{x}_i \in \mathcal{X}_i^{(k)}, \; i = 1, \dots, N \tag{4b}$$

$$g_j^{(k)}(\mathbf{y}^{(k)}(\mathbf{x})) \leq 0, \; j = 1, \dots, M \tag{4c}$$

The desired trajectory $\mathbf{z}^{(*,k)} := (\mathbf{x}^{(*,k)}, \boldsymbol{\lambda}^{(*,k)})$ is the solution of:

$$\boxed{\max_{\boldsymbol{\lambda}\in\mathcal{D}^{(k)}} \min_{\mathbf{x}\in\mathcal{X}^{(k)}} \mathcal{L}_{p,d}^{(k)}(\mathbf{x}, \boldsymbol{\lambda}) \qquad k \in \mathbb{N}}$$

Desired Trajectory Formulation

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad f_0^{(k)}(\mathbf{y}^{(k)}(\mathbf{x})) + \sum_{i=1}^{N} f_i^{(k)}(\mathbf{x}_i) \tag{4a}$$

$$\text{subject to}: \; \mathbf{x}_i \in \mathcal{X}_i^{(k)}, \; i = 1, \ldots, N \tag{4b}$$

$$g_j^{(k)}(\mathbf{y}^{(k)}(\mathbf{x})) \leq 0, \; j = 1, \ldots, M \tag{4c}$$

The desired trajectory $\mathbf{z}^{(*,k)} := (\mathbf{x}^{(*,k)}, \boldsymbol{\lambda}^{(*,k)})$ is the solution of:

$$\boxed{\max_{\boldsymbol{\lambda}\in\mathcal{D}^{(k)}} \min_{\mathbf{x}\in\mathcal{X}^{(k)}} \mathcal{L}_{p,d}^{(k)}(\mathbf{x}, \boldsymbol{\lambda}) \qquad k \in \mathbb{N}}$$

$$\mathcal{L}_{p,d}^{(k)}(\mathbf{x}, \boldsymbol{\lambda}) := \mathcal{L}^{(k)}(\mathbf{x}, \boldsymbol{\lambda}) + \frac{p}{2}\|\mathbf{x}\|_2^2 - \frac{d}{2}\|\boldsymbol{\lambda}\|_2^2$$

- $\mathcal{L}^{(k)}(\mathbf{x}, \boldsymbol{\lambda})$ is the Lagrangian associated with (4)
- $\boldsymbol{\lambda} \in \mathbb{R}_+^M$ as the dual variable associated with (4c)
- $p \geq 0$, $d > 0$ are Tikhonov-type regularization parameters

First-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

First-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

[S1] (control application): Apply $\mathbf{x}^{(k)}$ to the system, and collect the measurement $\widehat{\mathbf{y}}^{(k)}$ of the output $\mathbf{y}^{(k)}(\mathbf{x}^{(k)})$.

First-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

[S1] (control application): Apply $\mathbf{x}^{(k)}$ to the system, and collect the measurement $\widehat{\mathbf{y}}^{(k)}$ of the output $\mathbf{y}^{(k)}(\mathbf{x}^{(k)})$.

[S2a] (gradient): Compute

$$\nabla \mathcal{L}^{(k)} := \nabla_{\mathbf{x}} f^{(k)}(\mathbf{x}^{(k)}) + (\mathbf{J}^k)^\mathsf{T} \nabla_{\mathbf{y}} f_0^{(k)}(\widehat{\mathbf{y}}^{(k)}) + (\nabla_{\mathbf{y}} \mathbf{g}^{(k)}(\widehat{\mathbf{y}}^{(k)}) \mathbf{J}^k)^\mathsf{T} \boldsymbol{\lambda}^{(k)} + p\mathbf{x}^{(k)}.$$

First-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

[S1] (control application): Apply $\mathbf{x}^{(k)}$ to the system, and collect the measurement $\widehat{\mathbf{y}}^{(k)}$ of the output $\mathbf{y}^{(k)}(\mathbf{x}^{(k)})$.

[S2a] (gradient): Compute

$$\begin{aligned}\nabla\mathcal{L}^{(k)} := \nabla_{\mathbf{x}} f^{(k)}(\mathbf{x}^{(k)}) + (\mathbf{J}^{k})^{\mathsf{T}} \nabla_{\mathbf{y}} f_0^{(k)}(\widehat{\mathbf{y}}^{(k)}) \\ + (\nabla_{\mathbf{y}} \mathbf{g}^{(k)}(\widehat{\mathbf{y}}^{(k)}) \mathbf{J}^{k})^{\mathsf{T}} \boldsymbol{\lambda}^{(k)} + p\mathbf{x}^{(k)}.\end{aligned}$$

[S2b] (primal step): Compute

$$\mathbf{x}^{(k+1)} = \mathrm{proj}_{\mathcal{X}^{(k)}} \left\{ \mathbf{x}^{(k)} - \alpha \widehat{\nabla} \mathcal{L}^{(k)} \right\}.$$

First-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

[S1] (control application): Apply $\mathbf{x}^{(k)}$ to the system, and collect the measurement $\widehat{\mathbf{y}}^{(k)}$ of the output $\mathbf{y}^{(k)}(\mathbf{x}^{(k)})$.

[S2a] (gradient): Compute

$$\begin{aligned}\nabla \mathcal{L}^{(k)} := \nabla_{\mathbf{x}} f^{(k)}(\mathbf{x}^{(k)}) + (\mathbf{J}^k)^\mathsf{T} \nabla_{\mathbf{y}} f_0^{(k)}(\widehat{\mathbf{y}}^{(k)}) \\ + (\nabla_{\mathbf{y}} \mathbf{g}^{(k)}(\widehat{\mathbf{y}}^{(k)}) \mathbf{J}^k)^\mathsf{T} \boldsymbol{\lambda}^{(k)} + p\mathbf{x}^{(k)}.\end{aligned}$$

[S2b] (primal step): Compute

$$\mathbf{x}^{(k+1)} = \mathrm{proj}_{\mathcal{X}^{(k)}} \left\{ \mathbf{x}^{(k)} - \alpha \widehat{\nabla} \mathcal{L}^{(k)} \right\}.$$

[S3] (dual step): Compute

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

First-Order Primal-Dual Algorithm with Feedback

At each time step k, perform the following steps:

[S1] (control application): Apply $\mathbf{x}^{(k)}$ to the system, and collect the measurement $\widehat{\mathbf{y}}^{(k)}$ of the output $\mathbf{y}^{(k)}(\mathbf{x}^{(k)})$.

[S2a] (gradient): Compute

$$\nabla \mathcal{L}^{(k)} := \nabla_{\mathbf{x}} f^{(k)}(\mathbf{x}^{(k)}) + (\mathbf{J}^{k})^{\mathsf{T}} \nabla_{\mathbf{y}} f_0^{(k)}(\widehat{\mathbf{y}}^{(k)}) + (\nabla_{\mathbf{y}} \mathbf{g}^{(k)}(\widehat{\mathbf{y}}^{(k)}) \mathbf{J}^{k})^{\mathsf{T}} \boldsymbol{\lambda}^{(k)} + p\mathbf{x}^{(k)}.$$

[S2b] (primal step): Compute

$$\mathbf{x}^{(k+1)} = \mathrm{proj}_{\mathcal{X}^{(k)}} \left\{ \mathbf{x}^{(k)} - \alpha \widehat{\nabla} \mathcal{L}^{(k)} \right\}.$$

[S3] (dual step): Compute

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

Zero-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

Zero-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

[S1a] (exploration): Apply $\mathbf{x}_+^{(k)} := \mathbf{x}^{(k)} + \epsilon \boldsymbol{\xi}^{(k)}$ and $\mathbf{x}_-^{(k)} := \mathbf{x}^{(k)} - \epsilon \boldsymbol{\xi}^{(k)}$, and collect measurements $\widehat{\mathbf{y}}_+^{(k)}$ and $\widehat{\mathbf{y}}_-^{(k)}$.

Zero-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

[S1a] (exploration): Apply $\mathbf{x}_+^{(k)} := \mathbf{x}^{(k)} + \epsilon \boldsymbol{\xi}^{(k)}$ and $\mathbf{x}_-^{(k)} := \mathbf{x}^{(k)} - \epsilon \boldsymbol{\xi}^{(k)}$, and collect measurements $\widehat{\mathbf{y}}_+^{(k)}$ and $\widehat{\mathbf{y}}_-^{(k)}$.

[S1b] (control application): Apply $\mathbf{x}^{(k)}$ to the system, and collect the measurement $\widehat{\mathbf{y}}^{(k)}$ of the output $\mathbf{y}^{(k)}(\mathbf{x}^{(k)})$.

Zero-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

[S1a] (exploration): Apply $\mathbf{x}_+^{(k)} := \mathbf{x}^{(k)} + \epsilon \boldsymbol{\xi}^{(k)}$ and $\mathbf{x}_-^{(k)} := \mathbf{x}^{(k)} - \epsilon \boldsymbol{\xi}^{(k)}$, and collect measurements $\widehat{\mathbf{y}}_+^{(k)}$ and $\widehat{\mathbf{y}}_-^{(k)}$.

[S1b] (control application): Apply $\mathbf{x}^{(k)}$ to the system, and collect the measurement $\widehat{\mathbf{y}}^{(k)}$ of the output $\mathbf{y}^{(k)}(\mathbf{x}^{(k)})$.

[S2a] (approximate gradient): Compute

$$
\begin{aligned}
\widehat{\nabla}\mathcal{L}^{(k)} := \nabla_{\mathbf{x}} f^{(k)}(\mathbf{x}^{(k)}) \\
&+ \frac{1}{2\epsilon}\boldsymbol{\xi}^{(k)}\left[f_0^{(k)}(\widehat{\mathbf{y}}_+^{(k)}) - f_0^{(k)}(\widehat{\mathbf{y}}_-^{(k)})\right] \\
&+ \frac{1}{2\epsilon}\boldsymbol{\xi}^{(k)}(\boldsymbol{\lambda}^{(k)})^{\mathsf{T}}\left[\mathbf{g}^{(k)}(\widehat{\mathbf{y}}_+^{(k)}) - \mathbf{g}^{(k)}(\widehat{\mathbf{y}}_-^{(k)})\right] + p\mathbf{x}^{(k)}.
\end{aligned}
$$

Zero-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

[S1a] (exploration): Apply $\mathbf{x}_+^{(k)} := \mathbf{x}^{(k)} + \epsilon \boldsymbol{\xi}^{(k)}$ and $\mathbf{x}_-^{(k)} := \mathbf{x}^{(k)} - \epsilon \boldsymbol{\xi}^{(k)}$, and collect measurements $\widehat{\mathbf{y}}_+^{(k)}$ and $\widehat{\mathbf{y}}_-^{(k)}$.

[S1b] (control application): Apply $\mathbf{x}^{(k)}$ to the system, and collect the measurement $\widehat{\mathbf{y}}^{(k)}$ of the output $\mathbf{y}^{(k)}(\mathbf{x}^{(k)})$.

[S2a] (approximate gradient): Compute

$$\begin{aligned}\widehat{\nabla}\mathcal{L}^{(k)} := \nabla_{\mathbf{x}} f^{(k)}(\mathbf{x}^{(k)}) \\ &+ \frac{1}{2\epsilon}\boldsymbol{\xi}^{(k)}\left[f_0^{(k)}(\widehat{\mathbf{y}}_+^{(k)}) - f_0^{(k)}(\widehat{\mathbf{y}}_-^{(k)})\right] \\ &+ \frac{1}{2\epsilon}\boldsymbol{\xi}^{(k)}(\boldsymbol{\lambda}^{(k)})^{\mathsf{T}}\left[\mathbf{g}^{(k)}(\widehat{\mathbf{y}}_+^{(k)}) - \mathbf{g}^{(k)}(\widehat{\mathbf{y}}_-^{(k)})\right] + p\mathbf{x}^{(k)}.\end{aligned}$$

[S2b] (approximate primal step): Compute

$$\mathbf{x}^{(k+1)} = \mathrm{proj}_{\mathcal{X}^{(k)}}\left\{\mathbf{x}^{(k)} - \alpha\widehat{\nabla}\mathcal{L}^{(k)}\right\}.$$

Zero-Order Primal-Dual Algorithm

At each time step k, perform the following steps:

[S1a] (exploration): Apply $\mathbf{x}_{+}^{(k)} := \mathbf{x}^{(k)} + \epsilon \boldsymbol{\xi}^{(k)}$ and $\mathbf{x}_{-}^{(k)} := \mathbf{x}^{(k)} - \epsilon \boldsymbol{\xi}^{(k)}$, and collect measurements $\widehat{\mathbf{y}}_{+}^{(k)}$ and $\widehat{\mathbf{y}}_{-}^{(k)}$.

[S1b] (control application): Apply $\mathbf{x}^{(k)}$ to the system, and collect the measurement $\widehat{\mathbf{y}}^{(k)}$ of the output $\mathbf{y}^{(k)}(\mathbf{x}^{(k)})$.

[S2a] (approximate gradient): Compute

$$\begin{aligned}\widehat{\nabla}\mathcal{L}^{(k)} := {} & \nabla_{\mathbf{x}} f^{(k)}(\mathbf{x}^{(k)}) \\ & + \frac{1}{2\epsilon}\boldsymbol{\xi}^{(k)}\left[f_0^{(k)}(\widehat{\mathbf{y}}_{+}^{(k)}) - f_0^{(k)}(\widehat{\mathbf{y}}_{-}^{(k)})\right] \\ & + \frac{1}{2\epsilon}\boldsymbol{\xi}^{(k)}(\boldsymbol{\lambda}^{(k)})^{\mathsf{T}}\left[\mathbf{g}^{(k)}(\widehat{\mathbf{y}}_{+}^{(k)}) - \mathbf{g}^{(k)}(\widehat{\mathbf{y}}_{-}^{(k)})\right] + p\mathbf{x}^{(k)}.\end{aligned}$$

[S2b] (approximate primal step): Compute

$$\mathbf{x}^{(k+1)} = \mathrm{proj}_{\mathcal{X}^{(k)}}\left\{\mathbf{x}^{(k)} - \alpha\widehat{\nabla}\mathcal{L}^{(k)}\right\}.$$

[S3] (dual step): Compute

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

Assumptions

1. The exploration signal $\boldsymbol{\xi}^{(k)}$ is deterministic, sampled from a continuous-time signal $\boldsymbol{\xi}(t)$ satisfying

$$\frac{1}{T}\int_{t}^{t+T} \boldsymbol{\xi}(\tau)\boldsymbol{\xi}(\tau)^{\mathsf{T}}\, d\tau = \mathbf{I}, \quad \text{for some } T > 0.$$

Assumptions

1. The exploration signal $\boldsymbol{\xi}^{(k)}$ is deterministic, sampled from a continuous-time signal $\boldsymbol{\xi}(t)$ satisfying

$$\frac{1}{T}\int_t^{t+T} \boldsymbol{\xi}(\tau)\boldsymbol{\xi}(\tau)^\mathsf{T}\, d\tau = \mathbf{I}, \quad \text{for some } T > 0.$$

E.g., $\boldsymbol{\xi}_i(t) = \sqrt{2}\sin(\omega_i t), \quad i = 1, \ldots, n, \, \omega_i \neq \omega_j, \forall i \neq j.$
(T is a common integer multiple of the sinusoidal signal periods.)

Assumptions

1. The exploration signal $\boldsymbol{\xi}^{(k)}$ is deterministic, sampled from a continuous-time signal $\boldsymbol{\xi}(t)$ satisfying

$$\frac{1}{T}\int_t^{t+T} \boldsymbol{\xi}(\tau)\boldsymbol{\xi}(\tau)^{\mathsf{T}}\, d\tau = \mathbf{I}, \quad \text{for some } T > 0.$$

   E.g., $\boldsymbol{\xi}_i(t) = \sqrt{2}\sin(\omega_i t), \quad i = 1, \ldots, n,\ \omega_i \neq \omega_j, \forall i \neq j.$
   (T is a common integer multiple of the sinusoidal signal periods.)
2. The projection in the primal step is active every T time units.

Assumptions

1. The exploration signal $\boldsymbol{\xi}^{(k)}$ is deterministic, sampled from a continuous-time signal $\boldsymbol{\xi}(t)$ satisfying

$$\frac{1}{T}\int_{t}^{t+T} \boldsymbol{\xi}(\tau)\boldsymbol{\xi}(\tau)^{\mathsf{T}}\, d\tau = \mathbf{I}, \quad \text{for some } T > 0.$$

   E.g., $\boldsymbol{\xi}_i(t) = \sqrt{2}\sin(\omega_i t), \quad i = 1, \ldots, n,\ \omega_i \neq \omega_j, \forall i \neq j.$
   (T is a common integer multiple of the sinusoidal signal periods.)
2. The projection in the primal step is active every T time units.
3. Variability of the desired trajectory and gradients is bounded:

$$\sup_{k\geq 0} \|\mathbf{z}^{(*,k+1)} - \mathbf{z}^{(*,k)}\|_2 \leq \sigma, \ \sup_{k\geq 0} \|\nabla f^{(k)}(\mathbf{x}) - \nabla f^{(k-1)}(\mathbf{x})\| \leq e_f$$

   and similarly for other functions.

Assumptions

1. The exploration signal $\boldsymbol{\xi}^{(k)}$ is deterministic, sampled from a continuous-time signal $\boldsymbol{\xi}(t)$ satisfying

$$\frac{1}{T}\int_t^{t+T} \boldsymbol{\xi}(\tau)\boldsymbol{\xi}(\tau)^{\mathsf{T}}\, d\tau = \mathbf{I}, \quad \text{for some } T > 0.$$

   E.g., $\boldsymbol{\xi}_i(t) = \sqrt{2}\sin(\omega_i t), \quad i = 1, \ldots, n, \, \omega_i \neq \omega_j, \forall i \neq j.$
   (T is a common integer multiple of the sinusoidal signal periods.)
2. The projection in the primal step is active every T time units.
3. Variability of the desired trajectory and gradients is bounded:

$$\sup_{k \geq 0} \|\mathbf{z}^{(*,k+1)} - \mathbf{z}^{(*,k)}\|_2 \leq \sigma, \; \sup_{k \geq 0} \|\nabla f^{(k)}(\mathbf{x}) - \nabla f^{(k-1)}(\mathbf{x})\| \leq e_f$$

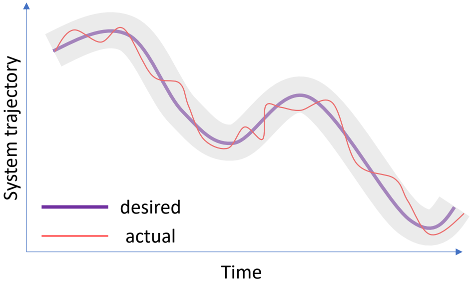
   and similarly for other functions.
4. Measurement error is bounded by e_y.

Tracking Result

Theorem

There exist $\alpha > 0$, $\varepsilon = O(\alpha + \epsilon^2 + e_f + e_y)$, *and* $c < 1$ *such that the sequence* $\{\mathbf{z}^{(k)}\}$ *converges Q-linearly to* $\{\mathbf{z}^{(*,k)}\}$ *up to an asymptotic error bound given by:*

$$\limsup_{k \to \infty} \|\mathbf{z}^{(k)} - \mathbf{z}^{(*,k)}\|_2 \leq \frac{\alpha\varepsilon + \sigma}{1 - c}.$$



Proof Idea

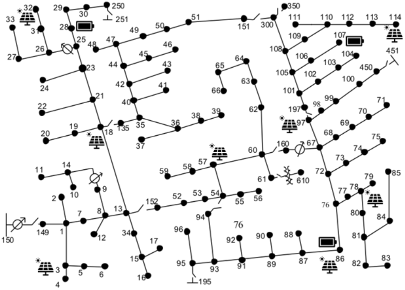
- Use QSA (Sean Meyn's talk) – currently works mostly with diminishing step size and no projection; or
- Prove directly – see:

  Y. Chen, A. Bernstein, A. Devraj, S. Meyn, "Model-free primal-dual methods for network optimization with application to real-time optimal power flow," 2020 American Control Conference (ACC), 3140-3147.

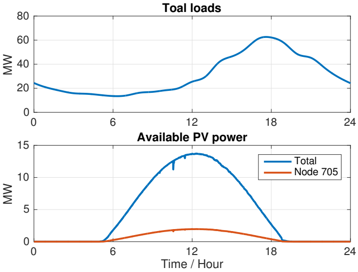
Application: Optimal Power Flow

Real-time optimization of the power injections of distributed energy resources (DERs) in a power system.

- IEEE 123-node test feeder
- 8 solar (PV) systems
- 3 battery systems



- Two possible network configurations
- Total load and available PV generation:



Application: Optimal Power Flow

Real-time optimization of the power injections of distributed energy resources (DERs) in a power system.

- Control variables

  $\mathbf{x} \in \mathbb{R}^{2N_{der}}$: active and reactive power injection of DERs; $\mathbf{x}_i = \{x_{i,p},\, x_{i,q}\}$

- Output variables

  $\mathbf{y} \in \mathbb{R}^{N_{buses}+1}$: voltages and feeder head power; $\mathbf{y} = \{\mathbf{v}, P_0\}$

- Objectives

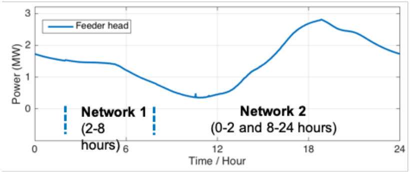
$$\text{Feeder head power following: } f_0(\mathbf{y}) = (P_0 - P_0^{\bullet})^2$$
$$\text{Local DER objective: } f_i(\mathbf{x}_i) = c_i(x_{i,p} - x_{i,p}^{\bullet})^2$$

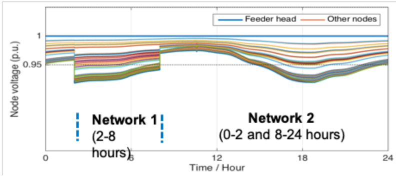
- Constraints

$$\text{Node voltage: } \underline{V}_i \leq v_i(\mathbf{x}) \leq \overline{V}_i$$
$$\text{Battery system: } \underline{X}_{i,p} \leq x_{i,p} \leq \overline{X}_{i,p}, \quad x_{i,p}^2 + x_{i,q}^2 \leq (\overline{S}_i^{bt})^2$$
$$\underline{SOC}_i \leq SOC_i \leq \overline{SOC}_i$$
$$\text{PV system: } 0 \leq x_{i,p} \leq \overline{X}_i^{pv}, \quad x_{i,p}^2 + x_{i,q}^2 \leq (\overline{S}_i^{pv})^2$$

Numerical Study: Results

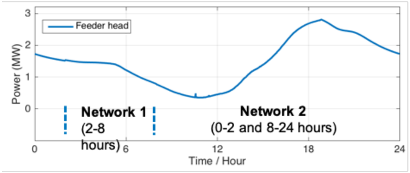
Uncontrolled behavior (no battery control and PV curtailment)

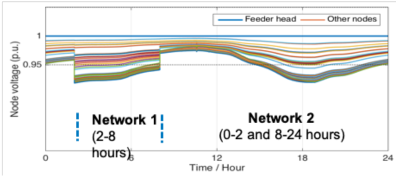




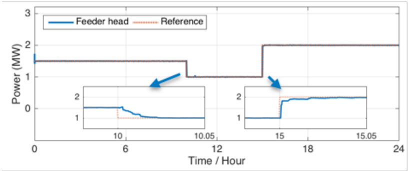
Numerical Study: Results

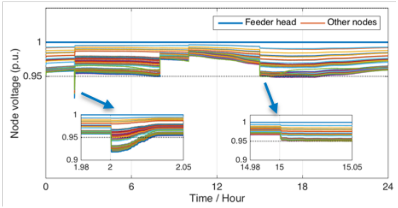
Uncontrolled behavior (no battery control and PV curtailment)





Real-time model-free optimization:





Numerical Study: Sensitivity to Noise

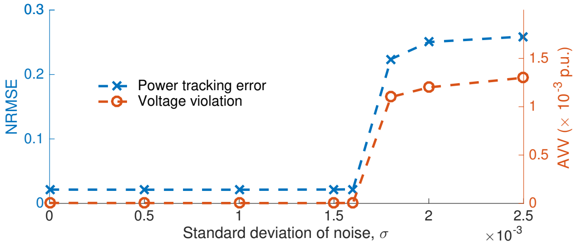
- Performance metric

$$\text{NRMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\frac{P_0^{(k)} - P_0^{\bullet(k)}}{P_0^{\bullet(k)}}\right)^2}$$

$$\text{AVV} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\left([v_i^{(k)} - \overline{V}_i]_+ + [\underline{V}_i - v_i^{(k)}]_+\right)$$

- Sensitivity to measurement noise

$$\widehat{y}_i^{(k)} = y_i^{(k)} + W y_i^{(k)}, \quad W \sim \mathcal{N}(0, \sigma^2)$$



Conclusion

- Real-time primal-dual methods to track desired trajectories of networked systems
- Zero-order deterministic feedback-based approximations
- Stability and tracking results
- Application to OPF

References

Y. Chen, A. Bernstein, A. Devraj, S. Meyn, "Model-free primal-dual methods for network optimization with application to real-time optimal power flow," 2020 American Control Conference (ACC), 3140-3147.

A. Bernstein, E. Dall'Anese, A. Simonetto, "Online primal-dual methods with measurement feedback for time-varying convex optimization," IEEE Transactions on Signal Processing 67 (8), 1978-1991, 2019.

M. Colombino, J.W. Simpson-Porco, A. Bernstein, "Towards robustness guarantees for feedback-based optimization," 2019 IEEE 58th Conference on Decision and Control (CDC), 6207-6214.

M. Colombino, E. Dall'Anese, A. Bernstein, "Online optimization as a feedback controller: Stability and tracking," IEEE Transactions on Control of Network Systems 7 (1), 422-432, 2019

C.Y. Chang, M. Colombino, J. Cortes, E. Dall'Anese, "Saddle-flow dynamics for distributed feedback-based optimization," IEEE Control Systems Letters 3 (4), 948-953, 2019.