Real-time Optimization of Distributed Energy Resources

Yujie Tang
Department of Electrical Engineering
California Institute of Technology

Caltech

Communication vs. Power Networks

Communication Networks

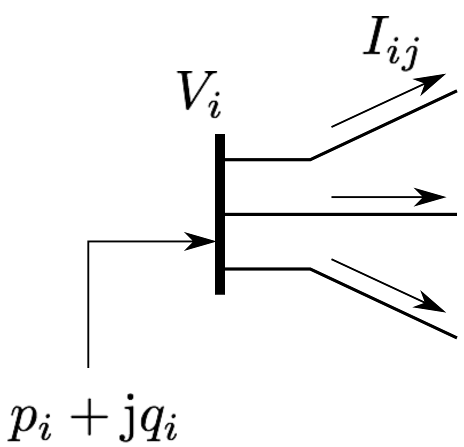
- Transmit **information** through **communication links**
- **Sources** and **sinks**

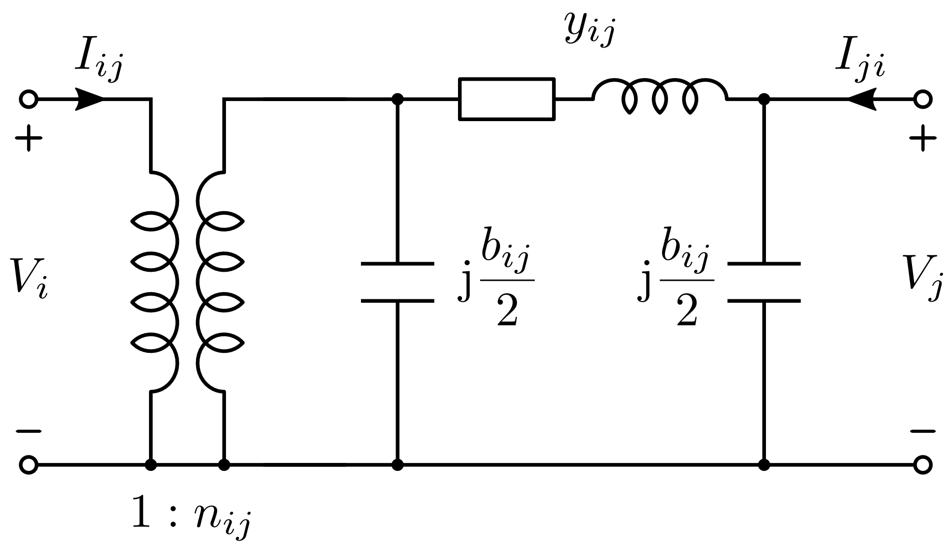
Power Networks

- Transmit **power** through **power lines** and **transformers**
- **Generators** and **loads**

Physics of Power Networks

- Kirchhoff's laws + Ohm's law





$$p_i + \mathrm{j}q_i = V_i \sum_{j \in \mathcal{N}_i} I_{ij}^*$$

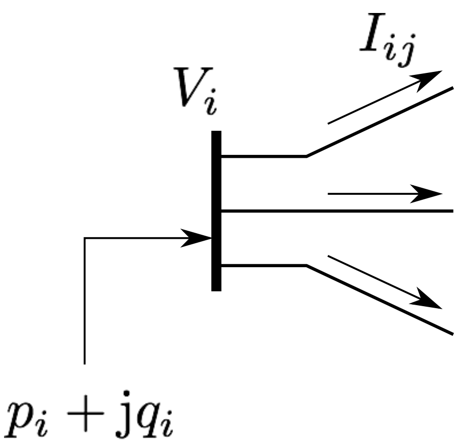
$$\begin{bmatrix} I_{ij} \\ I_{ji} \end{bmatrix} = \begin{bmatrix} \left(y_{ij} + \mathrm{j}\frac{b_{ij}}{2}\right) n_{ij}^2 & -y_{ij} n_{ij}^* \\ -y_{ij} n_{ij} & y_{ij} + \mathrm{j}\frac{b_{ij}}{2} \end{bmatrix} \begin{bmatrix} V_i \\ V_j \end{bmatrix}$$

Physics of Power Networks

- Kirchhoff's laws + Ohm's law
- Power flow equations

$$\begin{bmatrix} I_{ij} \\ I_{ji} \end{bmatrix} = \begin{bmatrix} \left(y_{ij} + \mathrm{j}\frac{b_{ij}}{2}\right) n_{ij}^2 & -y_{ij} n_{ij}^* \\ -y_{ij} n_{ij} & y_{ij} + \mathrm{j}\frac{b_{ij}}{2} \end{bmatrix} \begin{bmatrix} V_i \\ V_j \end{bmatrix}$$

$$p_i + \mathrm{j}q_i = V_i \sum_{j \in \mathcal{N}_i} I_{ij}^*$$



  - Nonlinear and nonconvex
  - Local perturbations may result in global effects

Optimal Power Flow (OPF)

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \text{cost}(x) \\ \text{subject to} \quad & \text{power flow equations} \\ & \text{demands are satisfied} \\ & V_i^{\min} \le |V_i| \le V_i^{\max} \\ & |I_{ij}| \le I_{ij}^{\max} \\ & x \in \mathcal{X} \end{aligned}$$

- Nonlinear and nonconvex
- Finding global optimum is NP hard in general

OPF Applications

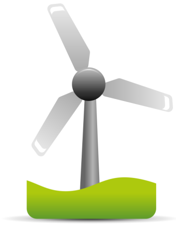
- volt/var control
- state estimation
- economic dispatch
- unit commitment
- stability and reliability assessment
- real-time DER control
- microgrid operation
- load-side regulation
- EV / storage scheduling
- DER planning

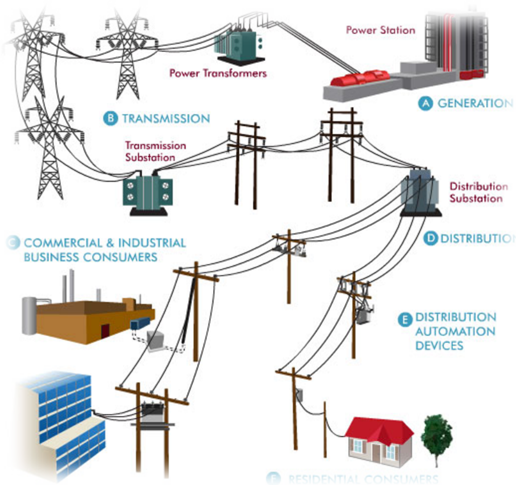
Timescales span from ***seconds*** *to* ***years***

Time-varying Optimization
and Its Application to Real-time OPF

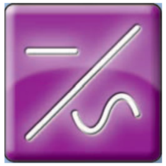
DERs in Power Networks

fluctuations, intermittency





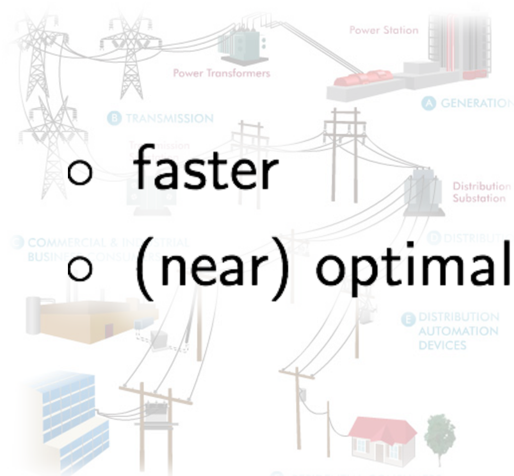
diverse control capabilities



extensive measurements

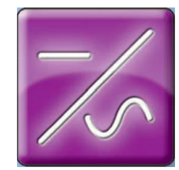
DERs in Power Networks

fluctuations, intermittency



- faster
- (near) optimal

diverse control capabilities



extensive measurements

Optimal Power Flow (OPF)

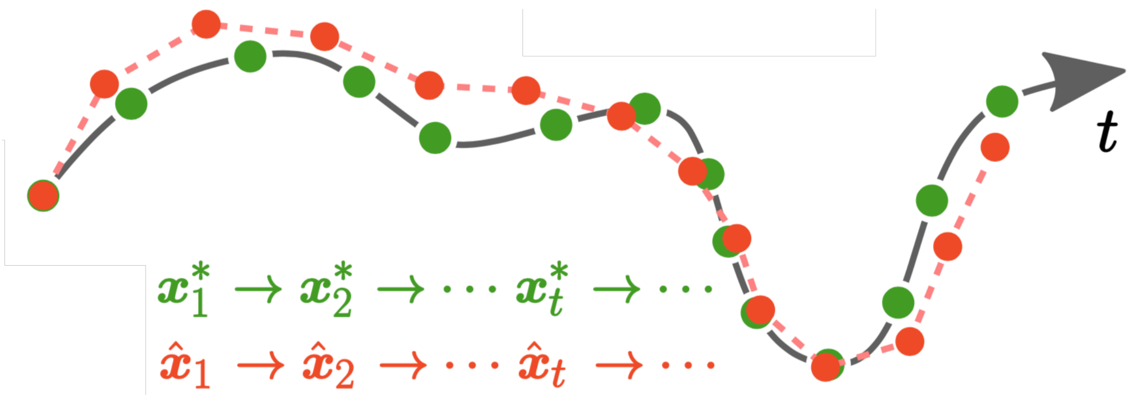
$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & \text{cost}(x) \\
\text{subject to} \quad & \text{power flow equations} \\
& \text{demands are satisfied} \\
& V_i^{\min} \leq |V_i| \leq V_i^{\max} \\
& |I_{ij}| \leq I_{ij}^{\max} \\
& x \in \mathcal{X}
\end{aligned}
$$

- Why not solve directly (batch/offline solution)?
  - Slow for systems with a large number of decision variables
  - The true optimum may have changed a lot after a batch solution has been found

Time-varying Optimization

- Technique for tracking optimal solution that evolves with time
  - optimal trajectory: $(x_t^*)_{t \in \mathcal{T}}$

$$\begin{aligned} \min_x \quad & c_t(x) \\ \text{s.t.} \quad & f_t(x) \le 0 \\ & x \in \mathcal{X}_t \end{aligned}$$



Time-varying Optimization

- Technique for tracking optimal solution that evolves with time
- Why not solve directly?
  - Can be slow for large systems
  - The true optimum may have changed a lot after a batch solution has been found
  - Feedback measurements can be naturally incorporated and better utilized in time-varying optimization

$$\begin{aligned} \min_{x} \quad & c_t(x) \\ \text{s.t.} \quad & f_t(x) \leq 0 \\ & x \in \mathcal{X}_t \end{aligned}$$

Time-varying Optimization

- J. Guddat, et al. *Parametric Optimization: Singularities, Pathfollowing and Jumps*, 1990.
- A. Y. Popkov. “Gradient methods for nonstationary unconstrained optimization problems”, 2005.
- Q. Ling and A. Ribeiro. "Decentralized dynamic optimization through the alternating direction method of multipliers”, 2013.
- A. Simonetto and G. Leus. “Double smoothing for time-varying distributed multiuser optimization”, 2014.
- C. Xi and U. A. Khan. "Distributed dynamic optimization over directed graphs”, 2016.
- A. Simonetto. “Time-varying convex optimization via time-varying averaged operators”, 2017.

Time-varying Optimization

- E. Dall'Anese and A. Simonetto. “Optimal power flow pursuit”, 2018.
- J. Chen and V. K. N. Lau. “Convergence analysis of saddle point problems in time varying wireless systems - control theoretical approach”, 2012.
- A. Balavoine, et al. “Discrete and continuous-time soft-thresholding with dynamic inputs”, 2014.

- A. Mokhtari, et al. "Online optimization in dynamic environments: Improved regret rates for strongly convex problems”, 2016.
- T. Yang, et al. “Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient”, 2016.

Time-varying Optimization

- Static optimization

$$x_k = T(x_{k-1}; c(\cdot), f(\cdot), \mathcal{X})$$

$$x^* = \lim_{k \to \infty} x_k$$

$$\begin{aligned} \min_x \quad & c(x) \\ \text{s.t.} \quad & f(x) \leq 0 \\ & x \in \mathcal{X} \end{aligned}$$

- Time-varying optimization
  - The problem itself is changing
  - Update the problem data at the same time as we run the iterations

$$\begin{aligned} \min_x \quad & c_t(x) \\ \text{s.t.} \quad & f_t(x) \leq 0 \\ & x \in \mathcal{X}_t \end{aligned}$$

$$\hat{x}_t = T(\hat{x}_{t-1}; c_t(\cdot), f_t(\cdot), \mathcal{X}_t)$$

Time-varying Optimization

- Example: projected gradient method
- Static optimization:

$$x_k = \mathcal{P}_{\mathcal{X}}\left[x_{k-1} - \alpha \nabla c(x_{k-1})\right]$$

- Time-varying optimization:

$$\hat{x}_t = \mathcal{P}_{\mathcal{X}_t}\left[\hat{x}_{t-1} - \alpha \nabla c_t(\hat{x}_{t-1})\right]$$

Time-varying Optimization

- Performance metric: **tracking error**
- Distance to the true optimal

$$\|\hat{x}_t - x_t^*\|$$

- Merit function value difference

$$\phi_t(\hat{x}_t) - \phi_t(x_t^*)$$

- Residual

$$\|\hat{x}_t - T(\hat{x}_t; c_t(\cdot), f_t(\cdot), \mathcal{X}_t)\|$$

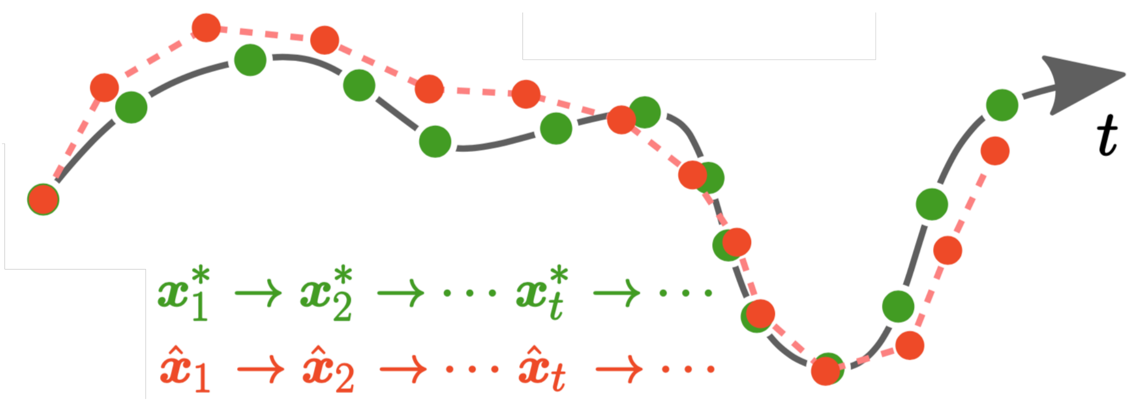
$$O\left(\sup_t \|x_t^* - x_{t-1}^*\|\right)$$

how fast the optimal solution moves

Time-varying Optimization

- Performance metric: **tracking error**
- **Distance to the true optimal**

$$\|\hat{x}_t - x_t^*\| = O\left(\sup_t \|x_t^* - x_{t-1}^*\|\right)$$



Time-varying Optimization

- Performance metric: **tracking error**
- **Distance to the true optimal**

$$\|\hat{x}_t - x_t^*\| = O\left(\sup_t \|x_t^* - x_{t-1}^*\|\right)$$

- A general observation:

Linear convergence in static optimization

$$\Downarrow$$

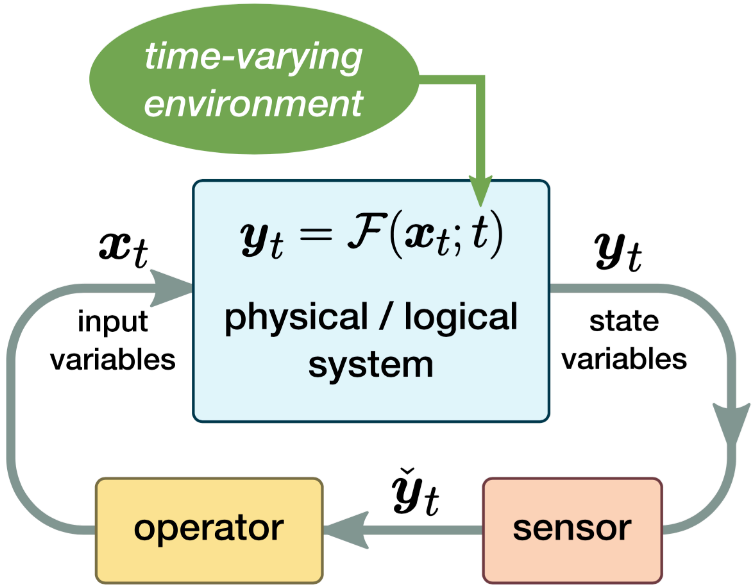
Bounded tracking error in time-varying optimization

Adding Feedback

- Time-varying optimization
  - The problem itself is changing
  - Update the problem data at the same time as we run the iterations

$$\hat{x}_t = T(\hat{x}_{t-1}; c_t(\cdot), f_t(\cdot), \mathcal{X}_t)$$

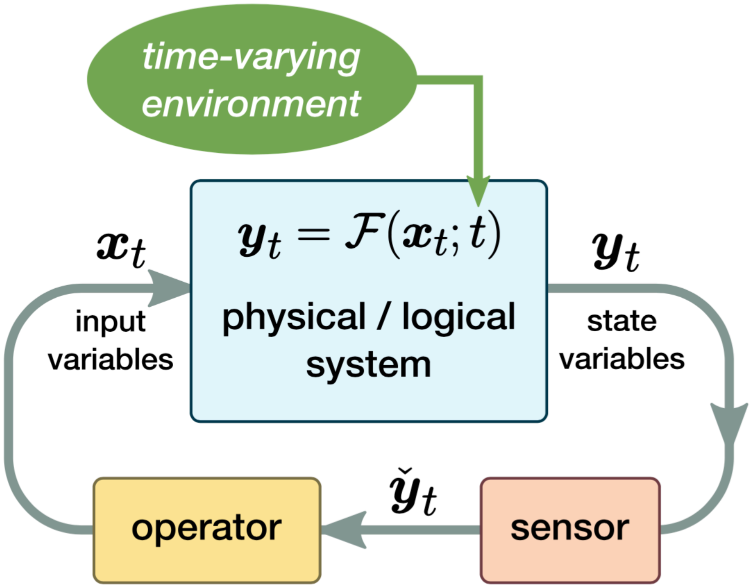
  - Apply $\hat{x}_t$ to the system.
    Then the system will provide y_t automatically.
  - Real-time measurements can be naturally incorporated.
  - Potentially more robust to model mismatch.



Application to OPF

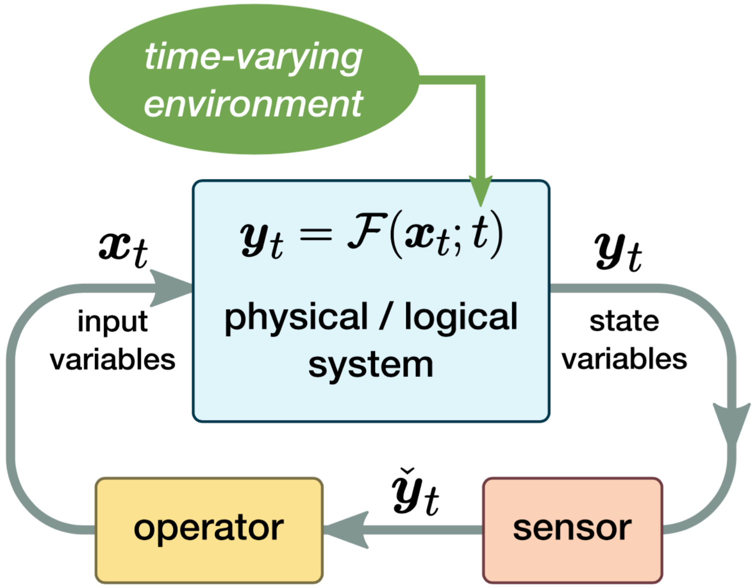
- Input variables:
  power injections of DERs
- State variables:
  voltages (magnitude + phase)
- System behavior:
  power flow equations

$$\mathcal{G}(x, y; t) = 0$$



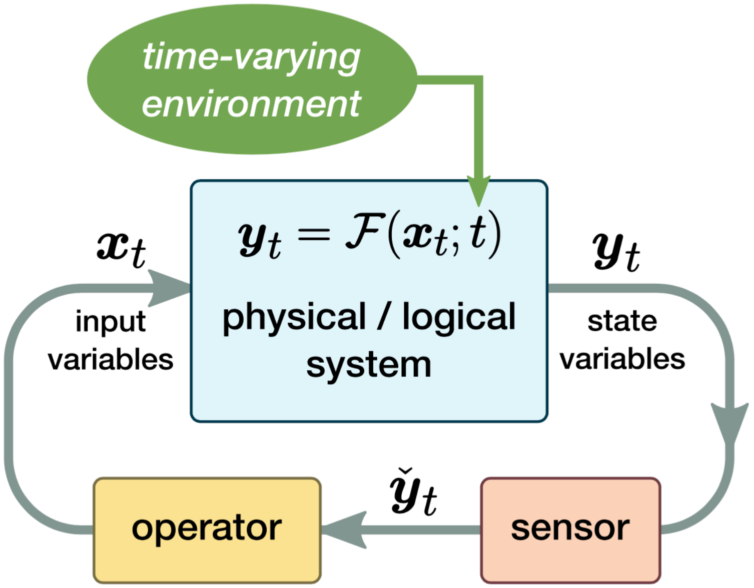
Application to OPF

- Input variables:
  power injections of DERs
- State variables:
  voltages (magnitude + phase)
- System behavior:
  power flow equations
  $$y = \mathcal{F}(x; t)$$
- Constraints:
  - voltage & current limits: $h_t(\mathcal{F}(x; t)) \leq 0$
  - power injection capacities of DERs: $x \in \mathcal{X}_t$



Application to OPF

$$\begin{aligned} \min_{x} \quad & c_t(x) \\ \text{s.t.} \quad & h_t(\mathcal{F}(x;t)) \le 0 \\ & x \in \mathcal{X}_t \end{aligned}$$



- Decision variable:
  power injections of DERs
- $\mathcal{F}(x;t)$ – power flow "function"
- $h_t(\mathcal{F}(x;t)) \le 0$ – voltage & current limits
- $x \in \mathcal{X}_t$ – capacities of DERs (convex)

Application to OPF

- Primal method
  - second-order
  - implementation based on L-BFGS
- Primal-dual method
  - first-order
  - regularized Lagrangian
  - distributed implementation

Real-time OPF

Second-order Primal Method

Second-order Primal Method

- Based on (quasi-)Newton method
- Primal method: Penalty functions are introduced

$$Q_t(x) := c_t(x) + \mu \sum_i \varphi_i(h_{t,i}(\mathcal{F}(x;t)))$$

- Quadratic approximation of $Q_t(x)$

$$\tilde{Q}_t(x; x_0, B) := \nabla Q_t(x_0)^T (x - x_0) + \frac{1}{2}(x - x_0)^T B (x - x_0)$$

Caltech

Second-order Primal Method

For each time t

1. Measure input & state variables and $\mathcal{X}_t$
2. Compute $\nabla c_t(\hat{x}_{t-1})$ from measurement data
3. Find

$$\hat{x}_t = \arg\min_{x \in \mathcal{X}_t} \tilde{Q}_t(x; \hat{x}_{t-1}, B_t)$$

4. Apply $\hat{x}_t$ to the power system
5. Update B_{t+1}

Second-order Primal Method

Different choice of B_t specifies a different algorithm

- $B_t \approx \nabla^2 Q_t$: approximate / quasi-Newton method
- $B_t = \eta_t I$: projected gradient method

Second-order Primal Method

Theorem [Tang, Dvijotham and Low, 2017]

Under certain conditions, if B_t's are positive definite and satisfy

$$\frac{\|B_t^{-1}(\nabla Q_t(x_t^*) - \nabla Q_t(\hat{x}_{t-1})) - (x_t^* - \hat{x}_{t-1})\|_{B_t}}{\|x_t^* - \hat{x}_{t-1}\|_{B_t}} \leq \epsilon < \sqrt{\frac{\lambda_m}{\lambda_M}}$$

Then

$$\|\hat{x}_t - x_t^*\| \leq \frac{\epsilon}{\sqrt{\lambda_m/\lambda_M} - \epsilon} \sup_t \|x_t^* - x_{t-1}^*\| \qquad \forall t$$

when $\|\hat{x}_0 - x_0^*\|$ satisfies this inequality.

Second-order Primal Method

Theorem [Tang, Dvijotham and Low, 2017]

Under certain conditions, if B_t's are positive definite and satisfy

$$\nabla Q_t(x_t^*) - \nabla Q_t(\hat{x}_{t-1}) \approx B_t(x_t^* - \hat{x}_{t-1})$$

Then

$$\|\hat{x}_t - x_t^*\| \leq \frac{\epsilon}{\sqrt{\lambda_m/\lambda_M} - \epsilon} \sup_t \|x_t^* - x_{t-1}^*\| \quad \forall t$$

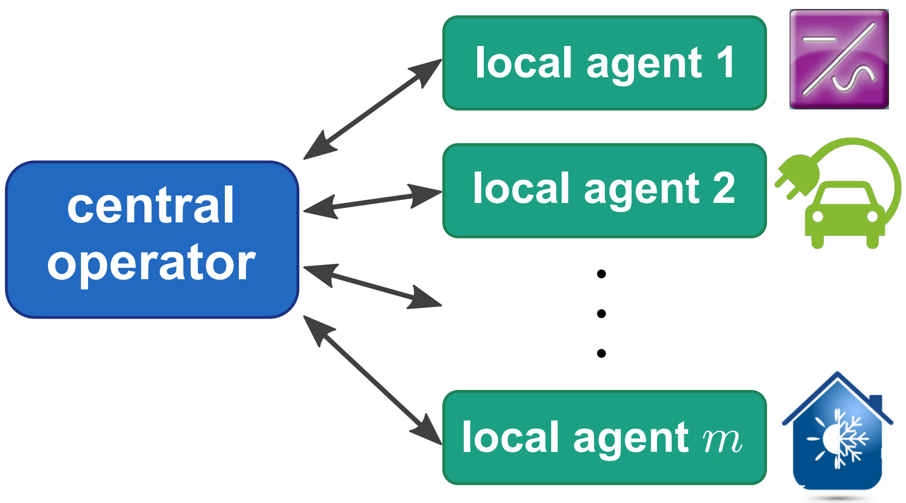
inverse of "condition number"

approximation accuracy

when $\|\hat{x}_0 - x_0^*\|$ satisfies this inequality.

- x_t^* denotes the optimum of the penalized OPF.
- A sufficient condition for such B_t to exist is that Q_t is locally strongly convex around x_t^*.
- Can be generalized when there is measurement noise / model mismatch.

Implementation Based on L-BFGS



- central operator
  - measure state variables
  - some centralized computation
- local agents
  - operate & measure controllable devices
  - local computation

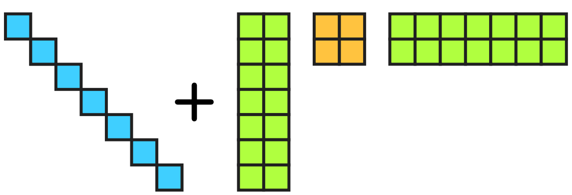
Y. Tang, K. Dvijotham and S. Low, "Real-time optimal power flow", 2017.
Y. Tang and S. Low, "Distributed algorithm for time-varying optimal power flow", 2017.

Implementation Based on L-BFGS

Central operator

- compute $\nabla Q_t(\hat{x}_{t-1})$
  - from measurement data
- compute approximate Hessian
  - L-BFGS:

$$B_t = \theta_t I - K_t M_t K_t^T$$

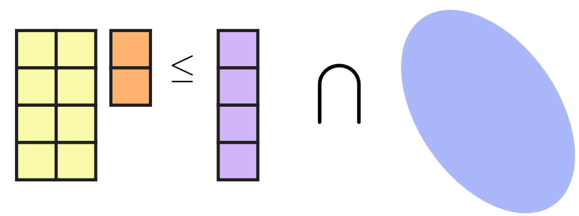


Local agents

$$x = (x_1, \ldots, x_m)$$

$$x_i \in \mathcal{X}_{i,t} \subseteq \mathbb{R}^{n_i}$$

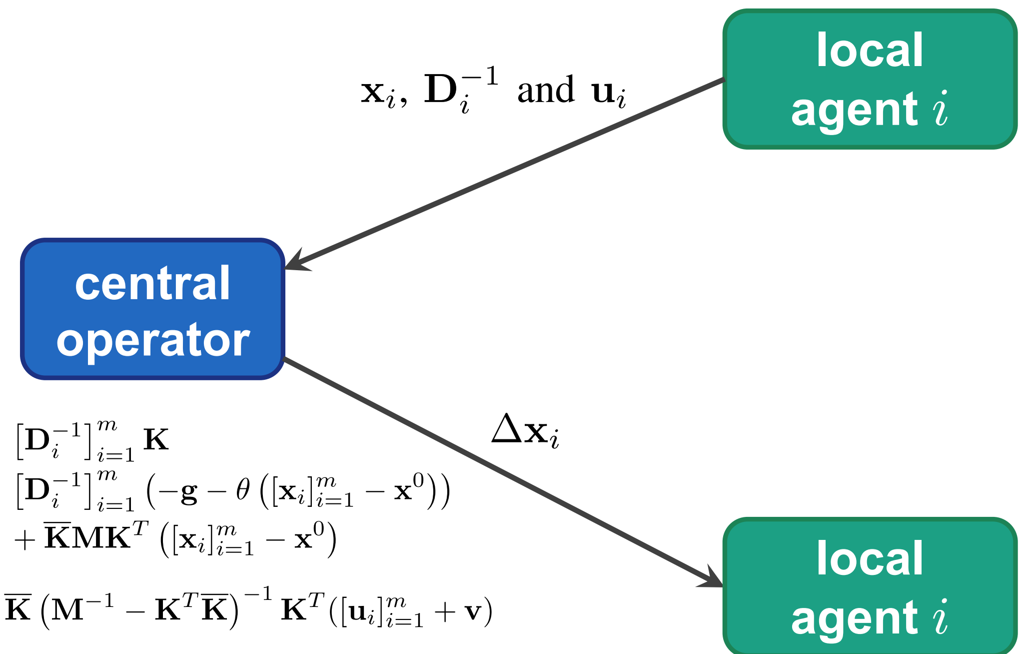
$$\mathcal{X}_{i,t} =$$



$$\hat{x}_t = \arg\min_{x \in \mathcal{X}_t} \tilde{Q}_t(x; \hat{x}_{t-1}, B_t)$$

Y. Tang, K. Dvijotham and S. Low, “Real-time optimal power flow”, 2017.
Y. Tang and S. Low, “Distributed algorithm for time-varying optimal power flow”, 2017.

Implementation Based on L-BFGS



$$\mathbf{S}_i = \mathrm{diag}(\mathbf{b}_i - \mathbf{A}_i \mathbf{x}_i)$$

$$z_i = -\left(\frac{1}{2}\mathbf{x}_i^T \mathbf{Q}_i \mathbf{x}_i + \mathbf{p}_i^T \mathbf{x}_i + r_i\right)$$

$$\mathbf{D}_i^{-1} \leftarrow \Big(\theta \mathbf{I} + \nu_i \mathbf{Q}_i + \mathbf{A}_i^T \mathbf{S}_i^{-1}\,\mathrm{diag}(\boldsymbol{\lambda}_i)\mathbf{A}_i + z_i^{-1}\nu_i(\mathbf{Q}_i\mathbf{x}_i + \mathbf{p}_i)(\mathbf{Q}_i\mathbf{x}_i + \mathbf{p}_i)^T\Big)^{-1}$$

$$\mathbf{u}_i \leftarrow -\epsilon_i \mathbf{D}_i^{-1}\left(\mathbf{A}_i^T \mathbf{S}_i^{-1}\mathbf{1} + z_i^{-1}(\mathbf{Q}_i\mathbf{x}_i + \mathbf{p}_i)\right)$$

$$\overline{\mathbf{K}} \leftarrow \left[\mathbf{D}_i^{-1}\right]_{i=1}^m \mathbf{K}$$

$$\mathbf{v} \leftarrow \left[\mathbf{D}_i^{-1}\right]_{i=1}^m \left(-\mathbf{g} - \theta\left([\mathbf{x}_i]_{i=1}^m - \mathbf{x}^0\right)\right) + \overline{\mathbf{K}}\mathbf{M}\mathbf{K}^T\left([\mathbf{x}_i]_{i=1}^m - \mathbf{x}^0\right)$$

$$\overline{\mathbf{v}} \leftarrow \overline{\mathbf{K}}\left(\mathbf{M}^{-1} - \mathbf{K}^T\overline{\mathbf{K}}\right)^{-1}\mathbf{K}^T([\mathbf{u}_i]_{i=1}^m + \mathbf{v})$$

$$\Delta \mathbf{x} \leftarrow [\mathbf{u}_i]_{i=1}^m + \mathbf{v} + \overline{\mathbf{v}}$$

$$\Delta\boldsymbol{\lambda}_i \leftarrow \mathbf{S}_i^{-1}(\epsilon_i \mathbf{1} + \mathrm{diag}(\boldsymbol{\lambda}_i)\mathbf{A}_i\Delta\mathbf{x}_i) - \boldsymbol{\lambda}_i$$

$$\Delta\nu_i \leftarrow z_i^{-1}\left(\epsilon_i + \nu_i(\mathbf{Q}_i\mathbf{x}_i + \mathbf{p}_i)^T\Delta\mathbf{x}_i\right) - \nu_i$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \alpha_i \Delta\mathbf{x}_i$$

$$\boldsymbol{\lambda}_i \leftarrow \boldsymbol{\lambda}_i + \alpha_i \Delta\boldsymbol{\lambda}_i$$

$$\nu_i \leftarrow \nu_i + \alpha_i \Delta\nu_i$$

Y. Tang, K. Dvijotham and S. Low, “Real-time optimal power flow”, 2017.
Y. Tang and S. Low, “Distributed algorithm for time-varying optimal power flow”, 2017.

Implementation Based on L-BFGS

- Fast convergence
- Low computation complexity
- Requires minimal communication
- Local agents do not report $x_{i,t}$ directly to central operator

- Suitable for situations where measurements are relatively "slow".

Y. Tang, K. Dvijotham and S. Low, "Real-time optimal power flow", 2017.
Y. Tang and S. Low, "Distributed algorithm for time-varying optimal power flow", 2017.

Real-time OPF

First-order Primal-dual Method

First-order Primal-dual Method

- Based on primal-dual gradient method
- Lagrangian

$$L_t(x, \lambda) = c_t(x) + \lambda^T h_t(\mathcal{F}(x;t))$$

$$\begin{aligned} \min_x \quad & c_t(x) \\ \text{s.t.} \quad & h_t(\mathcal{F}(x;t)) \leq 0 \\ & x \in \mathcal{X}_t \end{aligned}$$

First-order Primal-dual Method

- Based on primal-dual gradient method
- Regularized Lagrangian

$$L_t(x, \lambda) = c_t(x) + \lambda^T h_t(\mathcal{F}(x;t)) - \frac{\epsilon}{2}\|\lambda\|^2$$

$$\begin{aligned} \min_x \quad & c_t(x) \\ \text{s.t.} \quad & h_t(\mathcal{F}(x;t)) \le 0 \\ & x \in \mathcal{X}_t \end{aligned}$$

- Why regularization?
  - $L_t(x, \lambda)$ strongly concave in λ
  - Important property for establishing tracking error bound
  - Introduces additional error term in tracking error bound

First-order Primal-dual Method

- Based on primal-dual gradient method
- Regularized Lagrangian

$$L_t(x,\lambda) = c_t(x) + \lambda^T h_t(\mathcal{F}(x;t)) - \frac{\epsilon}{2}\|\lambda\|^2$$

$$\begin{aligned} \min_x \quad & c_t(x) \\ \text{s.t.} \quad & h_t(\mathcal{F}(x;t)) \le 0 \\ & x \in \mathcal{X}_t \end{aligned}$$

- Regularized primal-dual gradient method:

$$\hat{x}_t = \mathcal{P}_{\mathcal{X}_t}\left[\hat{x}_{t-1} - \alpha \nabla_x L_t(\hat{x}_{t-1}, \hat{\lambda}_{t-1})\right]$$

$$\hat{\lambda}_t = \mathcal{P}_{\mathbb{R}^p_+}\left[\hat{\lambda}_{t-1} + \beta \nabla_\lambda L_t(\hat{x}_{t-1}, \hat{\lambda}_{t-1})\right]$$

First-order Primal-dual Method

For each time t

1. Measure input & state variables and $\mathcal{X}_t$

   $\check{y}_t =$ measured value of state variables

$$\begin{aligned} \min_x \quad & c_t(x) \\ \text{s.t.} \quad & h_t(\mathcal{F}(x;t)) \leq 0 \\ & x \in \mathcal{X}_t \end{aligned}$$

2. Find

$$\hat{x}_t = \mathcal{P}_{\mathcal{X}_t}\left[\hat{x}_{t-1} - \alpha\left(\nabla c_t(\hat{x}_{t-1}) + (H_t(\check{y}_t)J_t(\hat{x}_{t-1}, \check{y}_t))^T\hat{\lambda}_{t-1}\right)\right]$$

$$\hat{\lambda}_t = \mathcal{P}_{\mathbb{R}^p_+}\left[(1-\beta\epsilon)\hat{\lambda}_{t-1} + \beta h_t(\check{y}_t)\right]$$

3. Apply $\hat{x}_t$ to the power system

First-order Primal-dual Method

Theorem [Tang, Dall'Anese, Bernstein and Low, 2018]

Let (x_t^*, λ_t^*) be a trajectory of local optimal solution.
Suppose $L_t(x, \lambda)$ is sufficiently locally strongly convex in x around (x_t^*, λ_t^*) for each t. Then there exist parameters α, β, ϵ such that when

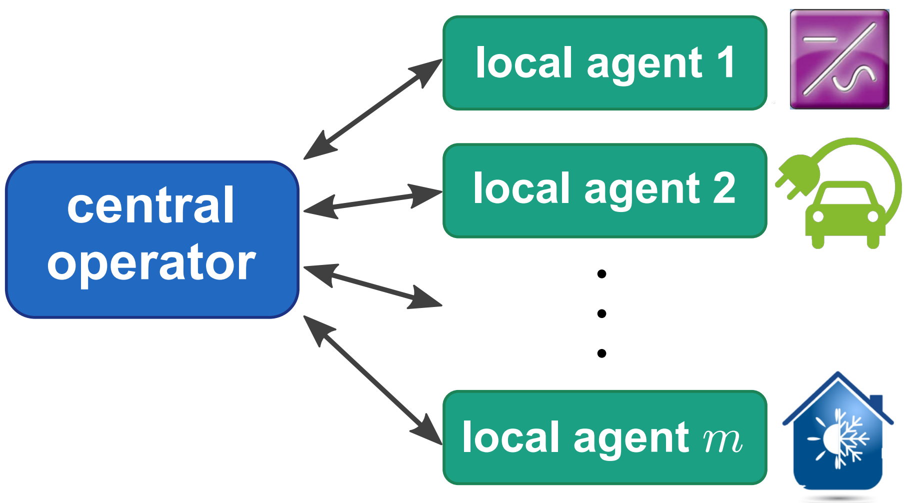
$$\sigma := \sup_t \left\| \begin{bmatrix} x_t^* - x_{t-1}^* \\ \lambda_t^* - \lambda_{t-1}^* \end{bmatrix} \right\|$$

is less than some threshold and $(\hat{x}_0, \hat{\lambda}_0)$ is sufficiently close to (x_0^*, λ_0^*), we have

$$\left\| \begin{bmatrix} \hat{x}_t - x_t^* \\ \hat{\lambda}_t - \lambda_t^* \end{bmatrix} \right\| \leq \frac{\rho(\alpha, \beta, \epsilon)}{1 - \rho(\alpha, \beta, \epsilon)} \left(\sigma + C\epsilon\sqrt{\alpha\beta} \sup_t \|\lambda_t^*\| \right)$$

error term due to regularization

First-order Primal-dual Method



- central operator
  - measure state variables
  - update dual variables
- local agents
  - operate & measure DERs
  - update primal variables

E. Dall'Anese and A. Simonetto. "Optimal power flow pursuit", 2018.
E. Dall'Anese, et al. "Optimal regulation of virtual power plants", 2018.

Implications from Simulations

- Second-order primal method
  - Tracks better than first-order methods
  - Requires less frequent measurements
  - Computing estimated Hessian is difficult

- First-order primal-dual method
  - Tracking performance not as good as second-order method
  - Requires more frequent measurements
  - Simpler to implement in a distributed manner

Implications from Simulations

- Balance trade-offs
  - measurement equipment capabilities
  - communication / cyber network
  - computation
  - the scale of the power network itself

- Constraint violations
  - Constraints on state variables can be violated
  - Violations are usually small and temporary

Ongoing Work

- Second-order primal: penalty function
- First-order primal-dual: regularized Lagrangian

Both approaches will introduce an addition error term!

- What if we use augmented Lagrangian?

$$\begin{aligned} \min_x \quad & c_t(x) \\ \text{s.t.} \quad & f_t(x) \le 0 \\ & x \in \mathcal{X}_t \end{aligned}$$

$$L_t(x, \lambda) = c_t(x) + \frac{\epsilon}{2} \sum_i \max\{\epsilon^{-1} f_{t,i}(x) + \lambda_i, 0\}^2 - \frac{\epsilon}{2} \|\lambda\|^2$$

"combination" of penalty function and regularized Lagrangian

Ongoing Work

- Saddle-point dynamics on augmented Lagrangian achieves **exponential stability**
  - Linear constraint case: G. Qu and N. Li, “On the exponential stability of primal-dual gradient dynamics”, 2018.

exponential stability (continuous time) ⟷ **linear convergence (discrete time)**

↓

bounded tracking error (time-varying)

- Can also be adopted in second-order methods.

Collaborators & References

- Steven Low (Caltech)
- Krishnamurthy Dvijotham (DeepMind)
- Emiliano Dall'Anese (NREL)
- Andrey Bernstein (NREL)
- Guanna Qu (Harvard)
- Na Li (Harvard)

- Y. Tang, K. Dvijotham and S. Low. "Real-time optimal power flow", 2017.
- Y. Tang and S. Low. "Distributed algorithm for time-varying optimal power flow", 2017.
- Y. Tang, E. Dall'Anese, A. Bernstein and S. Low. "A Feedback-Based Regularized Primal-Dual Gradient Method for Time-Varying Nonconvex Optimization", 2018.

Thank you!