Robust, Asynchronous and Distributed Algorithms for Control and Estimation in Smart Grids

Ph.D. Defense

Marco Todescato

Advisor: Prof. Ruggero Carli
Co-advisor: Prof. Luca Schenato

1^{st} April 2016

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Distributed Optimization

Design of Multi-Agent based, Distributed, Scalable and Robust algorithms for Large Scale Systems

Distributed

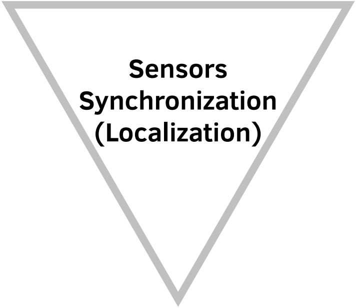
Requiring only local communications among the "smart" agents, elements of the network

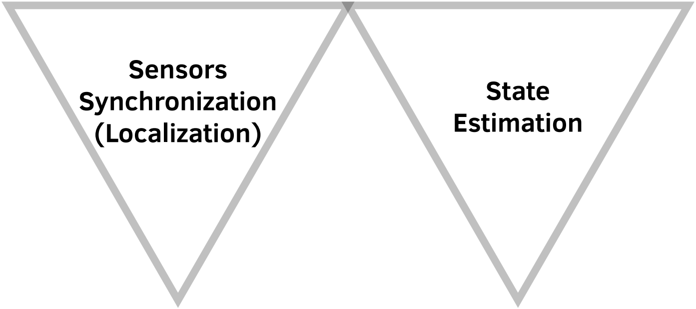
Scalable

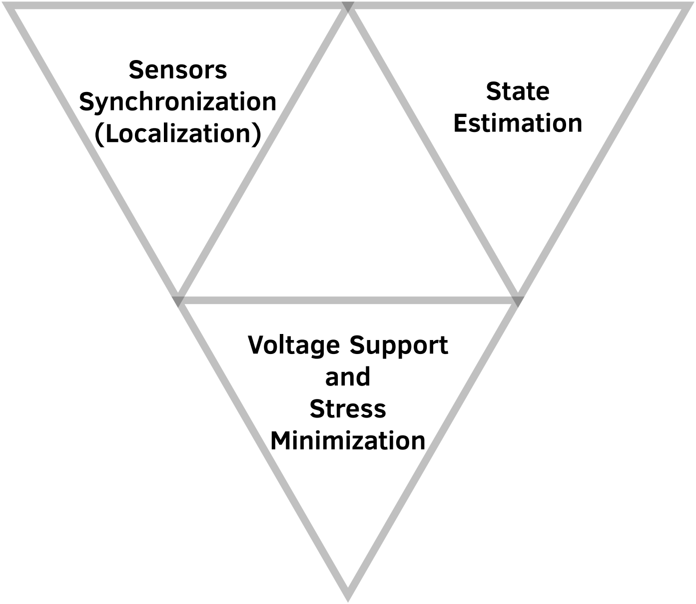
Not requiring SW upgrade due to HW upgrade

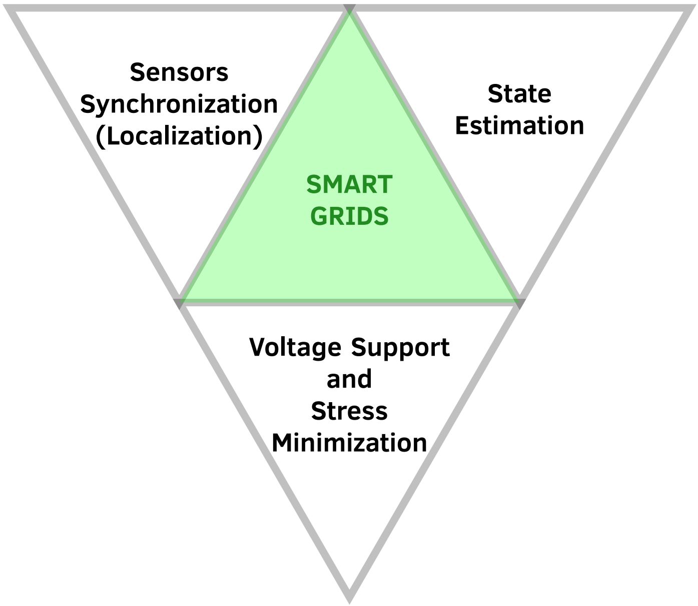
Robust

Resilient to failure in the communication channel


Sensors
Synchronization
(Localization)


Sensors Synchronization (Localization)
State Estimation


Sensors
Synchronization
(Localization)
State
Estimation
Voltage Support
and
Stress
Minimization


Sensors
Synchronization
(Localization)
State
Estimation
SMART
GRIDS
Voltage Support
and
Stress
Minimization

Localization:

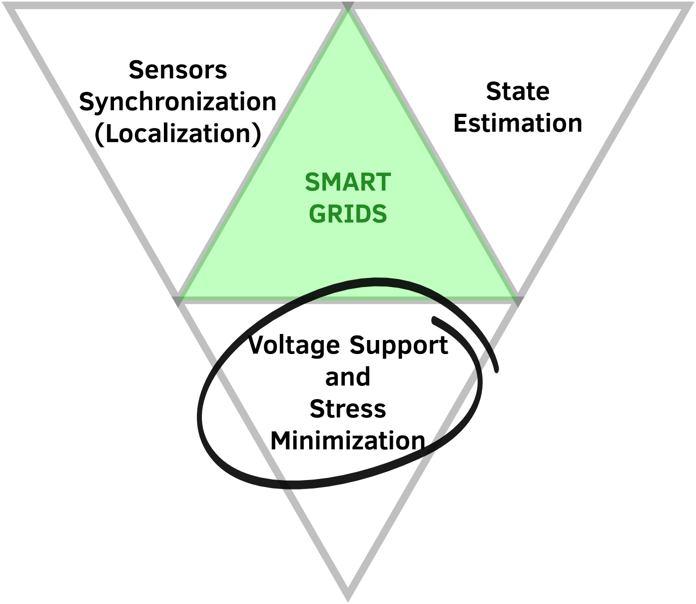
- two algorithms, i.e., *consensus-based* and *gradient-based*
- proved exponential convergence to the optimal least-square solution (in mean square sense) for *random communications*
- robustness (with exponential convergence) to packet losses and delays in the communication channel for deterministic communications

Localization:

- two algorithms, i.e., *consensus-based* and *gradient-based*
- proved exponential convergence to the optimal least-square solution (in mean square sense) for *random communications*
- robustness (with exponential convergence) to packet losses and delays in the communication channel for deterministic communications

State Estimation:

- two algorithms, i.e., *ADMM-based* and *Block-Jacobi* (generalized gradient)
- exponential convergence to the optimal (least-squares) solution
- Block-Jacobi: robust to packet losses and delays in the communication


Sensors
Synchronization
(Localization)
State
Estimation
SMART
GRIDS
Voltage Support
and
Stress
Minimization

Outline

- Modeling
- Stress Minimization & the Planning Problem
- Distributed Stress Minimization
- Conclusions

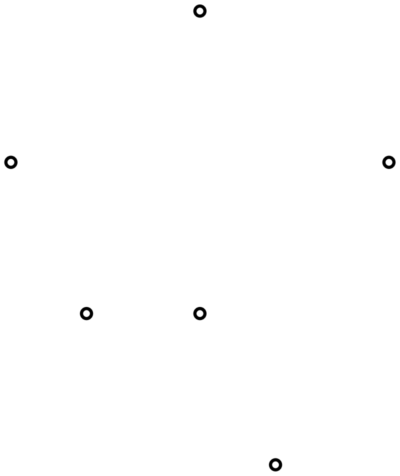
Outline

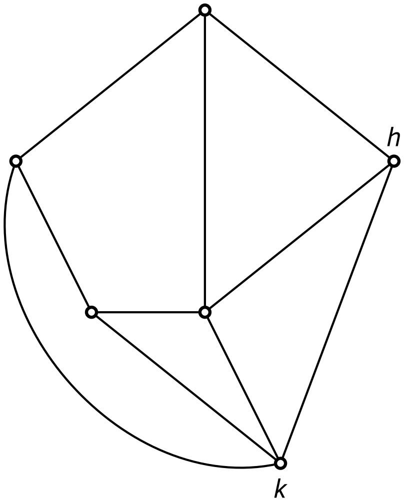
Modeling

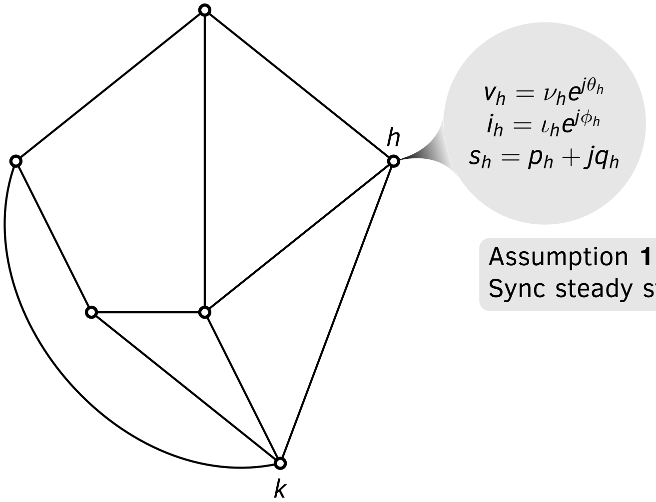
Stress Minimization & the Planning Problem

Distributed Stress Minimization

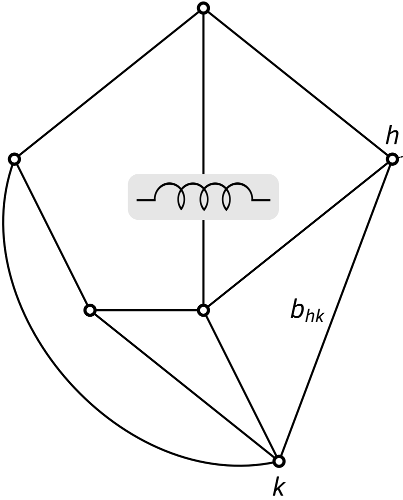
Conclusions




h
k



Assumption **1**.
Sync steady state regime

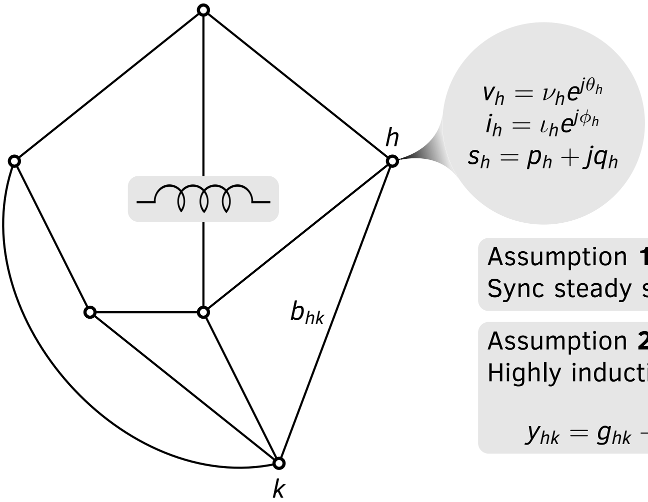


$$v_h = \nu_h e^{j\theta_h}$$
$$i_h = \iota_h e^{j\phi_h}$$
$$s_h = p_h + jq_h$$

Assumption **1**.
Sync steady state regime

Assumption **2**.
Highly inductive lines

$$y_{hk} = g_{hk} + jb_{hk} \simeq jb_{hk}$$



Assumption **1**.
Sync steady state regime

Assumption **2**.
Highly inductive lines

$$y_{hk} = g_{hk} + jb_{hk} \simeq jb_{hk}$$

B: $[B]_{hk} = b_{hk}$, $[B]_{hh} = -\sum_k b_{hk} + b^h_{\text{shunt}}$ susceptance matrix;
A graph incidence matrix;
$\mathbf{v}$, $\mathbf{i}$, $\mathbf{s}$, ν, θ vector notation.

Reactive Power Flow Equations (RPFEs)

(KCL + KVL) + PFEs

$$\mathbf{i} = jB\mathbf{v} \;\; , \quad \mathbf{s} = \mathrm{diag}(\mathbf{v})\bar{\mathbf{i}} \quad \Longrightarrow \quad \mathbf{s} = \mathrm{diag}(\mathbf{v})\overline{jB\mathbf{v}}$$

Reactive Power Flow Equations (RPFEs)

(KCL + KVL) + PFEs

$$\mathbf{i} = jB\mathbf{v} \quad , \quad \mathbf{s} = \mathrm{diag}(\mathbf{v})\bar{\mathbf{i}} \quad \Longrightarrow \quad \mathbf{s} = \mathrm{diag}(\mathbf{v})\overline{jB\mathbf{v}}$$

APFEs

$$\mathbf{p} = \mathrm{diag}(\boldsymbol{\nu})B\left(\mathrm{diag}(\sin A\boldsymbol{\theta})\boldsymbol{\nu}\right)$$

RPFEs

$$\mathbf{q} = \mathrm{diag}(\boldsymbol{\nu})B\left(\mathrm{diag}(\cos A\boldsymbol{\theta})\boldsymbol{\nu}\right)$$

Reactive Power Flow Equations (RPFEs)

(KCL + KVL) + PFEs

$$\mathbf{i} = jB\mathbf{v} \quad , \quad \mathbf{s} = \mathrm{diag}(\mathbf{v})\bar{\mathbf{i}} \quad \Longrightarrow \quad \mathbf{s} = \mathrm{diag}(\mathbf{v})\overline{jB\mathbf{v}}$$

APFEs

$$\mathbf{p} = \mathrm{diag}(\boldsymbol{\nu})B\left(\mathrm{diag}(\sin A\boldsymbol{\theta})\boldsymbol{\nu}\right)$$

RPFEs

$$\mathbf{q} = \mathrm{diag}(\boldsymbol{\nu})B\left(\mathrm{diag}(\cos A\boldsymbol{\theta})\boldsymbol{\nu}\right)$$

Reactive Power Flow Equations (RPFEs)

(KCL + KVL) + PFEs

$$\mathbf{i} = jB\mathbf{v} \;,\quad \mathbf{s} = \mathrm{diag}(\mathbf{v})\bar{\mathbf{i}} \quad \Longrightarrow \quad \mathbf{s} = \mathrm{diag}(\mathbf{v})\overline{jB\mathbf{v}}$$

APFEs

$$\mathbf{p} = \mathrm{diag}(\boldsymbol{\nu})B\left(\mathrm{diag}(\sin A\boldsymbol{\theta})\boldsymbol{\nu}\right)$$

RPFEs

$$\mathbf{q} = \mathrm{diag}(\boldsymbol{\nu})B\left(\mathrm{diag}(\cos A\boldsymbol{\theta})\boldsymbol{\nu}\right)$$

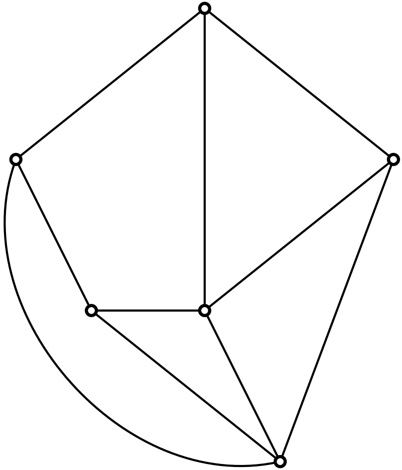
⇓

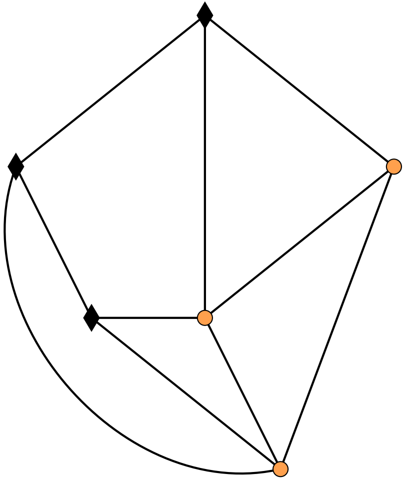
Decoupling Assumption:
$\|A\boldsymbol{\theta}\|_\infty \leq \gamma\,,\quad \gamma \in [0, \frac{\pi}{2}[$

⇓

$$\mathbf{q} = \mathrm{diag}(\boldsymbol{\nu})B\boldsymbol{\nu}$$

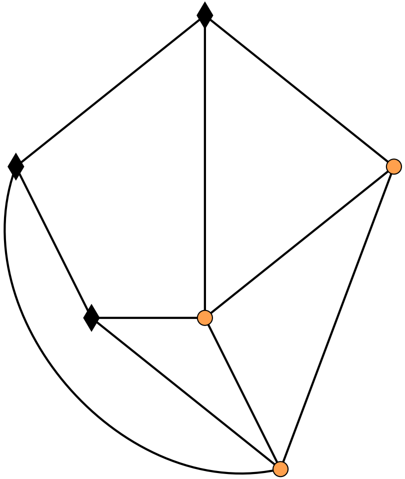
We are interested only in the relation between voltage amplitudes $\boldsymbol{\nu}$ and reactive power $\mathbf{q}$





◆ *g*enerators (PV nodes)
● *ℓ*oads (PQ nodes)

$$B = \begin{bmatrix} B_{\ell\ell} & B_{\ell g} \\ B_{g\ell} & B_{gg} \end{bmatrix}, \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_\ell \\ \boldsymbol{\nu}_g \end{bmatrix}, \mathbf{q} = \begin{bmatrix} \mathbf{q}_\ell \\ \mathbf{q}_g \end{bmatrix}$$



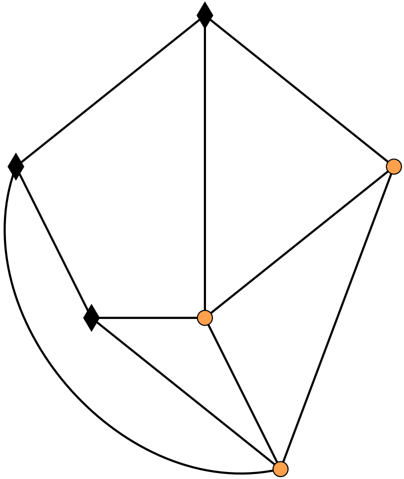
♦ *g*enerators (PV nodes)
● *ℓ*oads (PQ nodes)

$$B = \begin{bmatrix} B_{\ell\ell} & B_{\ell g} \\ B_{g\ell} & B_{gg} \end{bmatrix}, \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_\ell \\ \boldsymbol{\nu}_g \end{bmatrix}, \mathbf{q} = \begin{bmatrix} \mathbf{q}_\ell \\ \mathbf{q}_g \end{bmatrix}$$

⇩

Load RPFEs

$$\begin{aligned} \mathbf{q}_\ell &= -\mathrm{diag}(\boldsymbol{\nu}_\ell)\left(B_{\ell\ell}\boldsymbol{\nu}_\ell + B_{\ell g}\boldsymbol{\nu}_g\right) \\ &= -\mathrm{diag}(\boldsymbol{\nu}_\ell)B_{\ell\ell}\left(\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*\right) \end{aligned}$$



♦ generators (PV nodes)
● ℓoads (PQ nodes)

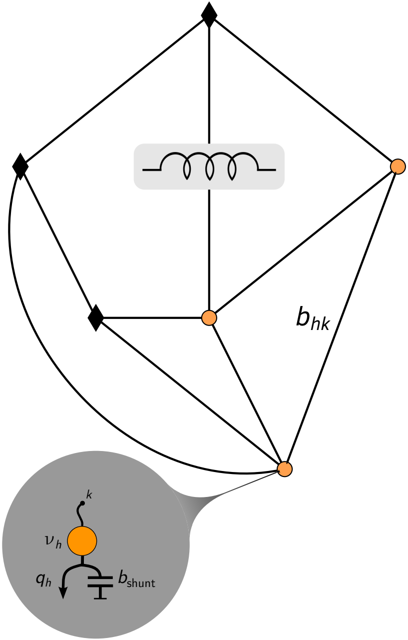
$$B = \begin{bmatrix} B_{\ell\ell} & B_{\ell g} \\ B_{g\ell} & B_{gg} \end{bmatrix}, \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_\ell \\ \boldsymbol{\nu}_g \end{bmatrix}, \mathbf{q} = \begin{bmatrix} \mathbf{q}_\ell \\ \mathbf{q}_g \end{bmatrix}$$

Load RPFEs

$$\begin{aligned} \mathbf{q}_\ell &= -\mathrm{diag}(\boldsymbol{\nu}_\ell)\left(B_{\ell\ell}\boldsymbol{\nu}_\ell + B_{\ell g}\boldsymbol{\nu}_g\right) \\ &= -\mathrm{diag}(\boldsymbol{\nu}_\ell)B_{\ell\ell}\left(\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*\right) \end{aligned}$$

Open Circuit Solution

$$\boldsymbol{\nu}_\ell^* := -B_{\ell\ell}^{-1}B_{\ell g}\boldsymbol{\nu}_g = \boldsymbol{\nu}_\ell|_{\mathbf{q}_\ell = 0}$$



♦ generators (PV nodes)
● ℓoads (PQ nodes)

$$B = \begin{bmatrix} B_{\ell\ell} & B_{\ell g} \\ B_{g\ell} & B_{gg} \end{bmatrix}, \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_\ell \\ \boldsymbol{\nu}_g \end{bmatrix}, \mathbf{q} = \begin{bmatrix} \mathbf{q}_\ell \\ \mathbf{q}_g \end{bmatrix}$$

⇩

Load RPFEs

$$\begin{aligned} \mathbf{q}_\ell &= -\mathrm{diag}(\boldsymbol{\nu}_\ell)\left(B_{\ell\ell}\boldsymbol{\nu}_\ell + B_{\ell g}\boldsymbol{\nu}_g\right) \\ &= -\mathrm{diag}(\boldsymbol{\nu}_\ell)B_{\ell\ell}\left(\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*\right) \end{aligned}$$

Open Circuit Solution

$$\boldsymbol{\nu}_\ell^* := -B_{\ell\ell}^{-1}B_{\ell g}\boldsymbol{\nu}_g = \boldsymbol{\nu}_\ell|_{\mathbf{q}_\ell=0}$$

Outline

Voltage Support

Standard (Security) Requirement

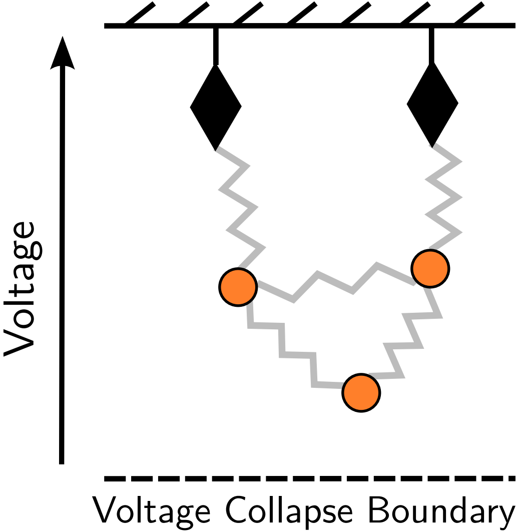
Load buses voltage magnitudes $\boldsymbol{\nu}_\ell$ must lie within a predefined percentage deviation α from a reference voltage ν_N

$$\frac{\|\boldsymbol{\nu}_\ell - \nu_N \mathbf{1}\|_\infty}{\nu_N} \leq \alpha$$

Reasons:

- loads and some components are designed to operate with a voltage in a narrow region around the network base voltage
- a flat voltage profile minimizes current flows and power losses
- a flat profile usually reduces the sensitivity of the voltage profile with respect to load changes
- **by conventional wisdom, a flat voltage profile indicates safety from voltage collapse**

Mechanical Equivalent



Security Requirement Inadequacy

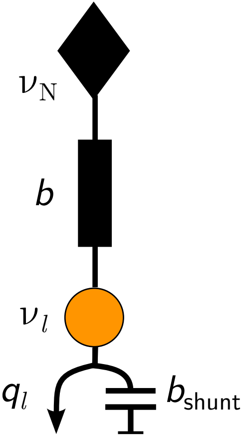
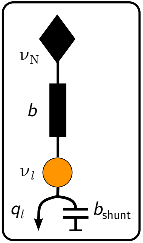


Figure: 2 buses case: one generator ν_N, one load (ν_ℓ, q_ℓ), one line b and, possibly, one shunt b_{shunt}

Security Requirement Inadequacy

$$\mathbf{q}_\ell = -\mathrm{diag}(\boldsymbol{\nu}_\ell) B_{\ell\ell} \left(\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*\right)$$



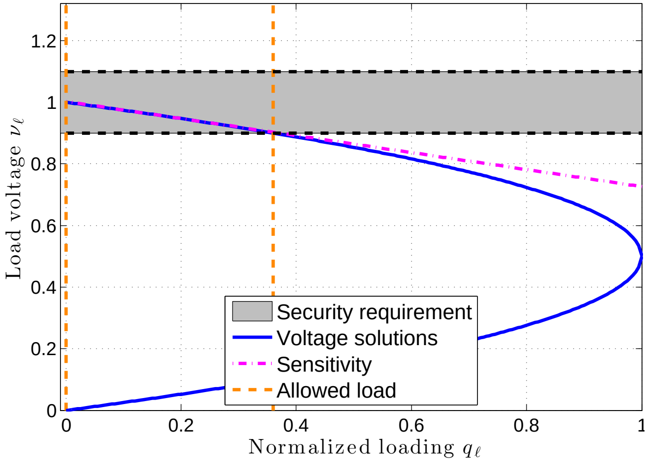
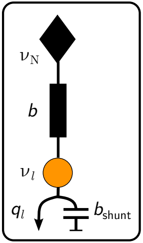


Figure: Absence of shunts, $b_{\text{shunt}} = 0$

Security Requirement Inadequacy

$$\mathbf{q}_\ell = -\mathrm{diag}(\boldsymbol{\nu}_\ell) B_{\ell\ell} \left(\boldsymbol{\nu}_\ell - \boldsymbol{\nu}^*_\ell\right)$$



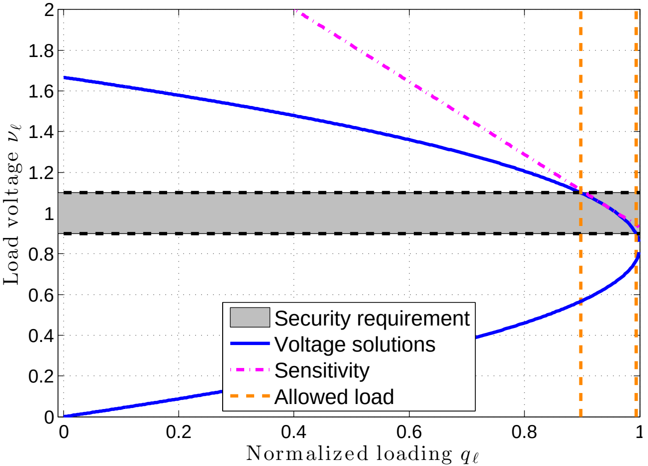
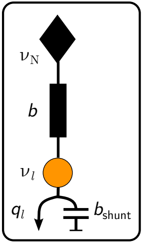


Figure: Presence of shunts, $b_{\mathrm{shunt}} \neq 0$

Security Requirement Inadequacy

$$\mathbf{q}_\ell = -\mathrm{diag}(\boldsymbol{\nu}_\ell) B_{\ell\ell} \left(\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*\right)$$



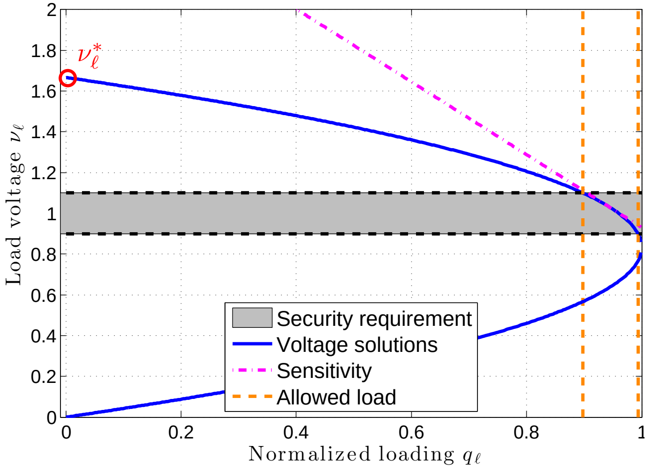


Figure: Presence of shunts, $b_{\mathrm{shunt}} \neq 0$

Stress Minimization

Idea: minimize distance from $\boldsymbol{\nu}_\ell^*$ subject to operational (security) requirement

Control variables: additional finite amount of reactive power $\mathbf{q}_{\mathrm{ctrl}}$ at load buses

Stress Minimization

$$\underset{\mathbf{q}_{\mathrm{ctrl}}}{\text{minimize}} \quad \|\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*\|_\infty$$

$$\text{subject to} \quad \begin{cases} \text{security requirement} \\ \underline{\mathbf{q}} \leq \mathbf{q}_{\mathrm{ctrl}} \leq \overline{\mathbf{q}} \text{ (injection limits)} \\ \text{Load RPFEs} \end{cases}$$

Stress Minimization

Idea: minimize distance from $\boldsymbol{\nu}_\ell^*$ subject to operational (security) requirement

Control variables: additional finite amount of reactive power $\mathbf{q}_{\mathrm{ctrl}}$ at load buses

Stress Minimization

$$\underset{\mathbf{q}_{\mathrm{ctrl}}}{\text{minimize}} \quad \|\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*\|_\infty$$

$$\text{subject to} \quad \begin{cases} \text{security requirement} \\ \underline{\mathbf{q}} \leq \mathbf{q}_{\mathrm{ctrl}} \leq \overline{\mathbf{q}} \ \text{(injection limits)} \\ \text{Load RPFEs} \end{cases}$$

Stress Cost/Measure

Issues:

- non linear
- non convex
- hard to solve

Linearization

Assumption: $\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}} \simeq 0$, i.e., loading + control overall sufficiently small

First Order Solution (of Load RPFEs)

$$\widehat{\boldsymbol{\nu}_\ell} = \underbrace{\mathrm{diag}(\boldsymbol{\nu}_\ell^*)}_{\text{o.c. solution}}\big(\mathbf{1} + \underbrace{\mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}})}_{\text{linear dev. in the loads}}\big)$$

$\mathcal{Q} \;:=\; \mathrm{diag}(\boldsymbol{\nu}_\ell^*) B_{\ell\ell} \mathrm{diag}(\boldsymbol{\nu}_\ell^*)$: weighted grounded laplacian matrix.
It is sparse and encodes the "stiffness" of the grid

Linearization

Assumption: $\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}} \simeq 0$, i.e., loading + control overall sufficiently small

First Order Solution (of Load RPFEs)

$$\widehat{\boldsymbol{\nu}_\ell} = \underbrace{\text{diag}(\boldsymbol{\nu}_\ell^*)}_{\text{o.c. solution}}\big(\mathbf{1} + \underbrace{\mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}})}_{\text{linear dev. in the loads}}\big)$$

Stress Cost

$$\|\boldsymbol{\nu}_\ell - \boldsymbol{\nu}_\ell^*\|_\infty$$

Security Requirement

$$\frac{\|\boldsymbol{\nu}_\ell - \nu_N \mathbf{1}\|_\infty}{\nu_N} \leq \alpha$$

Linearization

Assumption: $\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}} \simeq 0$, i.e., loading + control overall sufficiently small

First Order Solution (of Load RPFEs)

$$\widehat{\boldsymbol{\nu}_\ell} = \underbrace{\mathrm{diag}(\boldsymbol{\nu}_\ell^*)}_{\text{o.c. solution}}\Big(\mathbf{1} + \underbrace{\mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}})}_{\text{linear dev. in the loads}}\Big)$$

Stress Cost

$$\|\mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\text{ctrl}})\|_\infty$$

Security Requirement

$$\underline{\xi} \le \mathcal{Q}^{-1}\mathbf{q}_{\text{ctrl}} \le \overline{\xi}$$

Convex Stress Minimization

Convex Stress Minimization

$$\begin{aligned} \underset{\mathbf{q}_{\mathrm{ctrl}}}{\text{minimize}} \quad & \|\mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}})\|_\infty \\ \text{subject to} \quad & \begin{cases} \underline{\boldsymbol{\xi}} \leq \mathcal{Q}^{-1}\mathbf{q}_{\mathrm{ctrl}} \leq \overline{\boldsymbol{\xi}} \\ \underline{\mathbf{q}} \leq \mathbf{q}_{\mathrm{ctrl}} \leq \overline{\mathbf{q}} \ \text{(injection limits)} \end{cases} \end{aligned}$$

On the Stress Cost

$$\|\mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}})\|_\infty$$

In this context, it has been derived in the most natural way, i.e., as approximation of the distance between the open circuit solution and the current operating condition

However, it has been formally proved [1] that it represents an index related to the existence of a solution of the *nonlinear RPFEs* at the loads.

[1] J. W. Simpson-Porco, F. Dörfler, and F. Bullo, "Voltage collapse in complex power grids," Nature Communications, Mar. 2015, to appear.

The Planning Problem

Motivations: limited number of expensive resources

The Planning Problem

Motivations: limited number of expensive resources

Goal: account for the number of available resources and optimally allocate them

Idea: use re-weighted ℓ_1-norm — convex approximation of $\mathrm{card}(\cdot)$

The Planning Problem

Motivations: limited number of expensive resources

Goal: account for the number of available resources and optimally allocate them

Idea: use re-weighted ℓ_1-norm — convex approximation of $\mathrm{card}(\cdot)$

Sparse Stress Minimization

$$\begin{aligned} \underset{\mathbf{q}_{\mathrm{ctrl}}}{\text{minimize}} \quad & \|\mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}})\|_\infty + \gamma \|\mathrm{diag}\left(\mathbf{w}(\mathbf{q}_{\mathrm{ctrl}})\right)\mathbf{q}_{\mathrm{ctrl}}\|_1 \\ \text{subject to} \quad & \begin{cases} \underline{\boldsymbol{\xi}} \leq \mathcal{Q}^{-1}\mathbf{q}_{\mathrm{ctrl}} \leq \overline{\boldsymbol{\xi}} \\ \underline{\mathbf{q}} \leq \mathbf{q}_{\mathrm{ctrl}} \leq \overline{\mathbf{q}} \end{cases} \end{aligned}$$

The Planning Problem

Motivations: limited number of expensive resources

Goal: account for the number of available resources and optimally allocate them

Idea: use re-weighted ℓ_1-norm — convex approximation of $\mathrm{card}(\cdot)$

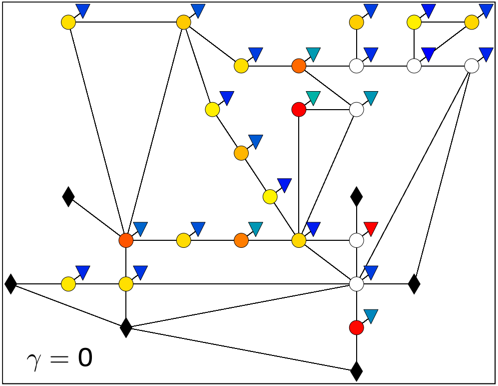
Sparse Stress Minimization

$$\underset{\mathbf{q}_{\mathrm{ctrl}}}{\text{minimize}} \quad \|\mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}})\|_\infty + \gamma \overbrace{\|\mathrm{diag}\left(\mathbf{w}(\mathbf{q}_{\mathrm{ctrl}})\right)\mathbf{q}_{\mathrm{ctrl}}\|_1}^{\text{re}-\text{weighted } \ell_1 - \text{norm}}$$

$$\text{subject to} \quad \begin{cases} \underline{\boldsymbol{\xi}} \leq \mathcal{Q}^{-1}\mathbf{q}_{\mathrm{ctrl}} \leq \overline{\boldsymbol{\xi}} \\ \underline{\mathbf{q}} \leq \mathbf{q}_{\mathrm{ctrl}} \leq \overline{\mathbf{q}} \end{cases}$$

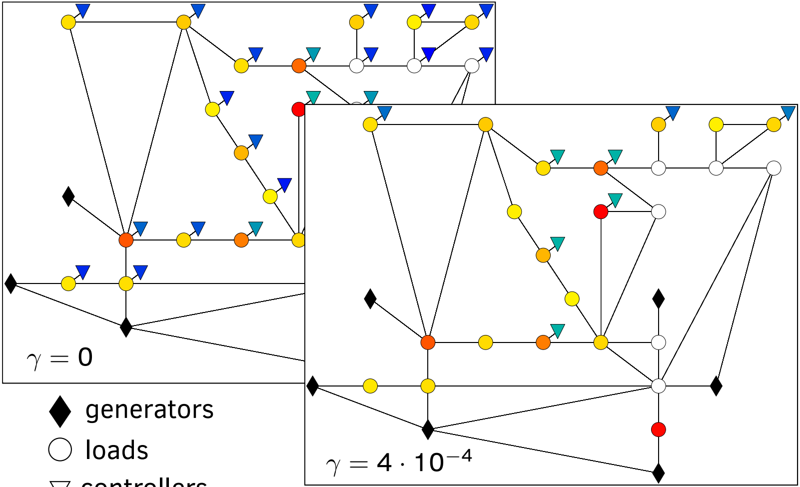
sparsity parameter (γ)

Simulations: Planning



- ◆ generators
- ○ loads
- ▽ controllers
- ■ — ■ increasing load absorption
- ■ — ■ increasing load injection
- □ no load

Simulations: Planning



◆ generators

○ loads

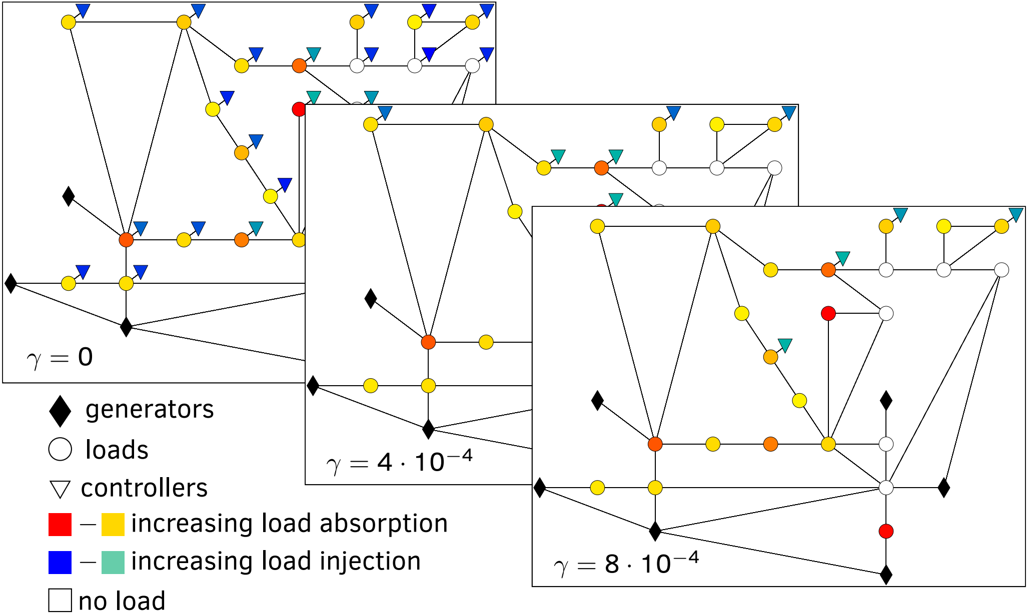
▽ controllers

red — yellow increasing load absorption

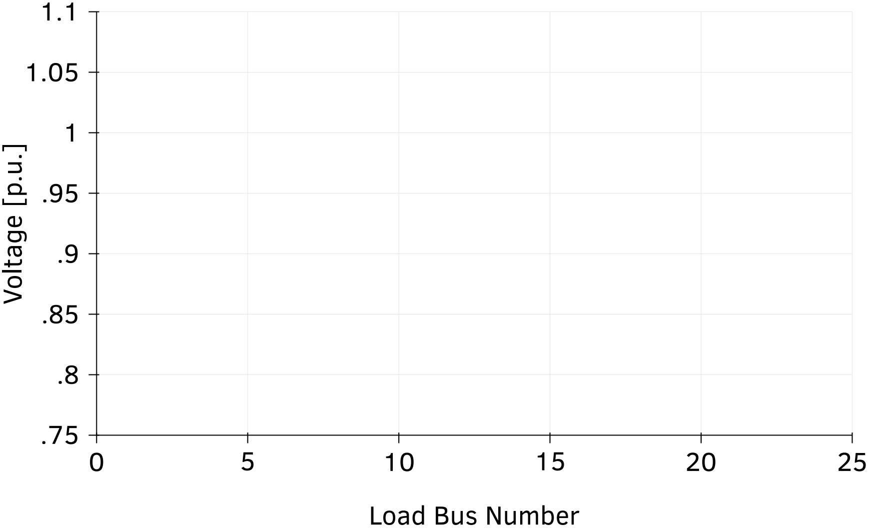
blue — green increasing load injection

□ no load

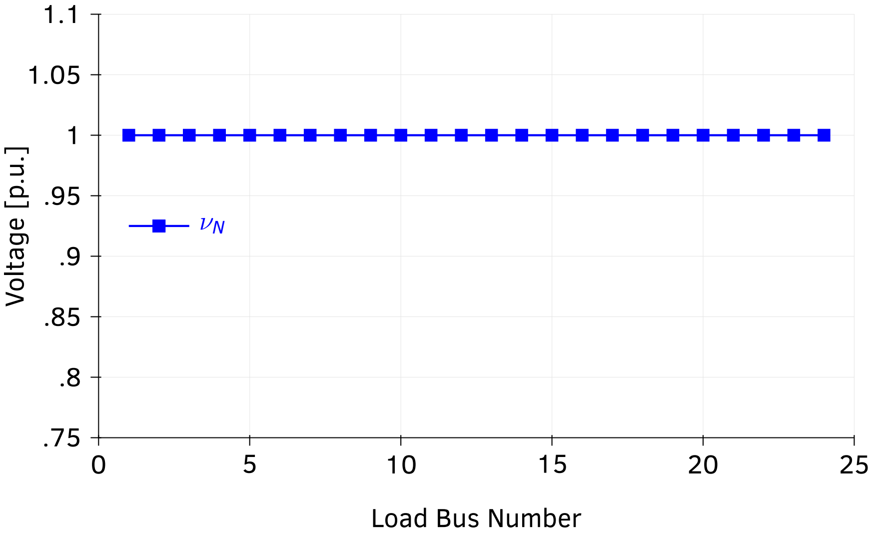
Simulations: Planning



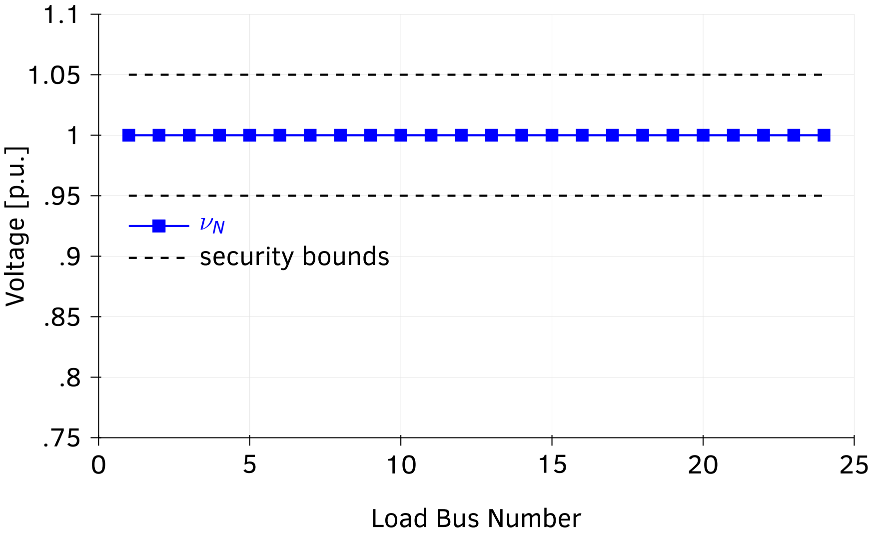
Simulations: Stress Minimization



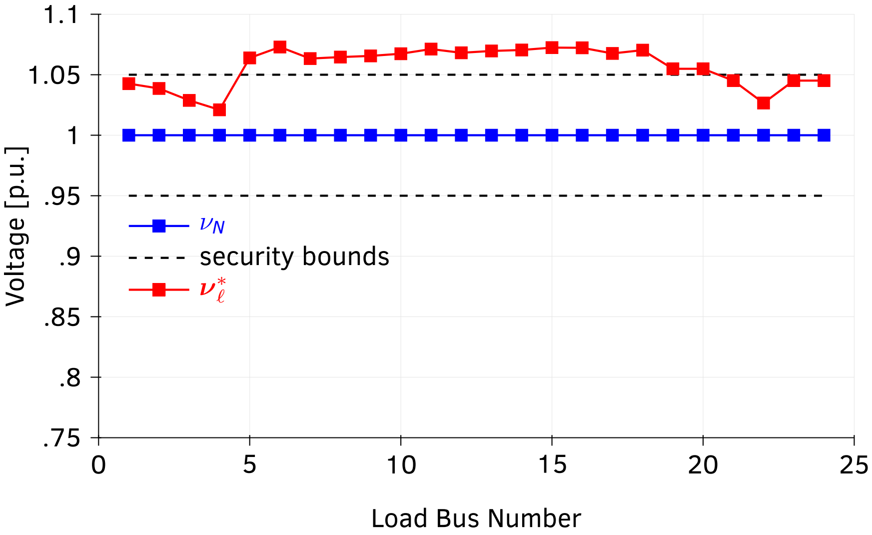
Simulations: Stress Minimization



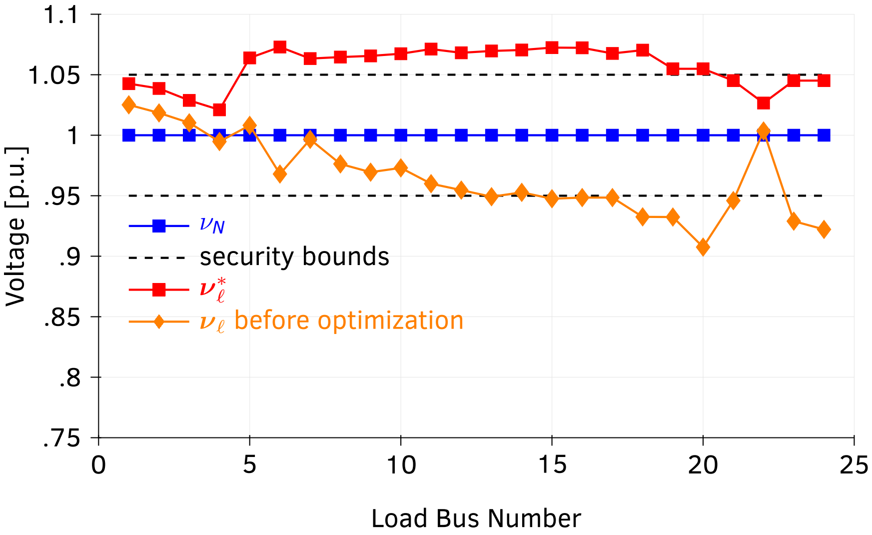
Simulations: Stress Minimization



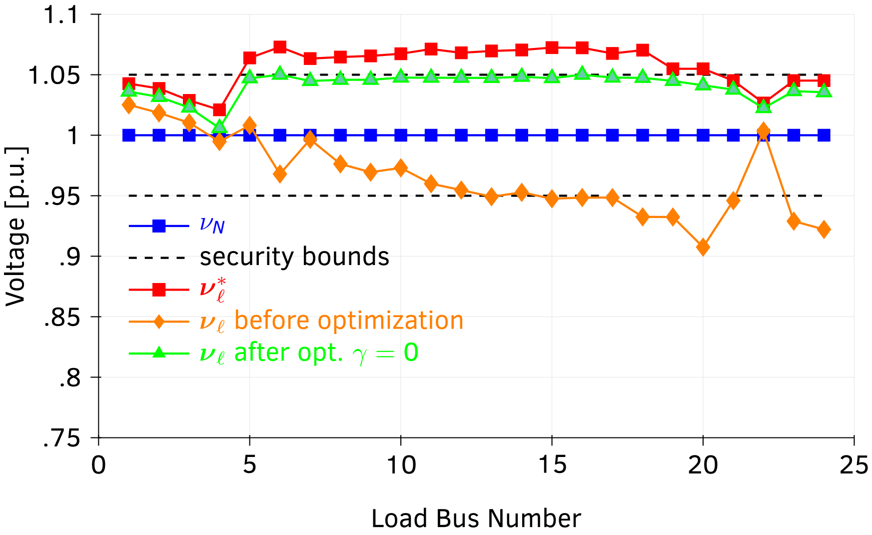
Simulations: Stress Minimization



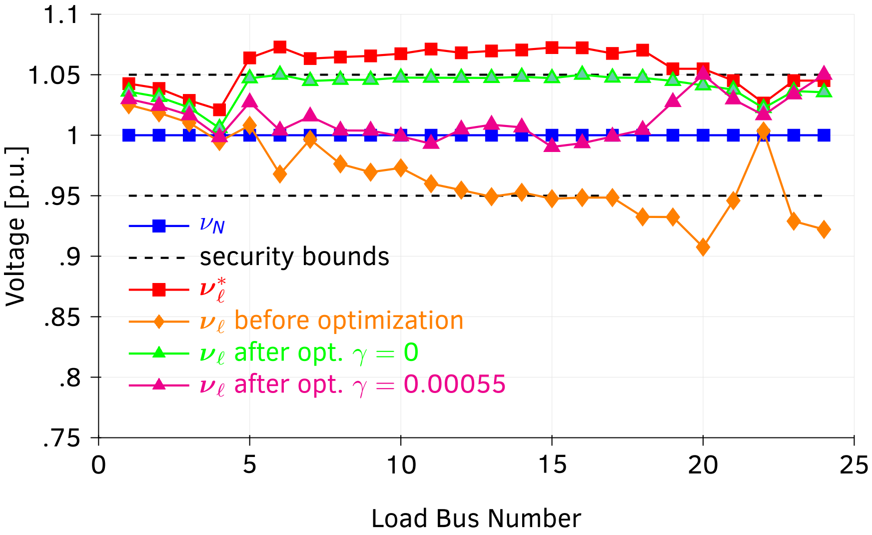
Simulations: Stress Minimization



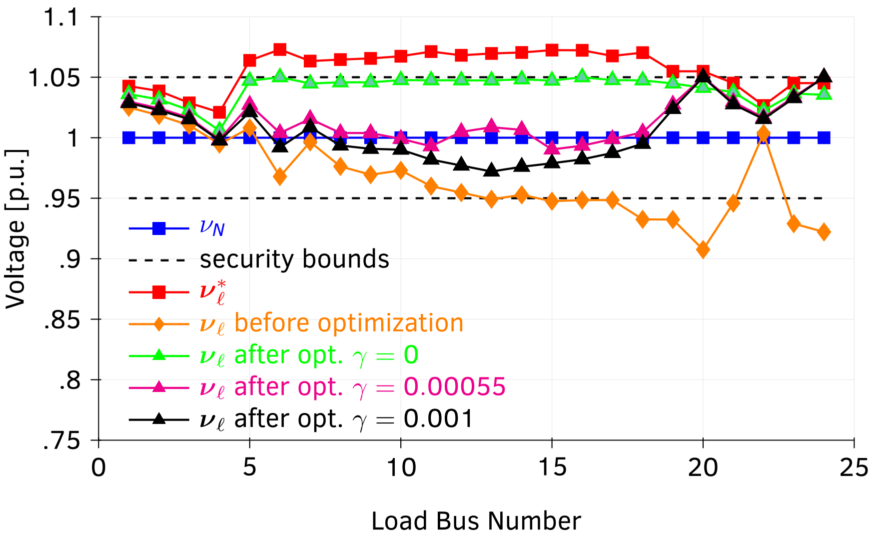
Simulations: Stress Minimization



Simulations: Stress Minimization



Simulations: Stress Minimization



Outline

Modeling

Stress Minimization & the Planning Problem

Distributed Stress Minimization

Conclusions

Preliminaries

Motivations:

- privacy: utilities might not want to share unecessary info
- feedback control: track time-varying loads

Goal: implement a distributed implementation for stress minimization

Assumptions:

1. every load equipped with a *smart-agent* with mild computing and communication capabilities
2. comminication graph built to coincide with the electric graph

Non sparse structure

Previous formulation (in injection coordinate)**:**

$$\underset{\mathbf{q}_{\mathrm{ctrl}}}{\text{minimize}} \ \|\mathcal{Q}^{-1}(\mathbf{q}_{\ell} + \mathbf{q}_{\mathrm{ctrl}})\|_{\infty}$$

Issue: $\mathcal{Q}^{-1}$ dense (full) matrix $\Rightarrow$ to compute cost and constraint we need info coming from all the loads

Non sparse structure

Previous formulation (in injection coordinate)**:**

$$\underset{\mathbf{q}_{\mathrm{ctrl}}}{\text{minimize}} \; \|\mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}})\|_\infty$$

Issue: $\mathcal{Q}^{-1}$ dense (full) matrix $\Rightarrow$ to compute cost and constraint we need info coming from all the loads

Idea: $\mathcal{Q}$ sparse with pattern of the adjacency of the graph connecting the loads

Reformulation - voltage coordinate

Voltage deviations: $\mathbf{x} := \mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}}) \quad \Longrightarrow \quad \mathbf{q}_{\mathrm{ctrl}} = \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell$

Reformulation - voltage coordinate

Voltage deviations: $\mathbf{x} := \mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}}) \implies \mathbf{q}_{\mathrm{ctrl}} = \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell$

Security requirements: $\underline{\boldsymbol{\xi}} \leq \mathcal{Q}^{-1}\mathbf{q}_{\mathrm{ctrl}} \leq \overline{\boldsymbol{\xi}} \implies \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}}$

Injection limits: $\underline{\mathbf{q}} \leq \mathbf{q}_{\mathrm{ctrl}} \leq \overline{\mathbf{q}} \implies \underline{\mathbf{q}} \leq \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell \leq \overline{\mathbf{q}}$

Reformulation - voltage coordinate

Voltage deviations: $\mathbf{x} := \mathcal{Q}^{-1}(\mathbf{q}_\ell + \mathbf{q}_{\mathrm{ctrl}}) \implies \mathbf{q}_{\mathrm{ctrl}} = \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell$

Security requirements: $\underline{\xi} \leq \mathcal{Q}^{-1}\mathbf{q}_{\mathrm{ctrl}} \leq \overline{\xi} \implies \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}}$

Injection limits: $\underline{\mathbf{q}} \leq \mathbf{q}_{\mathrm{ctrl}} \leq \overline{\mathbf{q}} \implies \underline{\mathbf{q}} \leq \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell \leq \overline{\mathbf{q}}$

Online Stress Minimization (in voltage coordinate)

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{x}\|_\infty \\ &\text{subject to} \quad \begin{cases} \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}} \\ \underline{\mathbf{q}} \leq \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell \leq \overline{\mathbf{q}} \end{cases} \end{aligned}$$

A smooth approximation for ∞-norm

Final Goal: Implement a gradient-based distributed strategy to solve for online stress minimization

Issue: ∞-norm not differentiable

A smooth approximation for ∞-norm

Final Goal: Implement a gradient-based distributed strategy to solve for online stress minimization

Issue: ∞-norm not differentiable

Idea: exploit a smooth approximation: *softmax* + $|\cdot|$ exponentiation

A smooth approximation for ∞-norm

Final Goal: Implement a gradient-based distributed strategy to solve for online stress minimization

Issue: ∞-norm not differentiable

Idea: exploit a smooth approximation: *softmax* + $|\cdot|$ exponentiation

A smooth decomposable approximation for ∞-norm

Consider the function $\tilde{f}_{\alpha,\epsilon}(\mathbf{x})$ defined, for $1 \ll \alpha$, $0 < \epsilon \ll 1$, as

$$\tilde{f}_{\alpha,\epsilon}(\mathbf{x}) := \mathrm{softmax}_{\alpha}(|\mathbf{x}|^{1+\epsilon}) = \frac{1}{\alpha} \log \left(\frac{1}{n} \sum_{i=1}^{n} \exp\left(\alpha |x_i|^{1+\epsilon}\right) \right)$$

then it holds that

$$\lim_{\substack{\alpha \to +\infty \\ \epsilon \to 0^+}} \tilde{f}_{\alpha,\epsilon}(\mathbf{x}) = \|\mathbf{x}\|_{\infty}$$

A smooth approximation for ∞-norm

Final Goal: Implement a gradient-based distributed strategy to solve for online stress minimization

Issue: ∞-norm not differentiable

Idea: exploit a smooth approximation: *softmax* + $|\cdot|$ exponentiation

A smooth decomposable approximation for ∞-norm

Consider the function $\tilde{f}_{\alpha,\epsilon}(\mathbf{x})$ defined, for $1 \ll \alpha$, $0 < \epsilon \ll 1$, as

$$\tilde{f}_{\alpha,\epsilon}(\mathbf{x}) := \mathrm{softmax}_{\alpha}(|\mathbf{x}|^{1+\epsilon}) = \frac{1}{\alpha} \log \left(\frac{1}{n} \sum_{i=1}^{n} \exp\left(\alpha |x_i|^{1+\epsilon}\right) \right)$$

then it holds that

$$\lim_{\substack{\alpha \to +\infty \\ \epsilon \to 0^+}} \tilde{f}_{\alpha,\epsilon}(\mathbf{x}) = \|\mathbf{x}\|_{\infty}$$

Moreover, consider $f_{\alpha,\epsilon}(\mathbf{x}) := n \exp\left(\alpha \tilde{f}_{\alpha,\epsilon}(\mathbf{x})\right)$. Then, $\mathrm{argmin}_{\mathbf{x}} \tilde{f} \equiv \mathrm{argmin}_{\mathbf{x}} f$.

A Distributed primal-dual feedback controller

Smooth Stress Minimization

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f_{\alpha,\epsilon}(\mathbf{x}) \\ \text{subject to} \quad & \begin{cases} \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}} & \textit{(volt. constr.)} \\ \underline{\mathbf{q}} \leq \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell \leq \overline{\mathbf{q}} & \textit{(inj. constr.)} \end{cases} \end{aligned} \tag{1}$$

A Distributed primal-dual feedback controller

Smooth Stress Minimization

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f_{\alpha,\epsilon}(\mathbf{x}) \\ \text{subject to} \quad & \begin{cases} \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}} & (\textit{volt. constr.}) \\ \underline{\mathbf{q}} \leq \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell \leq \overline{\mathbf{q}} & (\textit{inj. constr.}) \end{cases} \end{aligned} \tag{1}$$

Lagrangian: $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f_{\alpha,\epsilon}(\mathbf{x}) + \boldsymbol{\lambda}^T(\textit{volt. constr.}) + \boldsymbol{\mu}^T(\textit{inj. constr.})$

A Distributed primal-dual feedback controller

Smooth Stress Minimization

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f_{\alpha,\epsilon}(\mathbf{x}) \\ \text{subject to} \quad & \begin{cases} \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}} & (\textit{volt. constr.}) \\ \underline{\mathbf{q}} \leq \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell \leq \overline{\mathbf{q}} & (\textit{inj. constr.}) \end{cases} \end{aligned} \tag{1}$$

Lagrangian: $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f_{\alpha,\epsilon}(\mathbf{x}) + \boldsymbol{\lambda}^T(\textit{volt. constr.}) + \boldsymbol{\mu}^T(\textit{inj. constr.})$

Dual Ascent algorithm

$$\mathbf{x}(t+1) = \underset{\mathbf{x}}{\operatorname{argmin}}\ \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))$$

$$\boldsymbol{\lambda}(t+1) = \left[\boldsymbol{\lambda}(t) + \rho\left(\textit{volt. constr.}|_{\mathbf{x}(t+1)}\right)\right]^+$$

$$\boldsymbol{\mu}(t+1) = \left[\boldsymbol{\mu}(t) + \rho\left(\textit{inj. constr.}|_{\mathbf{x}(t+1)}\right)\right]^+$$

A Distributed primal-dual feedback controller

Smooth Stress Minimization

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f_{\alpha,\epsilon}(\mathbf{x}) \\ \text{subject to} \quad & \begin{cases} \underline{\mathbf{x}} \leq \mathbf{x} \leq \overline{\mathbf{x}} & (\textit{volt. constr.}) \\ \underline{\mathbf{q}} \leq \mathcal{Q}\mathbf{x} - \mathbf{q}_\ell \leq \overline{\mathbf{q}} & (\textit{inj. constr.}) \end{cases} \end{aligned} \tag{1}$$

Lagrangian: $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f_{\alpha,\epsilon}(\mathbf{x}) + \boldsymbol{\lambda}^T(\textit{volt. constr.}) + \boldsymbol{\mu}^T(\textit{inj. constr.})$

Dual Ascent algorithm

$$\mathbf{x}(t+1) = \underset{\mathbf{x}}{\operatorname{argmin}}\ \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t))$$

$$\boldsymbol{\lambda}(t+1) = \left[\boldsymbol{\lambda}(t) + \rho\left(\textit{volt. constr.}|_{\mathbf{x}(t+1)}\right)\right]^+$$

$$\boldsymbol{\mu}(t+1) = \left[\boldsymbol{\mu}(t) + \rho\left(\textit{inj. constr.}|_{\mathbf{x}(t+1)}\right)\right]^+$$

Convergence:
Assume Slater's conditions hold. Then, $\exists \overline{\rho}$ s.t. $\forall \rho \leq \overline{\rho}$ the primal-dual algorithm converges to the solution of (1)

Sketch of proof

Note: we reload the notation. $A\mathbf{x} - \mathbf{b}$ denote all the constraints. $\boldsymbol{\lambda}$ all the multipliers.

1. Assuming Slater's conditions there is zero duality gap, i.e., solving primal and dual problem is equivalent

$$\underset{\boldsymbol{\lambda}}{\text{maximize}}\; d(\boldsymbol{\lambda}) := \inf_{\mathbf{x}}\; f(\mathbf{x}) + \boldsymbol{\lambda}^T(A\mathbf{x} - \mathbf{b}) \qquad \text{subject to } \boldsymbol{\lambda} \geq 0$$

2. Thanks to strong convexity of f $\exists!$ solution
3. It is possible to show that $d(\boldsymbol{\lambda})$ is continuously differentiable and that

$$\nabla_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}) = A\mathbf{x}^*_{\boldsymbol{\lambda}} - b \qquad \mathbf{x}^*_{\boldsymbol{\lambda}} := \text{argmin}_{\mathbf{x}}\, f + \boldsymbol{\lambda}^T(Ax - b)$$

   Moreover, being $x^*_{\boldsymbol{\lambda}}$ Lipschitz in $\boldsymbol{\lambda}$, so is $\nabla_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda})$ which is linear in $\mathbf{x}^*_{\boldsymbol{\lambda}}$.
4. Finally, the dual ascent algorithm is a projected gradient on $d(\boldsymbol{\lambda})$ which is known to converge for a sufficiently small step size

Controller scheme

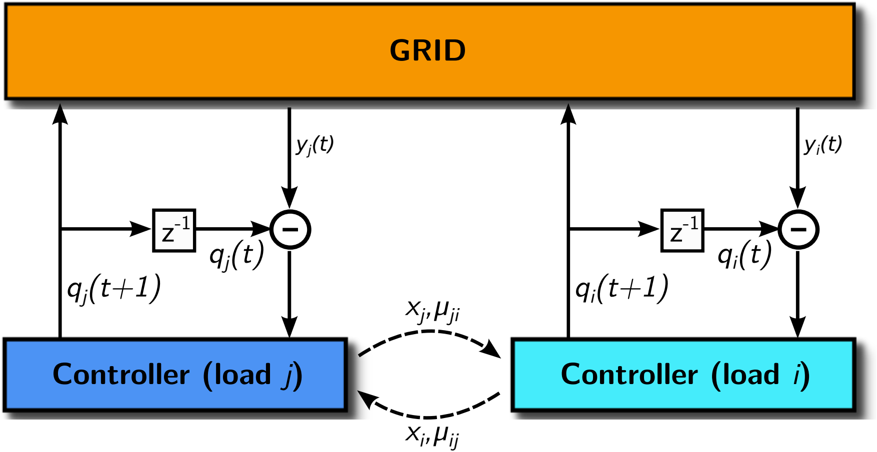
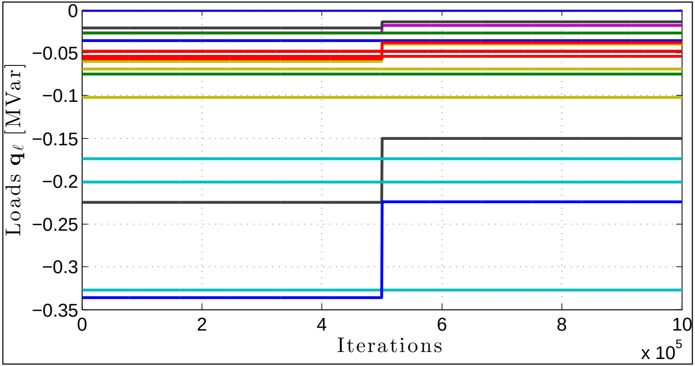
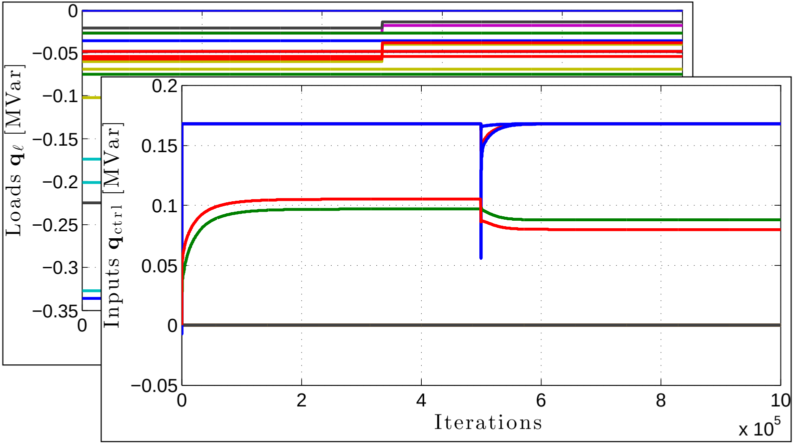


Figure: Feedback control scheme: reactive power measurements y are feedback to the controller which, thanks to local exchange of information between neighbor controllers, compute the new actuation value q

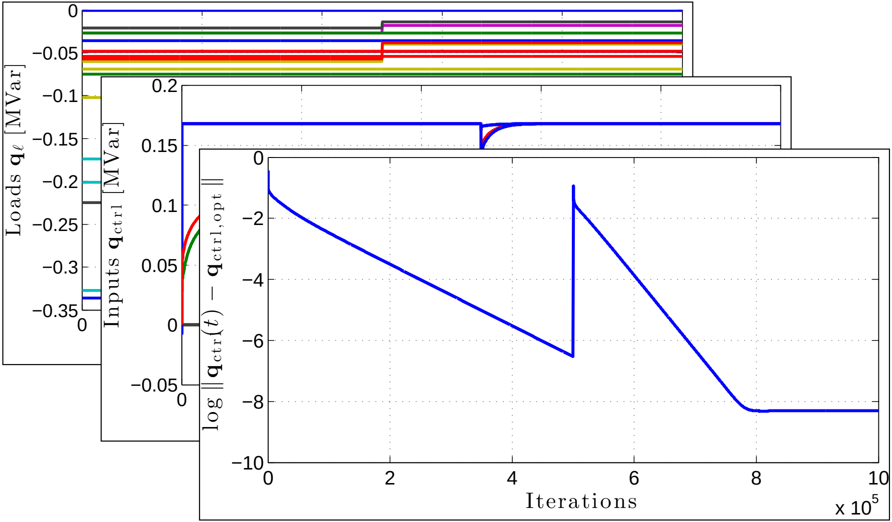
Simulations - Load step



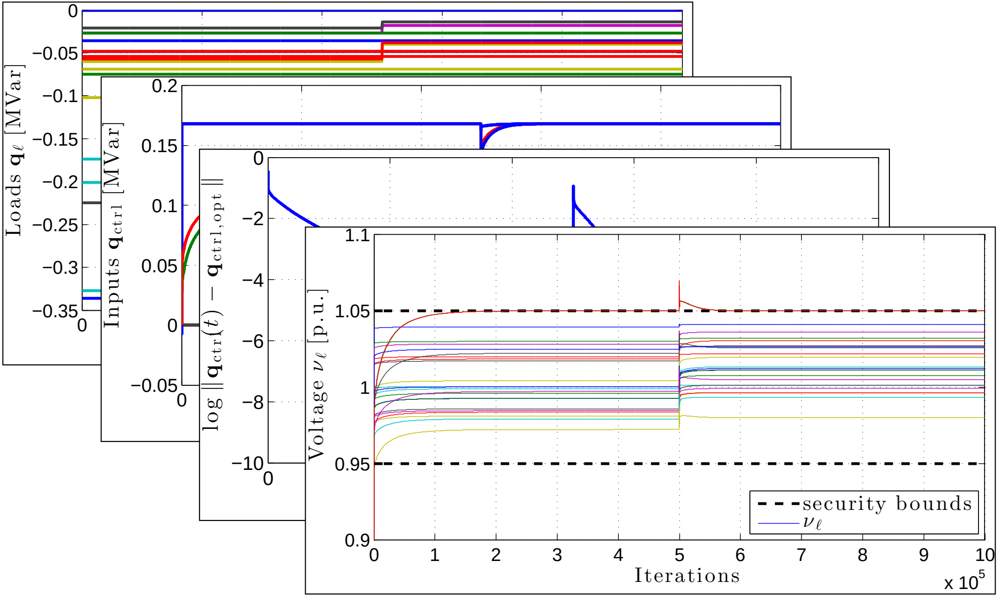
Simulations - Load step



Simulations - Load step



Simulations - Load step



Outline

Modeling

Stress Minimization & the Planning Problem

Distributed Stress Minimization

Conclusions

Conclusions

Security requirement possible inadequacy

Novel optimization framework accounting for

- **stress minimization**
- **optimal planning**

Distributed implementation trough feedback control

Thank you