ECE 598 Computational Power Systems

Online Optimization Algorithms

Vladimir Dvorkin

University of Michigan

Last lecture recap

- Look around you and form teams of 2 people (1 min)
- Quickly review your notes or the slide deck (1 min)
- Share your **three** personal highlights with your partner (3 min)
- Get iClicker app ready

When to solve AC-OPF?

Planning (offline) stage

- Uncertainty of problem parameters (e.g, load and renewable forecast)
- Economically efficient generator schedule (set-points)
- Anticipation of contingencies
- Long comput. horizon (hours)

Real-time (online) stage

- Actual problem parameters (e.g, actual load and renewable realizations)
- Adjustments of scheduled generation to manage any deviation
- Real-time contingencies
- Narrow comput. horizon (seconds)

When to solve AC-OPF?

Planning (offline) stage

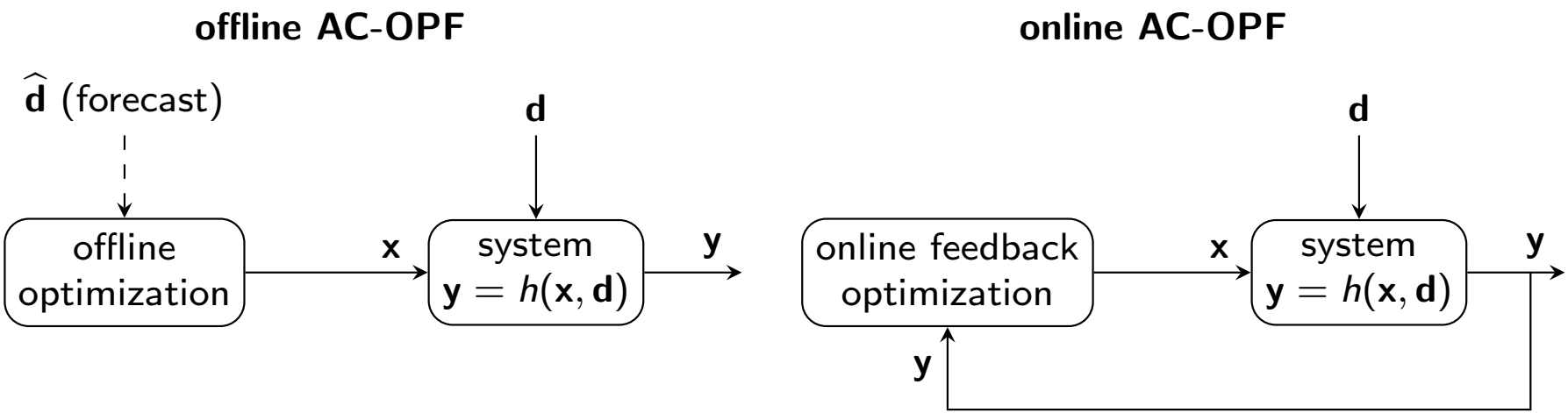
- Uncertainty of problem parameters (e.g, load and renewable forecast)
- Economically efficient generator schedule (set-points)
- Anticipation of contingencies
- Long comput. horizon (hours)

Real-time (online) stage

- Actual problem parameters (e.g, actual load and renewable realizations)
- Adjustments of scheduled generation to manage any deviation
- Real-time contingencies
- Narrow comput. horizon (seconds)

At both planning and real-time stages!
(but using different optimization algorithms)

Towards online AC-OPF



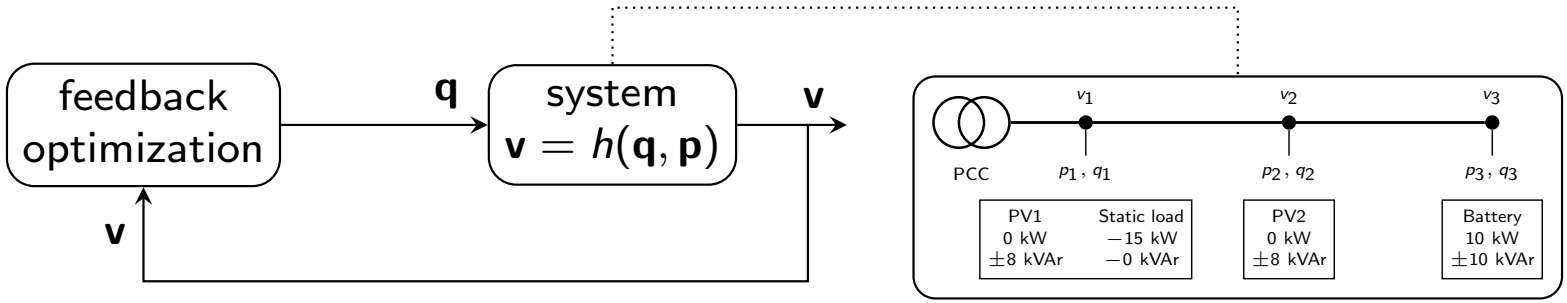
Purpose of online AC-OPF optimization:

- Power flow and injection adjustment given actual loads (as opposed to forecast)
- Tracking of AC-OPF solutions (as opposed to enforcing offline solutions)
- Recognize grid dynamics while acting on steady-state equations

Benefits of online AC-OPF optimization:

- Increases the robustness against time-varying disturbances
- Reduces model-dependence, i.e., make optimization model-free
- Fast response to line and generator outages
- Minimizes computational effort

Online distribution AC-OPF



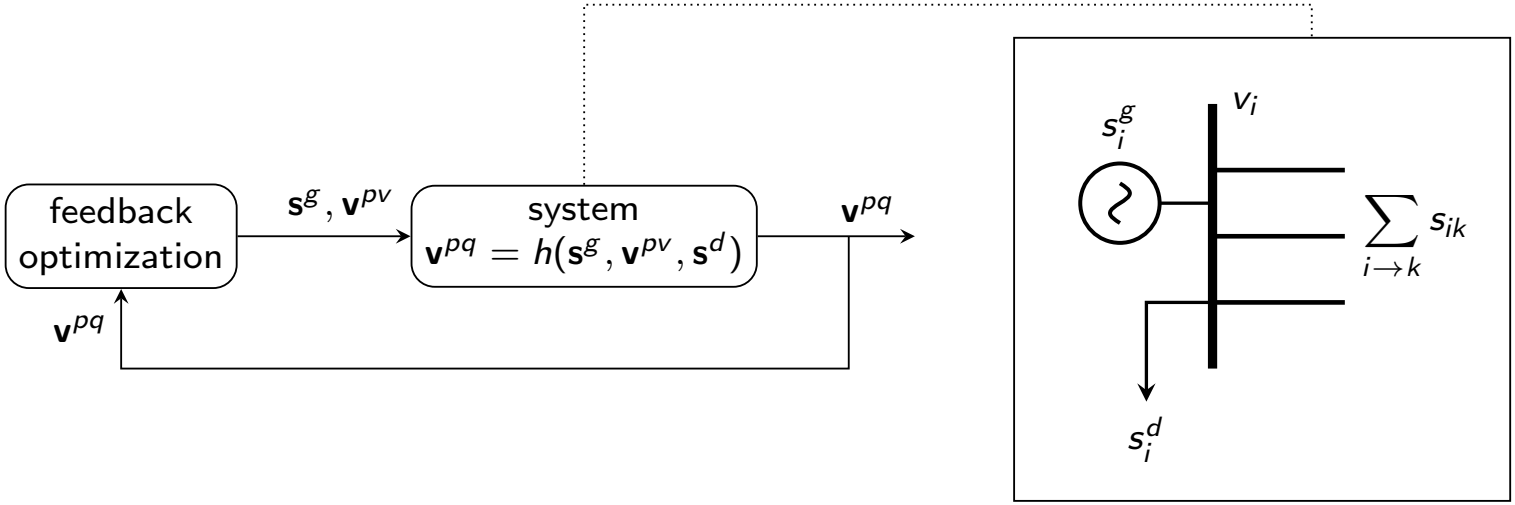
Feedback optimization:

$$
\begin{aligned}
\underset{\mathbf{q}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{q}^\top \mathbf{C}\mathbf{q} && \text{cost function} \\
\text{subject to} \quad & \mathbf{v} = v_0\mathbf{1} + \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} && \text{LinDistFlow equations} \\
& \underline{\mathbf{q}} \leqslant \mathbf{q} \leqslant \overline{\mathbf{q}} && \text{injection limits} \\
& \underline{\mathbf{v}} \leqslant \mathbf{v} \leqslant \overline{\mathbf{v}} && \text{voltage limits}
\end{aligned}
$$

- $\mathbf{q}$ inverter's reactive injection
- $\mathbf{v}$ squared voltage magnitude
- $\mathbf{p}$ uncontrolled active power injection

How is this optimization different from offline AC-OPF?

Online transmission AC-OPF



$$
\begin{aligned}
\underset{\mathbf{s}^g, \mathbf{v}^{pv}}{\text{minimize}} \quad & c(\mathbf{s}^g) \\
\text{subject to} \quad & s_i^g - s_i^d = \sum_{i \to k} s_{ik}(v_i, v_k) && \text{complex power flow} \\
& \underline{s}_i^g \leqslant s_i^g \leqslant \overline{s}_i^g && \text{generation limits} \\
& \underline{v}_i \leqslant v_i \leqslant \overline{v}_i && \text{voltage limits} \\
& s_{ik}(v_i, v_k) \leqslant \overline{s}_{ik}, \quad \forall (i,k) && \text{complex power flow limits}
\end{aligned}
$$

How is this optimization different from offline AC-OPF?

Online optimization algorithms

Solving non-linear optimization as dynamical systems

- Non-linear optimization program

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g(\mathbf{x}) \leqslant \mathbf{0} \quad : \boldsymbol{\mu} \\ & h(\mathbf{x}) = \mathbf{0} \quad : \boldsymbol{\lambda} \end{aligned}$$

  where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is strictly convex, $g : \mathbb{R}^n \mapsto \mathbb{R}^m$, $h : \mathbb{R}^n \mapsto \mathbb{R}^k$

- Let $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n \mid g(\mathbf{x}) \leqslant \mathbf{0}, h(\mathbf{x}) = \mathbf{0}\}$ denote its feasible set.

- Our goal is to solve this non-linear optimization by designing a dynamical system

$$\dot{x} = F(x)$$

  that convergences to the optimal solution $\mathbf{x}^\star$

- Forward Euler discretization:

$$\begin{aligned} \dot{\mathbf{x}} = \quad F(\mathbf{x}) &\Longleftrightarrow \mathbf{x}_{t+1} = \mathbf{x}_t + \eta F(\mathbf{x}_t) \\ \dot{\mathbf{x}} = -F(\mathbf{x}) &\Longleftrightarrow \mathbf{x}_{t+1} = \mathbf{x}_t - \eta F(\mathbf{x}_t) \end{aligned}$$

Solving non-linear optimization as dynamical systems (cont'd)

- The Lagrange optimality conditions give us an idea of such dynamical system

$$\nabla f(\mathbf{x}^\star) + \frac{\partial g(\mathbf{x}^\star)}{\partial \mathbf{x}}^\top \boldsymbol{\mu}^\star + \frac{\partial h(\mathbf{x}^\star)}{\partial \mathbf{x}}^\top \boldsymbol{\lambda}^\star = \mathbf{0}$$

- Control-affine dynamical system

$$\dot{x} = \underbrace{-\nabla f(\mathbf{x})}_{\text{optimality}} \underbrace{-\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}^\top \boldsymbol{\mu}}_{\text{safety } \text{“}\leqslant\text{”}} \underbrace{-\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}^\top \boldsymbol{\lambda}}_{\text{safety } \text{“}=\text{”}}$$

- The dynamical system strives to the minimizer of f, while the two drift terms steer the trajectory to feasible points

- Some algorithms to design such dynamical systems:
  - Saddle-point flow
  - Projected gradient flow
  - Safe gradient flow

Saddle-point flow

- Equality-constrained non-linear optimization

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) \\ & h(\mathbf{x}) = \mathbf{0} \quad : \boldsymbol{\lambda} \end{aligned}$$

  with Lagrangian function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top h(\mathbf{x})$$

  (Convex in $\mathbf{x}$, concave in $\boldsymbol{\lambda}$)

- The trajectories of the system

$$\dot{\mathbf{x}} = \underbrace{-\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})^\top}_{\text{descent}} \quad \dot{\boldsymbol{\lambda}} = \underbrace{\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})^\top}_{\text{ascent}}$$

  converge to the saddle-point of $\mathcal{L}$

Saddle-point flow - example

- Linearly-constrained optimization

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \quad : \boldsymbol{\lambda} \end{aligned}$$

- The saddle-point flow is described by

$$\begin{aligned} \dot{\mathbf{x}} &= -\nabla f(\mathbf{x})^\top - \mathbf{A}^\top \boldsymbol{\lambda} \\ \dot{\boldsymbol{\lambda}} &= \mathbf{A}\mathbf{x} - \mathbf{b} \end{aligned}$$

- Any equilibrium point $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$ satisfy

$$\mathbf{0} = \nabla f(\mathbf{x}^\star)^\top + \mathbf{A}^\top \boldsymbol{\lambda}^\star, \quad \mathbf{0} = \mathbf{A}\mathbf{x}^\star - \mathbf{b}$$

(Karush-Kuhn-Tucker conditions)

Saddle-point flow - example

- Linearly-constrained optimization

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \quad : \boldsymbol{\lambda} \end{aligned}$$

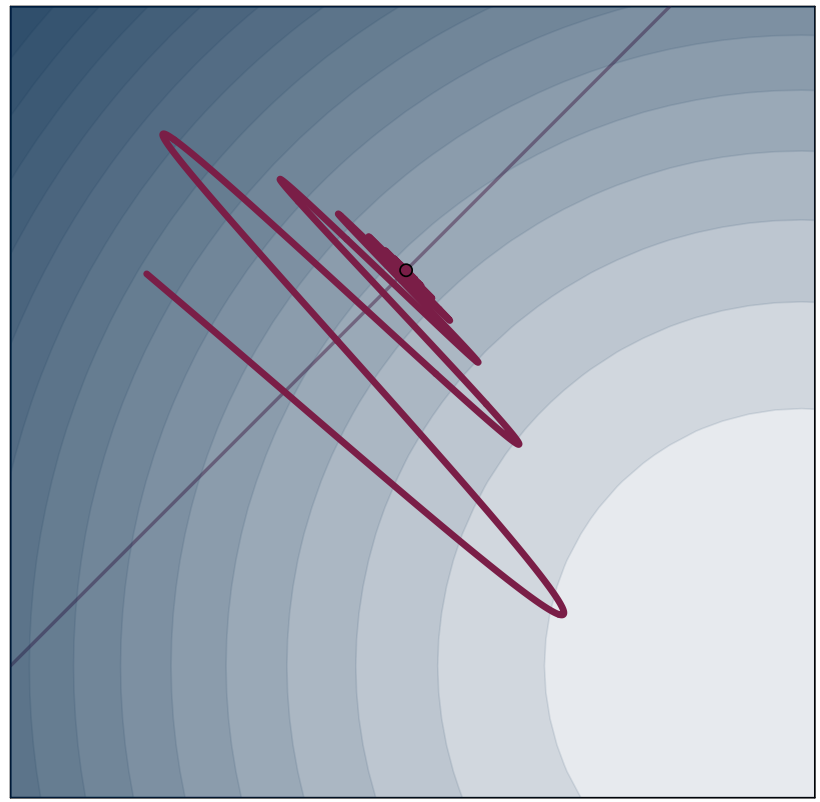
- The saddle-point flow is described by

$$\begin{aligned} \dot{\mathbf{x}} &= -\nabla f(\mathbf{x})^\top - \mathbf{A}^\top \boldsymbol{\lambda} \\ \dot{\boldsymbol{\lambda}} &= \mathbf{A}\mathbf{x} - \mathbf{b} \end{aligned}$$

- Any equilibrium point $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$ satisfy

$$\mathbf{0} = \nabla f(\mathbf{x}^\star)^\top + \mathbf{A}^\top \boldsymbol{\lambda}^\star, \quad \mathbf{0} = \mathbf{A}\mathbf{x}^\star - \mathbf{b}$$

(Karush-Kuhn-Tucker conditions)



$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & 0.125\|\mathbf{x}\|_2^2 - 0.5x_1 + 0.25x_2 \\ \text{subject to} \quad & x_1 - x_2 = 0 \end{aligned}$$

Strengthening saddle-point flow

- What if $f(\mathbf{x})$ is not strictly convex?

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top h(\mathbf{x}) + \frac{\rho}{2} \|h(\mathbf{x})\|_2^2$$

- What if $h(\mathbf{x})$ is non-convex?

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top h(\mathbf{x}) + \frac{\rho}{2} \|\boldsymbol{\lambda}\|_2^2$$

- Add a regularization term to meet certain regularity conditions

Strengthening saddle-point flow

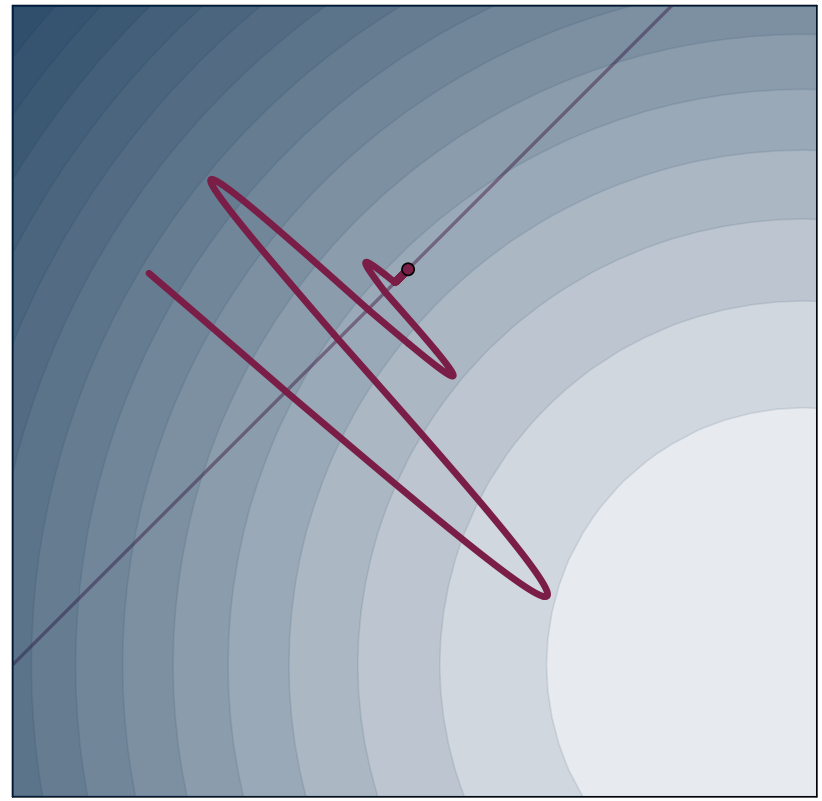
- What if $f(\mathbf{x})$ is not strictly convex?

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top h(\mathbf{x}) + \frac{\rho}{2} \|h(\mathbf{x})\|_2^2$$

- What if $h(\mathbf{x})$ is non-convex?

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top h(\mathbf{x}) + \frac{\rho}{2} \|\boldsymbol{\lambda}\|_2^2$$

- Add a regularization term to meet certain regularity conditions



$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^2}{\text{minimize}} \quad & 0.125\|\mathbf{x}\|_2^2 - 0.5x_1 \\ & + 0.25x_2 + \frac{\rho}{2} \|x_1 - x_2\|_2^2 \\ \text{subject to} \quad & x_1 - x_2 = 0 \end{aligned}$$

Regularization of the objective function enhances convergence

Projected gradient flow

- Non-linear optimization program

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g(\mathbf{x}) \leqslant \mathbf{0} \\ & h(\mathbf{x}) = \mathbf{0} \end{aligned}$$

- The classic projected gradient descent (PGD)

$$\mathbf{x}_{t+1} = P_{\mathcal{C}}\left[\mathbf{x}_t - \eta \nabla f(\mathbf{x})^\top\right]$$

where $P_{\mathcal{C}}[\,\mathbf{y}\,] = \underset{\mathbf{x}}{\text{argmin}}\ \|\mathbf{x}-\mathbf{y}\|_2^2$ is the projection on the feasible region $\mathcal{C}$

- The trajectories of the projected gradient flow (PGF)

$$\dot{x} = P_{\mathcal{C}}\left[-\nabla f(\mathbf{x})^\top\right](x)$$

converge to the optimal solution $\mathbf{x}^\star$

- PGF comes from PGD in the limit as $\eta \to 0$
- PGD is a forward Euler discretization of PGF

Projected gradient flow - example

- Linearly-constrained optimization

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ & \mathbf{A}\mathbf{x} \leqslant \mathbf{b} \end{aligned}$$

- Forward Euler discretization of PGF:

$$\begin{aligned} \widehat{\mathbf{x}} &= \mathbf{x}_t - \eta \nabla f(\mathbf{x})^\top \\ \mathbf{x}_{t+1} &= \underset{\mathbf{x}}{\text{argmin}} \ \|\widehat{\mathbf{x}} - \mathbf{x}\|_2^2 \\ & \text{subject to} \quad \mathbf{A}\mathbf{x} \leqslant \mathbf{b} \end{aligned}$$

- Non-smooth yet feasible trajectory to the optimal solution

Projected gradient flow - example

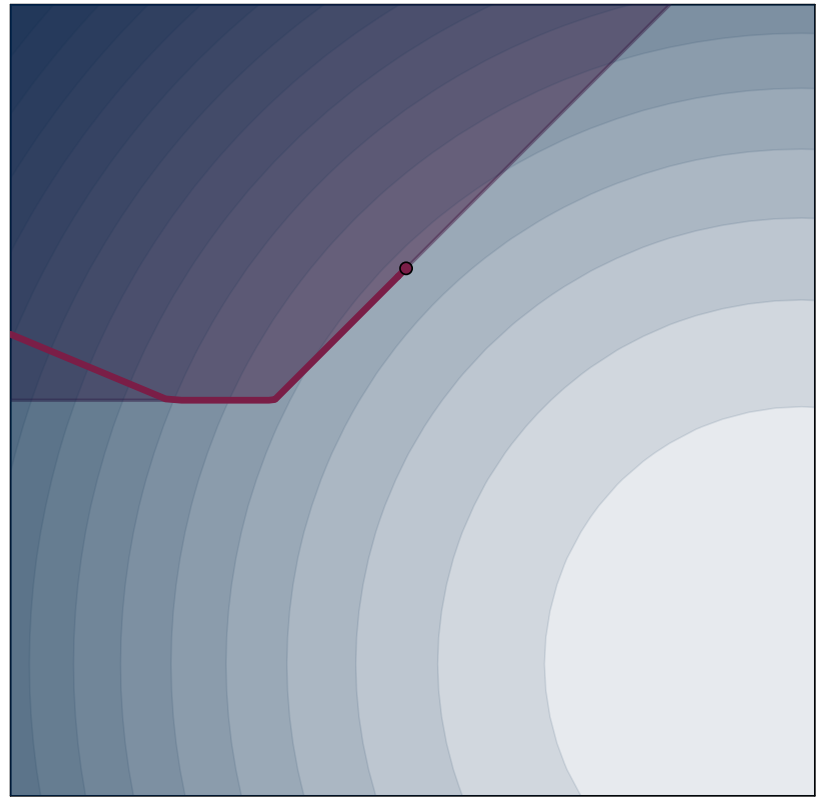
- Linearly-constrained optimization

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ & \mathbf{A}\mathbf{x} \leqslant \mathbf{b} \end{aligned}$$

- Forward Euler discretization of PGF:

$$\begin{aligned} \widehat{\mathbf{x}} &= \mathbf{x}_t - \eta \nabla f(\mathbf{x})^\top \\ \mathbf{x}_{t+1} &= \underset{\mathbf{x}}{\text{argmin}} \ \|\widehat{\mathbf{x}} - \mathbf{x}\|_2^2 \\ & \text{subject to} \quad \mathbf{A}\mathbf{x} \leqslant \mathbf{b} \end{aligned}$$

- Non-smooth yet feasible trajectory to the optimal solution



$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^2}{\text{minimize}} \quad & 0.125\|\mathbf{x}\|_2^2 - 0.5x_1 + 0.25x_2 \\ \text{subject to} \quad & x_1 - x_2 \leqslant 0 \\ & x_2 \geqslant 0 \end{aligned}$$

Projected saddle-point flow (mixed saddle flow)

- Non-linear optimization

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X} \\ & g(\mathbf{x}) \leqslant \mathbf{0} \quad : \boldsymbol{\mu} \end{aligned}$$

  with partial Lagrangian function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\mu}^\top g(\mathbf{x})$$

- The trajectories of the system

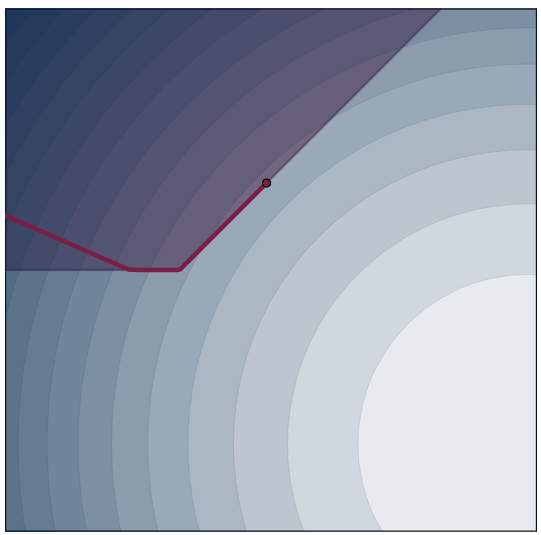
$$\dot{\mathbf{x}} = P_{\mathcal{X}}\big[\underbrace{-\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \boldsymbol{\mu})^\top}_{-\nabla f(\mathbf{x})^\top - \nabla g(\mathbf{x})^\top \boldsymbol{\mu}}\big] \quad \dot{\boldsymbol{\mu}} = P_{\mathbb{R}^m_+}\big[\underbrace{\nabla_{\boldsymbol{\mu}}\mathcal{L}(\mathbf{x}, \boldsymbol{\mu})^\top}_{g(\mathbf{x})}\big]$$

  converge to the saddle-point of the full Lagrangian function

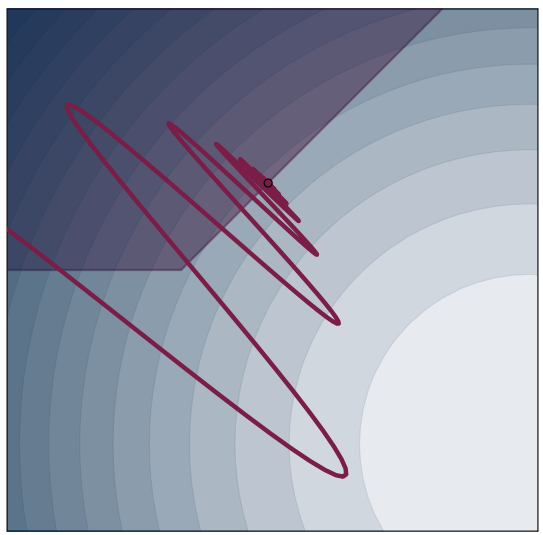
Projected saddle-point flow - example

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^2}{\text{minimize}} \quad & 0.125\|\mathbf{x}\|_2^2 - 0.5x_1 + 0.25x_2 \\ \text{subject to} \quad & x_2 \geqslant x_1 \qquad \text{(dualize)} \\ & x_2 \geqslant 0 \qquad \text{(project)} \end{aligned}$$

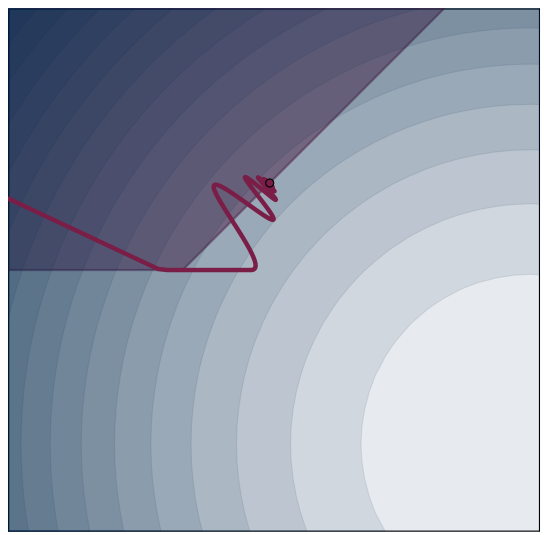
Projected gradient flow



Saddle-point flow



Projected saddle-point flow



Safe gradient flow

- The saddle-point flow is smooth but may lead to infeasible intermediate states
- The projected gradient flow is non-smooth but remains inside the feasible region
- Safe gradient flow enjoys the best of the two flows

- Control-affine dynamical system

$$\dot{x} = \underbrace{-\nabla f(\mathbf{x})}_{\text{optimality}} \underbrace{-\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}^{\top} \boldsymbol{\mu}}_{\text{safety } \text{“}\leqslant\text{”}} \underbrace{-\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}^{\top} \boldsymbol{\lambda}}_{\text{safety } \text{“}=\text{”}} \tag{1}$$

- At every step, select duals $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ by solving an optimization

$$\begin{bmatrix} \boldsymbol{\mu}(\mathbf{x}) \\ \boldsymbol{\lambda}(\mathbf{x}) \end{bmatrix} \in \underset{\boldsymbol{\mu}, \boldsymbol{\lambda} \in K_\alpha(\mathbf{x})}{\operatorname{argmin}} \left\| \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}^{\top} \boldsymbol{\mu} + \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}^{\top} \boldsymbol{\lambda} \right\|_2^2$$

  where $K_\alpha(\mathbf{x})$ is the admissible control set
- “Minimizes the drift (to ensure optimality), while maintaining feasibility”:
  - Trajectories are feasible if they start from a feasible point
  - Trajectories starting from infeasible points converge to a feasible point

Safe gradient flow (cont'd)

- The admissible control set is inspired by the theory of control barrier functions[1]
- Problem constraints define a valid control barrier function
- The resultant admissible control set is

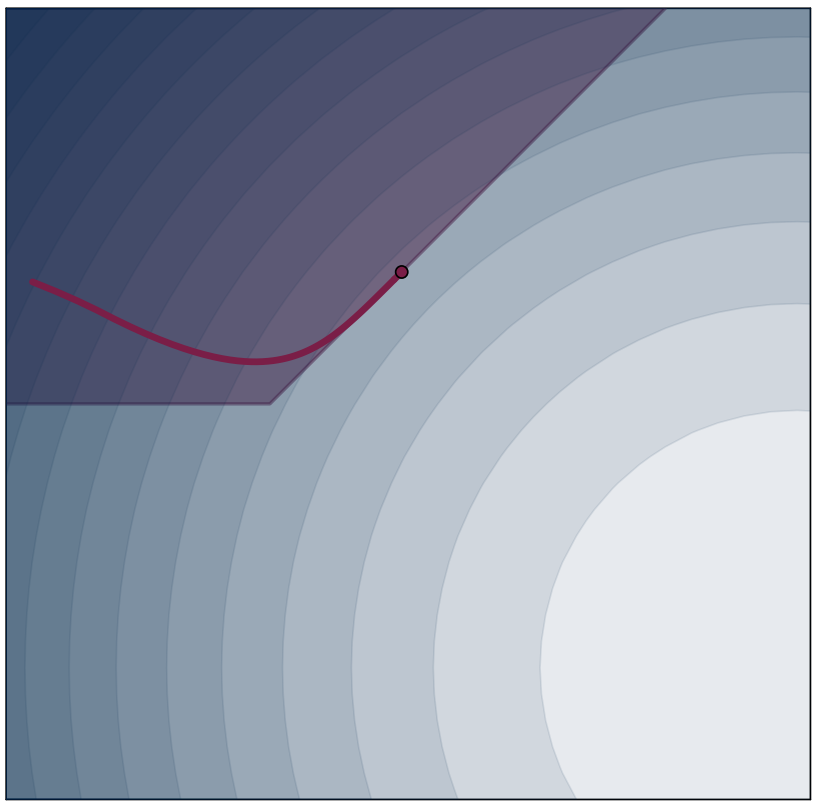
$$K_\alpha(\mathbf{x}) = \Bigg\{ (\boldsymbol{\mu}, \boldsymbol{\lambda}) \in \mathbb{R}^m_+ \times \mathbb{R}^k \;\Bigg|\; \begin{aligned} &-\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}^\top \boldsymbol{\mu} - \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}^\top \boldsymbol{\lambda} \leqslant \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \nabla f(\mathbf{x}) - \alpha g(\mathbf{x}) \\ &-\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}^\top \boldsymbol{\mu} - \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}^\top \boldsymbol{\lambda} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \nabla f(\mathbf{x}) - \alpha h(\mathbf{x}) \end{aligned} \Bigg\}$$

[1] Allibhoy & Cortés. Control barrier function-based design of gradient flows for constrained nonlinear programming. IEEE Transactions on Automatic Control. 2023

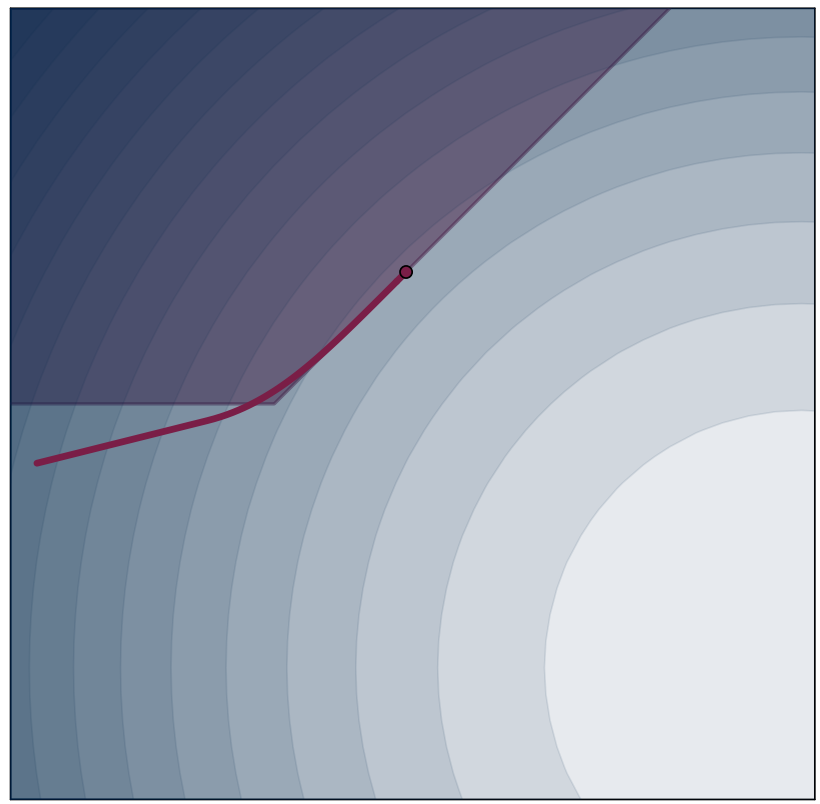
Safe gradient flow - example

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^2}{\text{minimize}} \quad & 0.125\|\mathbf{x}\|_2^2 - 0.5x_1 + 0.25x_2 \\ \text{subject to} \quad & x_1 - x_2 \leqslant 0 \\ & x_2 \geqslant 0 \end{aligned}$$

Feasible initialization



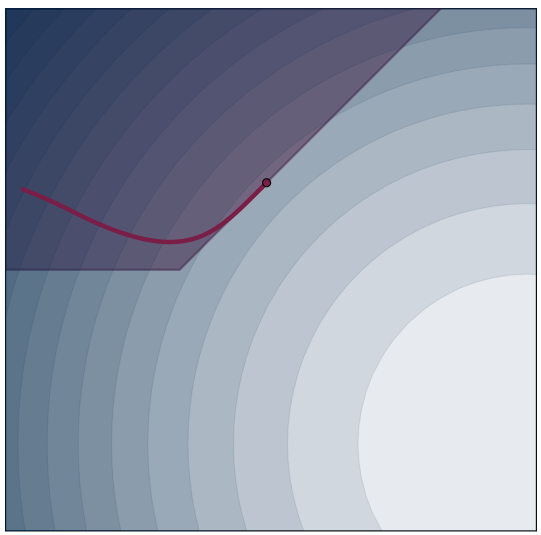
Infeasible initialization



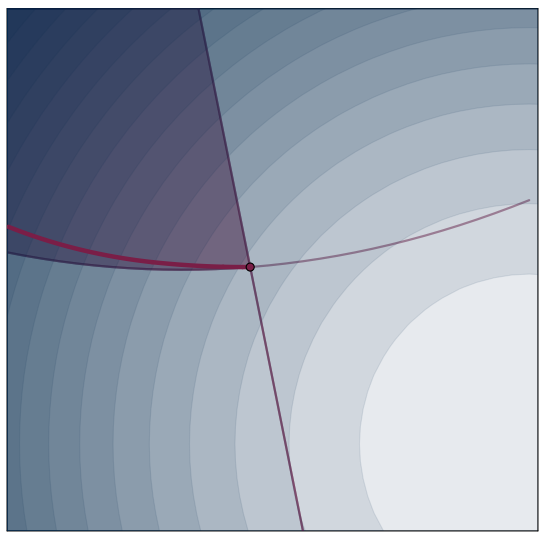
Safe gradient flow - example (cont'd)

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^2}{\text{minimize}} \quad & 0.125\|\mathbf{x}\|_2^2 - 0.5x_1 + 0.25x_2 \\ \text{subject to} \quad & \end{aligned}$$

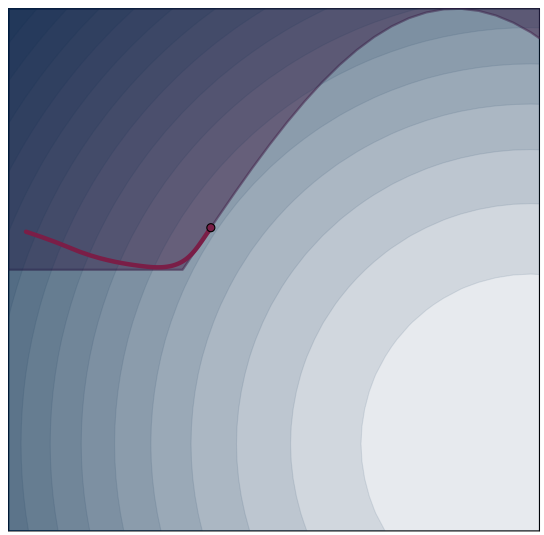
Linear constraints



Non-linear convex constraints



Non-linear non-convex constraints



Summary

- Optimization algorithms serve as robust feedback controllers
- AC-OPF is amenable to online algs: tracking optimal solutions under time-varying parameters while–depending on the alg–ensuring optimality and feasibility
- Online optimization algorithms:
  - Saddle-point flow: smooth tracking with no feas guarantees
  - Projected gradient flow: non-smooth tracking with feas guarantees
  - Safe gradeint flow: smooth tracking with feas guarantees

Next time up: applications of online optimization to power systems

Resources

Online optimization algorithms:

- Hauswirth, Adrian, et al. "Optimization algorithms as robust feedback controllers." Annual Reviews in Control 57 (2024): 100941.

- Allibhoy, Ahmed, and Jorge Cortés. "Control-barrier-function-based design of gradient flows for constrained nonlinear programming." IEEE Transactions on Automatic Control 69.6 (2023): 3499-3514.

Applications to OPF:

- Ortmann, L., Hauswirth, A., Caduff, I., Dörfler, F., & Bolognani, S. (2020). Experimental validation of feedback optimization in power distribution grids. Electric Power Systems Research, 189, 106782.

- Bolognani, S., Carli, R., Cavraro, G., & Zampieri, S. (2014). Distributed reactive power feedback control for voltage regulation and loss minimization. IEEE Transactions on Automatic Control, 60(4), 966-981.

- Dall'Anese, E., & Simonetto, A. (2016). Optimal power flow pursuit. IEEE Transactions on Smart Grid, 9(2), 942-952.