

Feedback Optimization

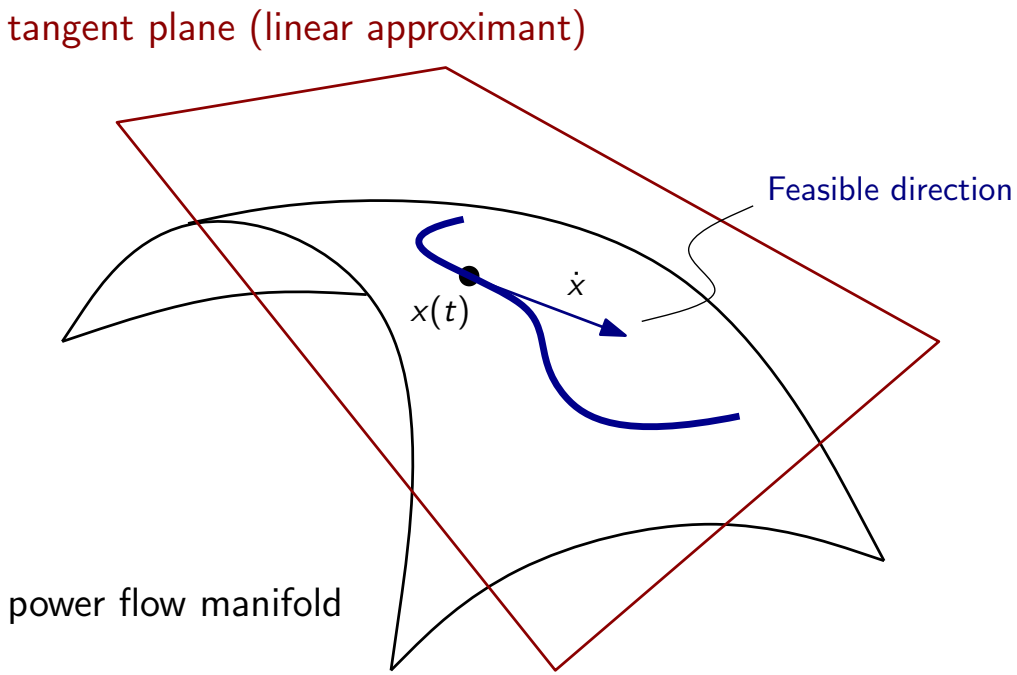
Saverio Bolognani, ETH Zurich

EECI School M07, May 6-10, 2024, Barcelona

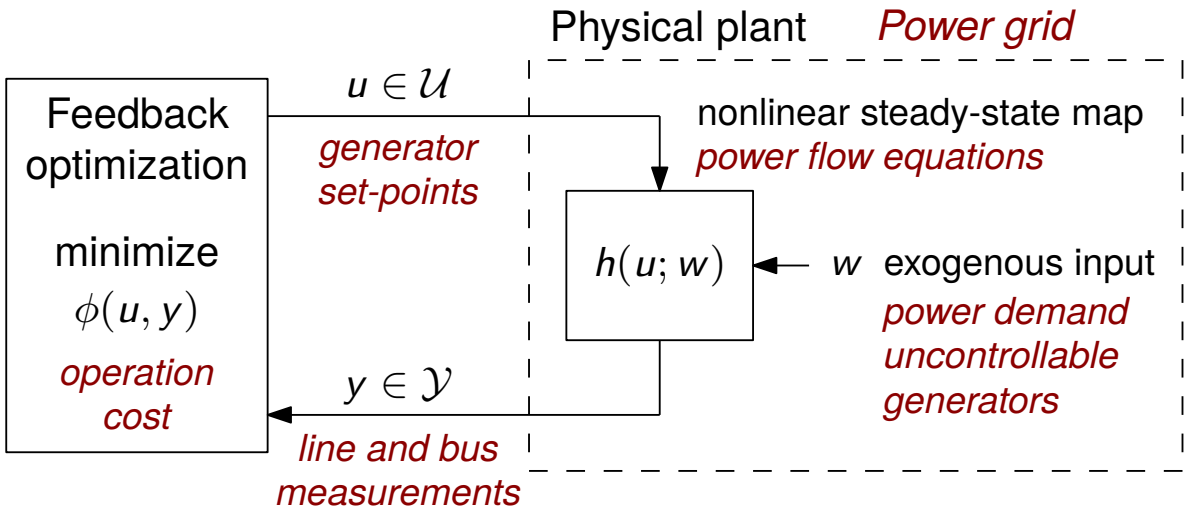
Dynamics on the PFM

Mathematical foundation + analysis:

- trajectories and **autonomous dynamics** on the PFM
  - existence, well-posedness
- convergence / **stability** on the PFM
- connection between
  - **critical points** and gradient flow equilibria ($\sim$ first-order optimality)
  - **minimizers** and stability ($\sim$ second-order optimality)



From trajectories to feedback control



How to design **feedback controllers** that yield the desired **(autonomous) closed-loop trajectory** $\dot{x} = f(x)$?

- add structure: **inputs** u**, measurements** y**, disturbances** w
- define **input/output map** $y = h(u; w)$ based on that structure
- design **feedback controller**

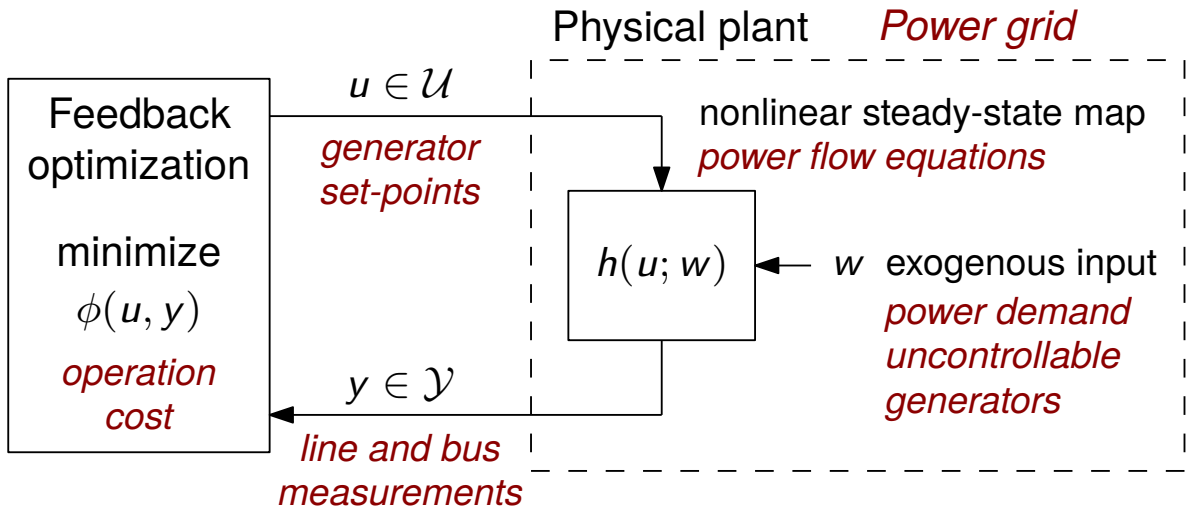
Outline

- Input-output steady-state model
- Examples of feedback optimization design
- Stability of the dynamic interconnection

Outline

- Input-output steady-state model
- Examples of feedback optimization design
- Stability of the dynamic interconnection

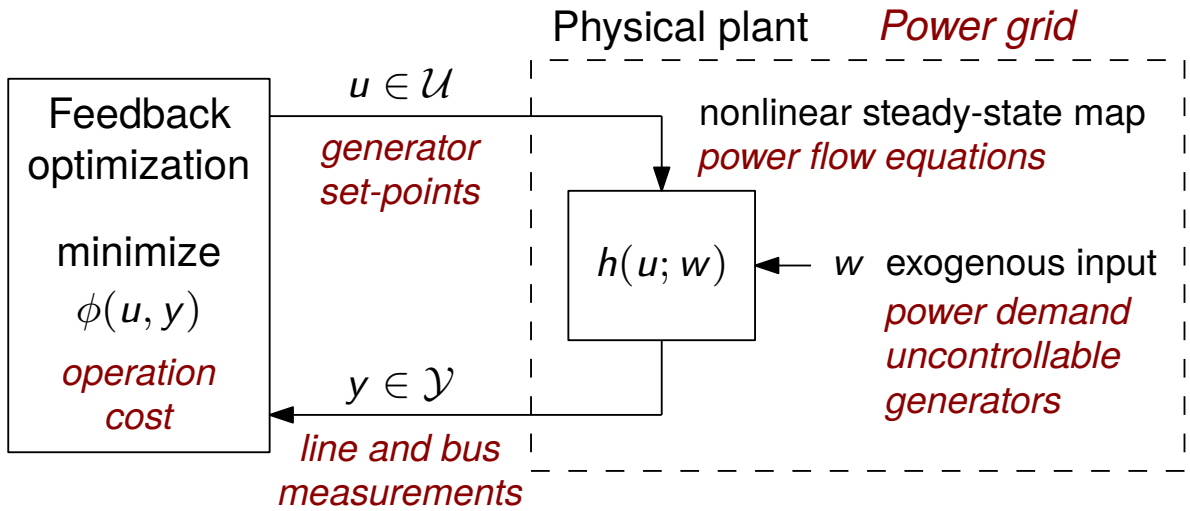
Inputs, outputs, disturbances



$$x = \begin{bmatrix} u \\ w \\ y \end{bmatrix} \begin{matrix} \text{controllable input} \\ \text{uncontrollable input} \\ \text{measured output} \end{matrix}$$

$\star$ Can be generalized to a generic change of coordinates

Inputs, outputs, disturbances



$$x = \begin{bmatrix} u \\ w \\ y \end{bmatrix} \begin{matrix} \text{controllable input} \\ \text{uncontrollable input} \\ \text{measured output} \end{matrix}$$

⋆ Can be generalized to a generic change of coordinates

Nothing new in power system modeling

	Exogenous		Endogenous
	Controllable	Uncontrollable	
PQ generation	p_i, q_i		v_i, θ_i
PQ load		p_i, q_i	v_i, θ_i
PV generation	p_i, v_i		q_i, θ_i
slack bus	v_1	θ_1	p_1, q_1

From $\mathcal{M}$ to $y = h(u; w)$

Chart for a $2n$-dimensional manifold:
invertible map between $\mathbb{R}^{2n}$ and a open subset of $\mathcal{M}$

Implicit function theorem

If a manifold is defined as

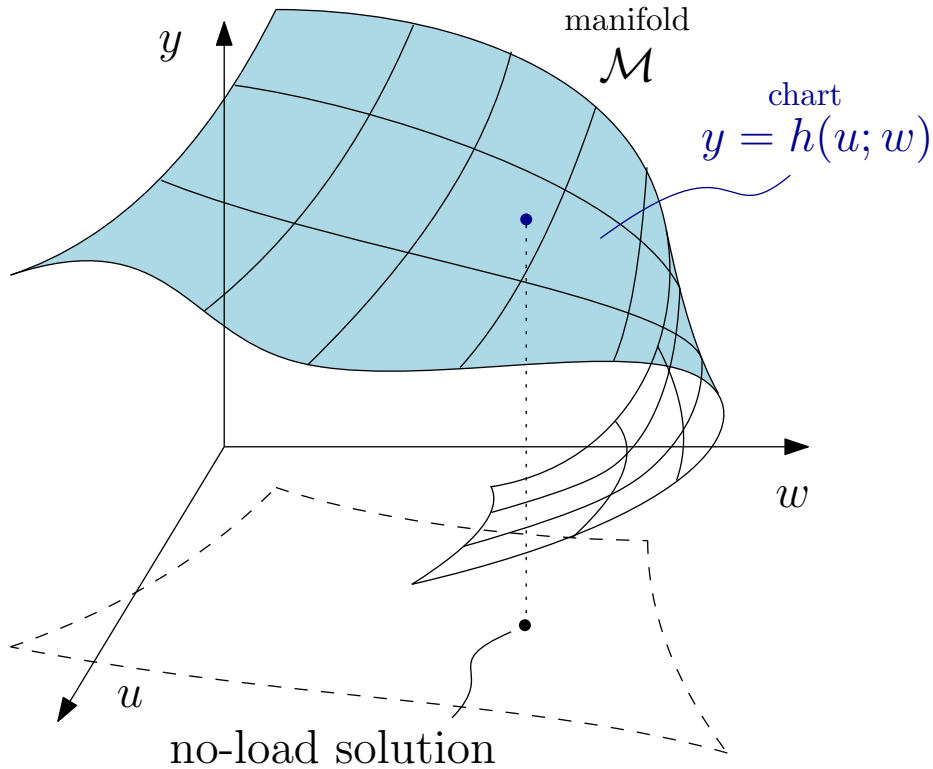
$$\mathcal{M} = \{(u, w, y) \,|\, F(u, w, y) = 0\}$$

then there exists a continuously differentiable function $y = h(u, w)$ such that

$$F(u, w, h(u, w)) = 0$$

in the open subset where

$\nabla_y F(u, w, y)$ is invertible

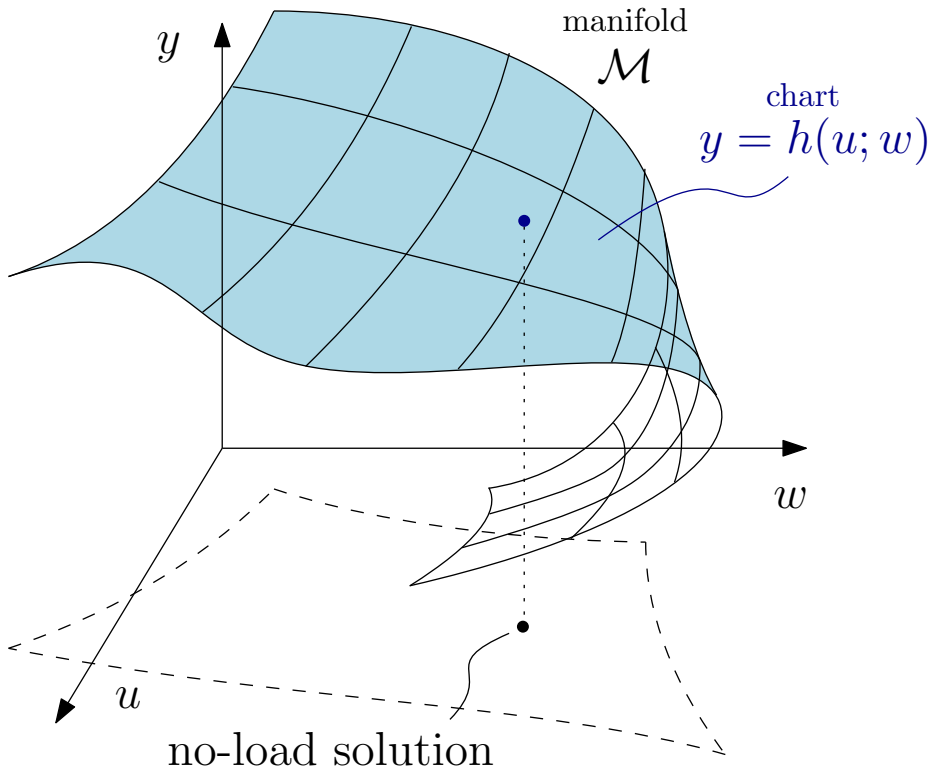


From $\mathcal{M}$ to $y = h(u; w)$

$$\nabla_{u,w} h(u,w) = -\left(\nabla_y F(u,w,y)\right)^{-1} \nabla_{u,w} F(u,w,y)$$

$\nabla_y F(u,w,y)$ is known as the power flow Jacobian and connected to

- power flow solvability
- voltage collapse



High-voltage PFM

Largest connected component of $\mathcal{M}$ that contains the **no-load solution** and where $\nabla_y F(u,w,y)$ is invertible

High-voltage PFM

- The high-voltage PFM $\mathcal{M}_{\text{high}}$ can rarely be derived in closed form
  - 2-bus example and little else
- **Inner approximations** are available, but they are usually **conservative** (see day 1)

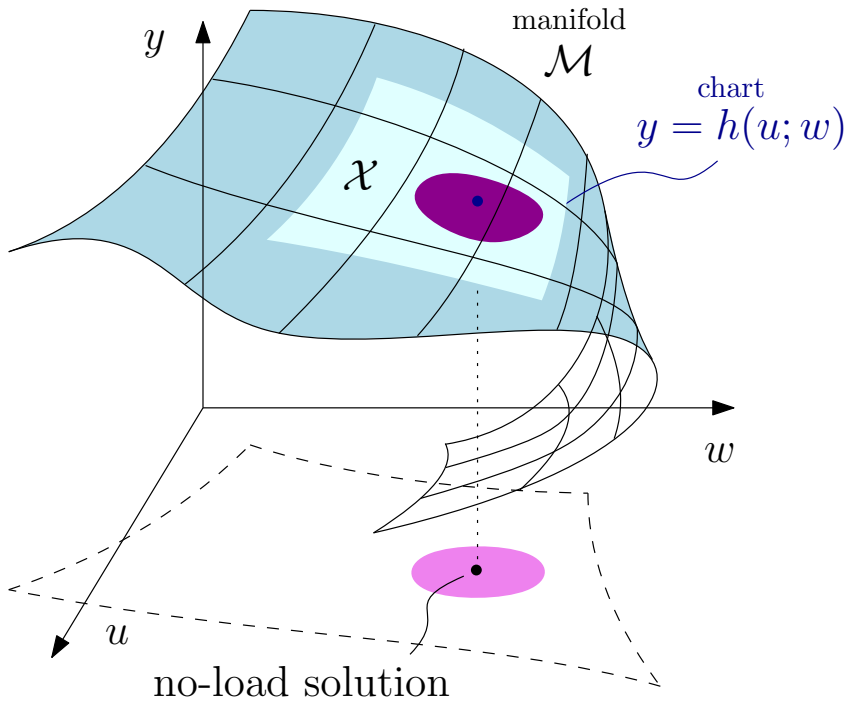
High-voltage PFM

- The high-voltage PFM $\mathcal{M}_{\text{high}}$ can rarely be derived in closed form
  - 2-bus example and little else
- **Inner approximations** are available, but they are usually **conservative** (see day 1)

Running assumption

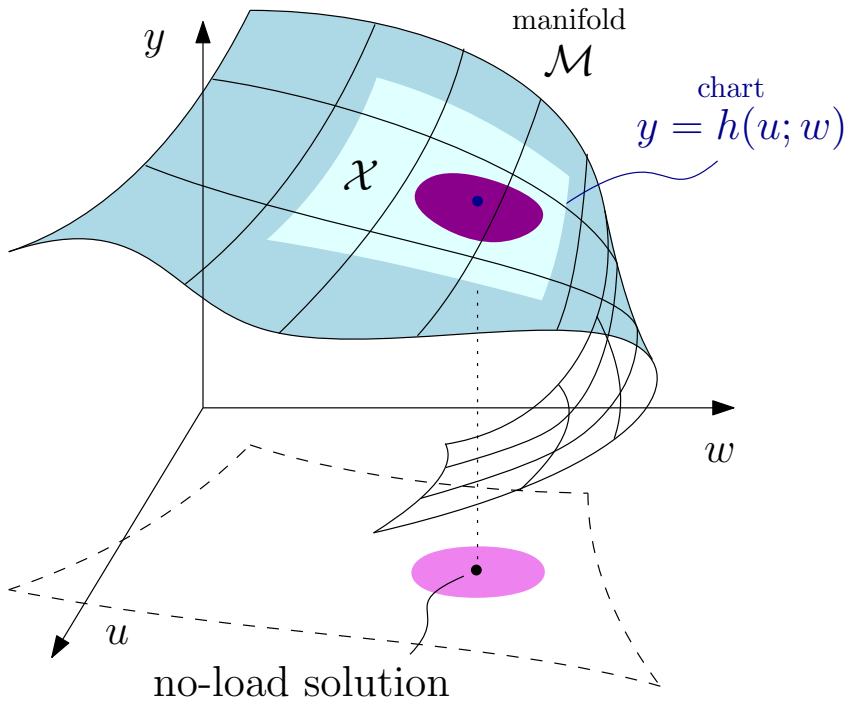
The operational constraints guarantee that the state of the grid belongs to the high-voltage region, i.e.

$$\mathcal{X} \subset \mathcal{M}_{\text{high}}$$



High-voltage PFM

- The high-voltage PFM $\mathcal{M}_{\text{high}}$ can rarely be derived in closed form
  - 2-bus example and little else
- **Inner approximations** are available, but they are usually **conservative** (see day 1)



Running assumption

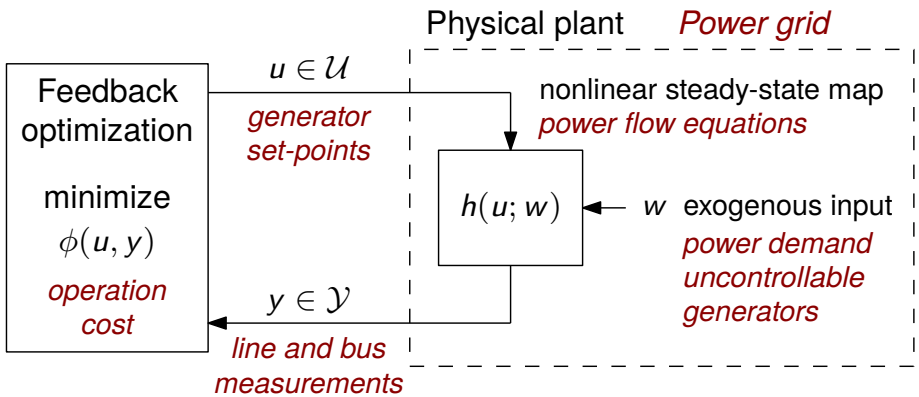
The operational constraints guarantee that the state of the grid belongs to the high-voltage region, i.e.

$$\mathcal{X} \subset \mathcal{M}_{\text{high}}$$

- It is easy to construct trajectories of $\dot{x} = f(x)$ that leave $\mathcal{M}_{\text{high}}$ in finite time
- This assumption is typically true **in nominal conditions**
- The assumption is often false during **contingencies**
- **A great open problem!**

Feedback optimization

Reformulated setup:



Feedback OPF

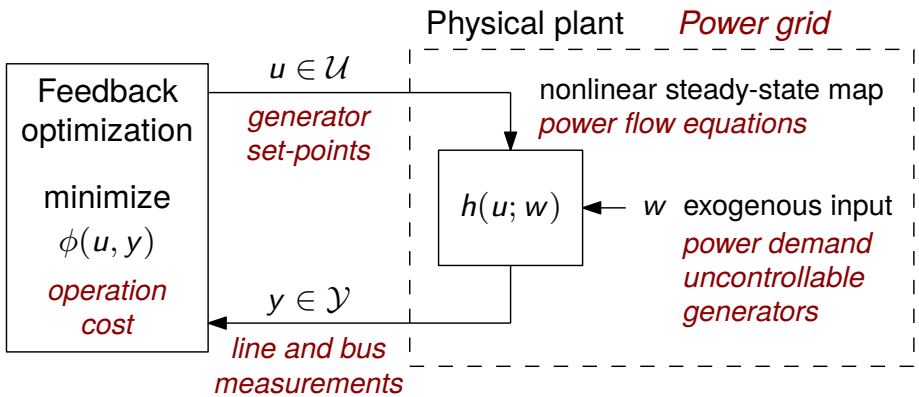
$$\begin{aligned} \text{minimize}_{u,y} \quad & \phi(u, y) \\ \text{subject to} \quad & y \in \mathcal{Y} \\ & u \in \mathcal{U} \\ & y = h(u; w) \end{aligned}$$

that is,

$$\min \phi(x), x \in \mathcal{X} = \mathcal{U} \times \{w\} \times \mathcal{Y}$$

Feedback optimization

Reformulated setup:



Feedback OPF

$$\begin{aligned} \text{minimize}_{u,y} \quad & \phi(u, y) \\ \text{subject to} \quad & y \in \mathcal{Y} \\ & u \in \mathcal{U} \\ & y = h(u; w) \end{aligned}$$

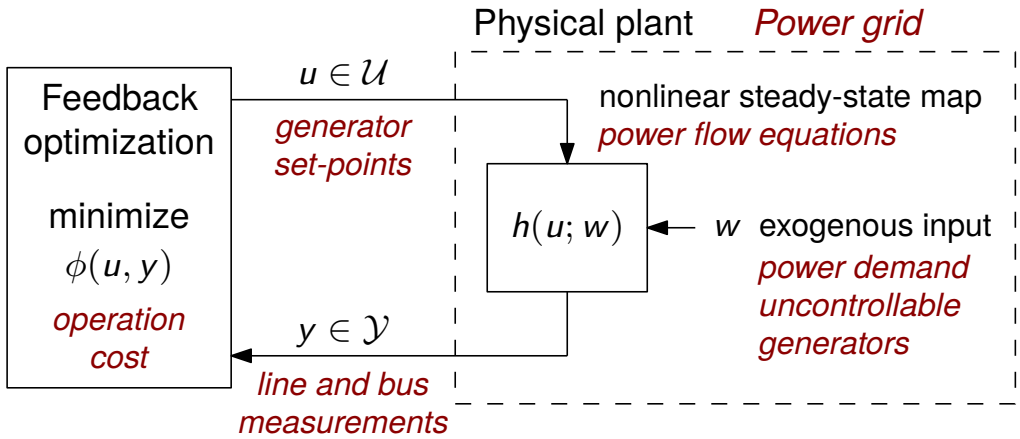
that is,

$$\min \phi(x), x \in \mathcal{X} = \mathcal{U} \times \{w\} \times \mathcal{Y}$$

Control design specifications:

- **Input saturation:** $u \in \mathcal{U}$ at all times (generation capacity, reserves)
- **Output feasibility:** $y \in \mathcal{Y}$ (voltage constraints, line flow constraints)
- **Optimality:** closed-loop dynamics $\dot{x} = f(x)$ converge to the OPF solution

Steady-state optimization



Optimization perspective

Analysis and design of algorithms with the tools of dynamical systems **but we implement them via the physics**

Control perspective

Feedback systems interpreted as solvers of a specific optimization problem **but we require general objective + constraints**

Keywords: Self-optimizing control, steady-state optimization, economic MPC, online optimization, steady-state control, real-time optimization, ...

Domains: Process control, congestion control in data networks, ...

Design of feedback optimizers

Borrow ideas from **iterative optimization algorithms** for **non-convex optimization** and interpret these algorithms as dynamical systems

- Gradient Flows

  [Brockett, 1991], [Bloch et al., 1992], [Helmke & Moore, 1994], ...
- Interior-point methods

  [Karmarkar, 1984], [Khachian, 1979], [Faybusovich, 1992], ...
- Acceleration & Momentum methods

  [Su et al., 2014], [Wibisono et al, 2016], [Krichene et al., 2015], [Wilson et al., 2016], [Lessard et al., 2016], ...
- Saddle-Point Flows

  [Arrow et al., 1958], [Kose, 1956], [Feijer & Paganini, 2010], [Cherukuri et al., 2017], [Holding & Lestas, 2014], [Cortés & Niederländer, 2018], [Qu & Li, 2018], ...

Design of feedback optimizers

Borrow ideas from **iterative optimization algorithms** for **non-convex optimization** and interpret these algorithms as dynamical systems

- Gradient Flows

  [Brockett, 1991], [Bloch et al., 1992], [Helmke & Moore, 1994], ...
- Interior-point methods

  [Karmarkar, 1984], [Khachian, 1979], [Faybusovich, 1992], ...
- Acceleration & Momentum methods

  [Su et al., 2014], [Wibisono et al, 2016], [Krichene et al., 2015], [Wilson et al., 2016], [Lessard et al., 2016], ...
- Saddle-Point Flows

  [Arrow et al., 1958], [Kose, 1956], [Feijer & Paganini, 2010], [Cherukuri et al., 2017], [Holding & Lestas, 2014], [Cortés & Niederländer, 2018], [Qu & Li, 2018], ...

Claim: In continuous-time, most algorithms reduce to either (projected) **gradient flows** (w/o momentum) or (projected) **saddle-point** flows.

Outline

- Input-output steady-state model
- Examples of feedback optimization design
- Stability of the dynamic interconnection

Example 1

$$
\begin{aligned}
\text{minimize}_{u,y} \quad & \phi(u, y) \\
\text{subject to} \quad & y \in \mathcal{Y} \quad \text{output constraints} \\
& u \in \mathcal{U} \quad \text{input saturation} \\
& y = h(u; w) \quad \text{power flow equations}
\end{aligned}
$$

Consider a **projected gradient-descent flow** in the u space $(\mathbb{R}^m)$.
We need:

- gradient $\nabla_u \tilde{\phi}(u)$, where $\tilde{\phi}(u) := \phi(u, h(u; w))$
- projection on the tangent cone of the feasible region

Example 1

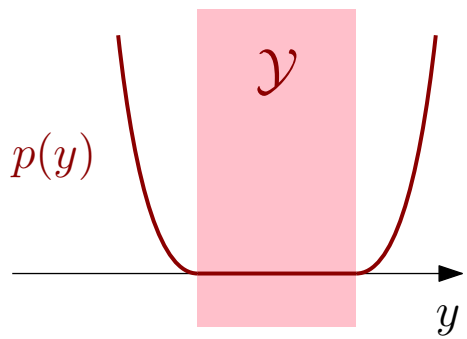
$$
\begin{aligned}
\text{minimize}_{u,y} \quad & \phi(u, y) \\
\text{subject to} \quad & y \in \mathcal{Y} \quad \text{output constraints} \\
& u \in \mathcal{U} \quad \text{input saturation} \\
& y = h(u; w) \quad \text{power flow equations}
\end{aligned}
$$

Consider a **projected gradient-descent flow** in the u space $(\mathbb{R}^m)$.
We need:

- gradient $\nabla_u \tilde{\phi}(u)$, where $\tilde{\phi}(u) := \phi(u, h(u; w))$
- projection on the tangent cone of the feasible region

Easy case: replace $y \in \mathcal{Y}$ via a penalty function $p(y)$

$$
\begin{aligned}
\text{minimize}_{u} \quad & \phi(u, y) + p(y) \\
\text{subject to} \quad & u \in \mathcal{U} \\
& y = h(u; w)
\end{aligned}
$$



Example 1

$\mathcal{Y} \to$ Penalty function (Hauswirth 2017, Tang 2017, Mazzi 2018, ...)

Gradient descent flow in the input space (via chain-rule)

$$\dot{u} = \Pi_{\mathcal{U}}\Big[-\nabla_u \phi(u,y) - \nabla h(u;w)^\top \nabla_y \phi(u,y) - \nabla h(u;w)^\top \nabla p(y)\Big]$$

Does this control law (projected gradient flow in the input space) induce a projected gradient flow on the manifold?

Yes. To convince yourself, look at the unprojected case. The direction

$$\delta_u = -\nabla_u \phi(u,y) - \nabla h(u;w)^\top \nabla_y \phi(u,y)$$

induces a direction in y

$$\delta_y = \nabla h(u;w) w_u = -\nabla h(u;w)\nabla_u \phi(u,y) - \nabla h(u;w)\nabla h(u;w)^\top \nabla_y \phi(u,y)$$

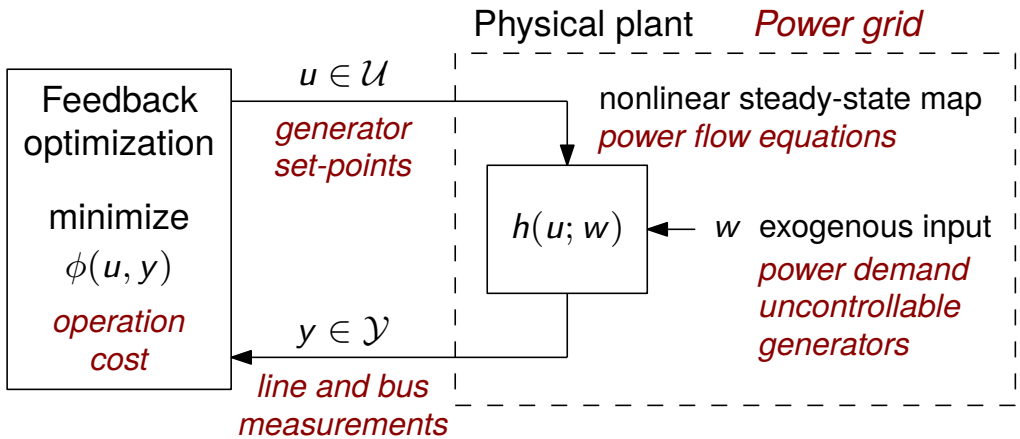
which corresponds, in the ambient space, to

$$\delta = \begin{bmatrix} I & \nabla h(u;w)^\top \\ \nabla h(u;w) & \nabla h(u;w)\nabla h(u;w)^\top \end{bmatrix} \nabla \phi(x)$$

i.e., a gradient flow in a different metric.

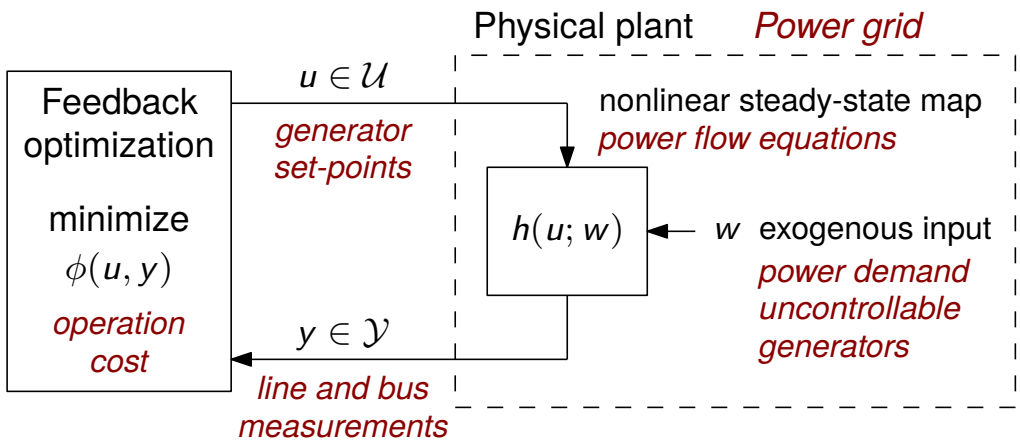
Example 1

$$\dot{u} = \Pi_{\mathcal{U}}\Big[-\nabla_u \phi(u,y) - \nabla h(u;w)^\top \nabla_y \phi(u,y) - \nabla h(u;w)^\top \nabla p(y) \Big]$$



Example 1

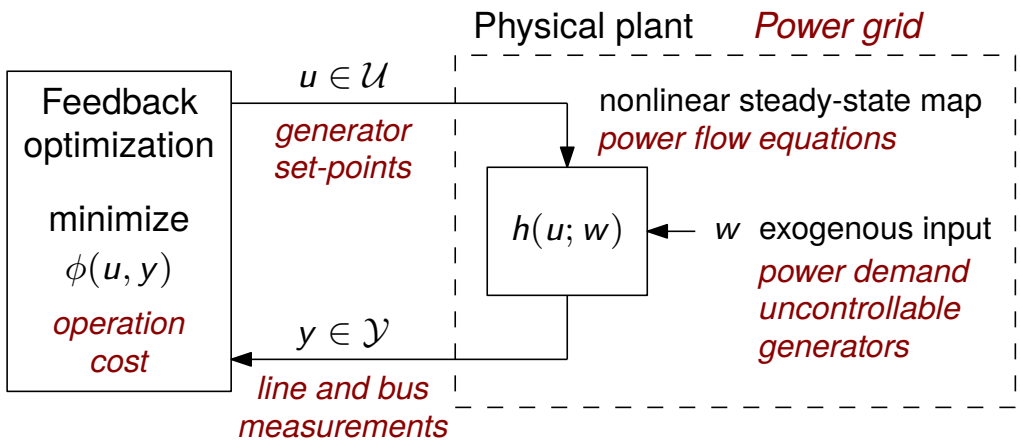
$$\dot{u} = \Pi_{\mathcal{U}}\Big[-\nabla_u \phi(u,y) - \nabla h(u;w)^\top \nabla_y \phi(u,y) - \nabla h(u;w)^\top \nabla p(y) \Big]$$



1. Does the controller needs to compute $h(u;w)$?
   Remember: solving $h(u;w)$ corresponds to solving PF equations

Example 1

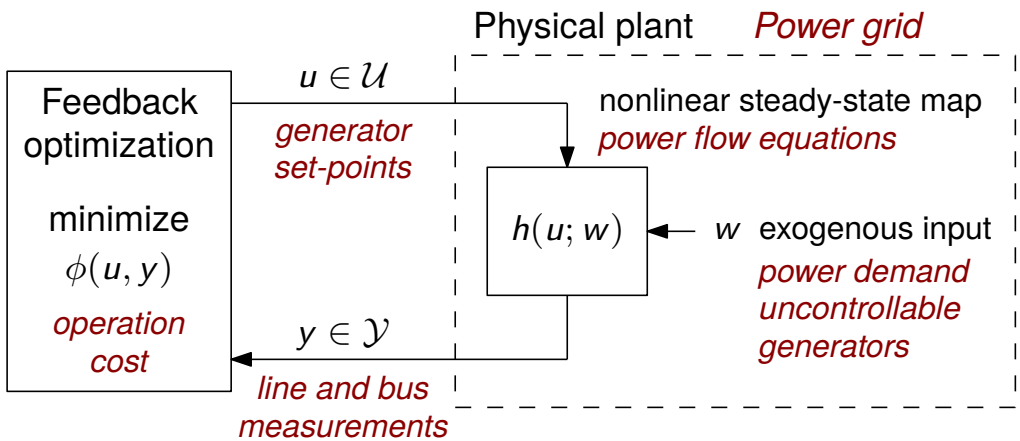
$$\dot{u} = \Pi_{\mathcal{U}}\Big[-\nabla_u \phi(u,y) - \nabla h(u;w)^\top \nabla_y \phi(u,y) - \nabla h(u;w)^\top \nabla p(y)\Big]$$



1. Does the controller needs to compute $h(u;w)$?
   Remember: solving $h(u;w)$ corresponds to solving PF equations
2. What does the controller need to know about the grid? (model)

Example 1

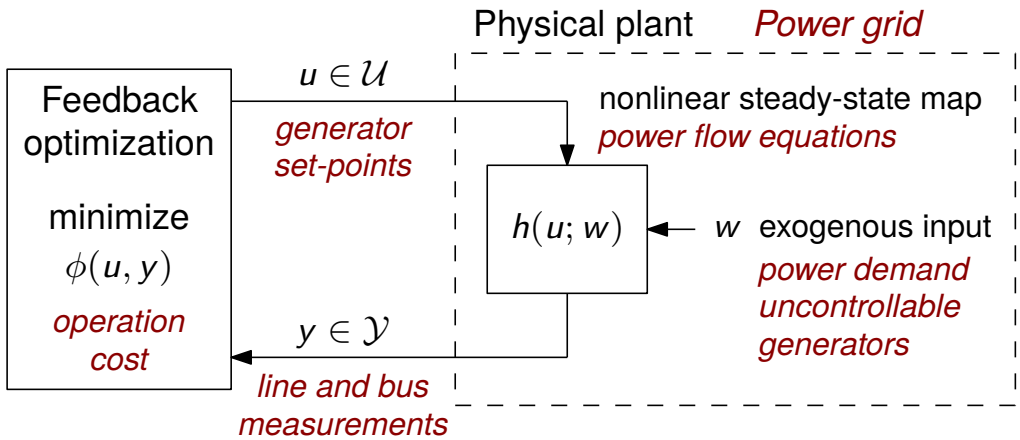
$$\dot{u} = \Pi_{\mathcal{U}}\Big[-\nabla_u \phi(u,y) - \nabla h(u;w)^\top \nabla_y \phi(u,y) - \nabla h(u;w)^\top \nabla p(y)\Big]$$



1. Does the controller needs to compute $h(u;w)$?
   Remember: solving $h(u;w)$ corresponds to solving PF equations
2. What does the controller need to know about the grid? (model)
3. How would the control algorithm look like?
   What is the complexity of projecting the gradient on $\mathcal{U}$?

Example 1

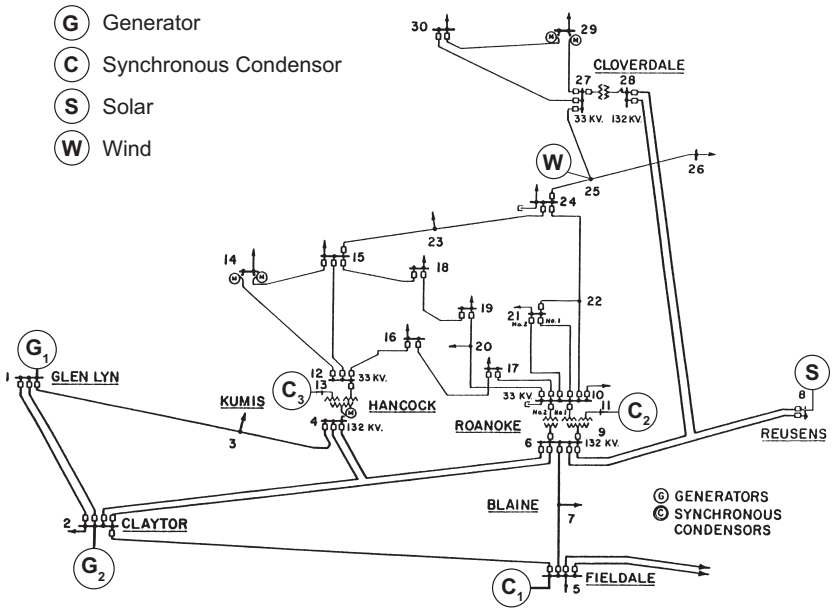
$$\dot{u} = \Pi_{\mathcal{U}}\Big[-\nabla_u \phi(u,y) - \nabla h(u;w)^\top \nabla_y \phi(u,y) - \nabla h(u;w)^\top \nabla p(y) \Big]$$



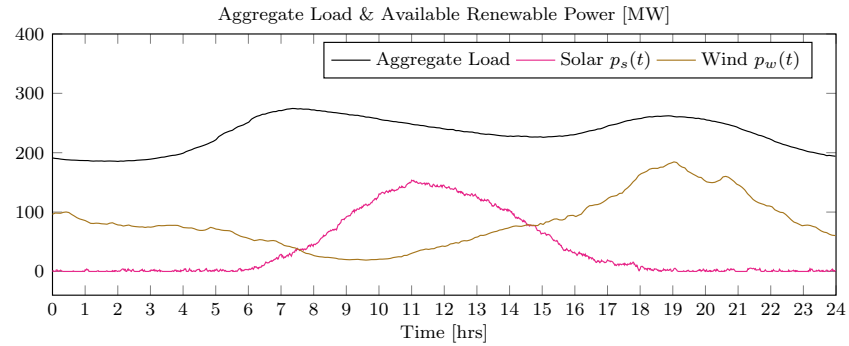
1. Does the controller needs to compute $h(u;w)$?
   Remember: solving $h(u;w)$ corresponds to solving PF equations
2. What does the controller need to know about the grid? (model)
3. How would the control algorithm look like?
   What is the complexity of projecting the gradient on $\mathcal{U}$?
4. Can you make the violation of the constraint $\mathcal{Y}$ arbitrarily small? How?

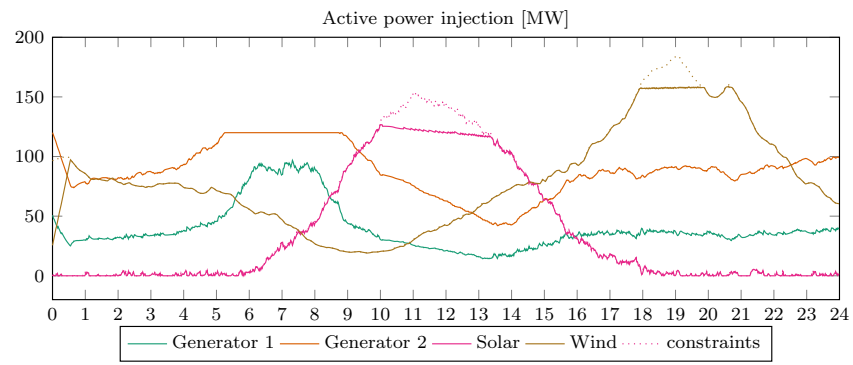
Example 1: simulation

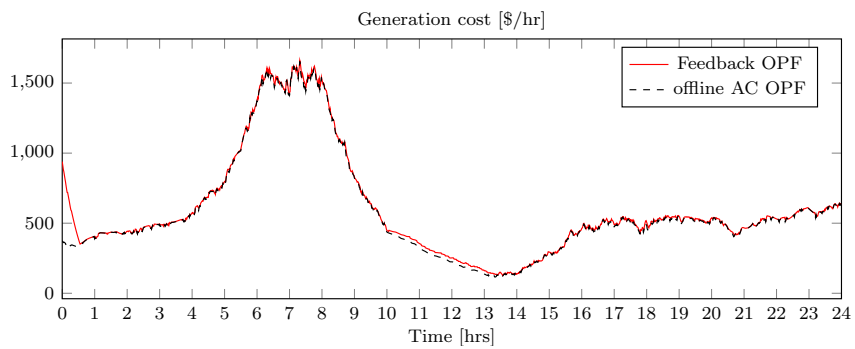
A. Hauswirth, A. Zanardi, S. Bolognani, F. Dörfler, G. Hug
Online Optimization in Closed Loop on the Power Flow Manifold
IEEE PES PowerTech, 2017.



	Cost		Constraints				
	$a^{(1)}$	$a^{(2)}$	$\underline{p}_i$	$\overline{p}_i$	$\underline{q}_i$	$\overline{q}_i$	Type
Generator 1	0.1	0.9	0	150	-40	50	slack
Generator 2	0.04	0.5	0	120	-40	50	PV
Condensor 1	0	0	0	0	-40	50	PV
Condensor 2	0	0	0	0	-40	50	PV
Condensor 3	0	0	0	0	-40	50	PV
Solar	0	0	0	$\overline{p}_s(t)$	-40	50	PQ
Wind	0	0	0	$\overline{p}_w(t)$	-40	50	PQ

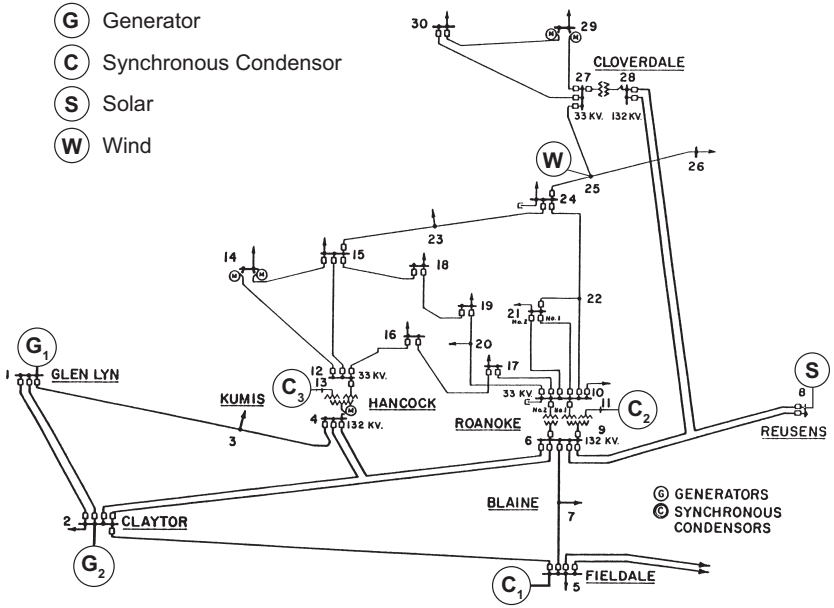






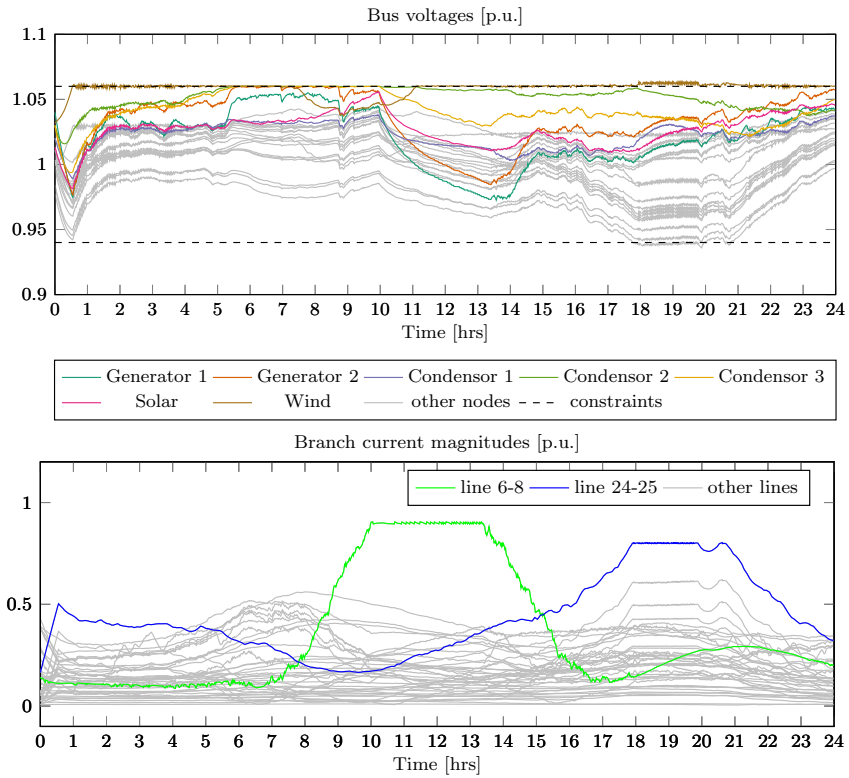
Example 1: simulation

A. Hauswirth, A. Zanardi, S. Bolognani, F. Dörfler, G. Hug
Online Optimization in Closed Loop on the Power Flow Manifold
IEEE PES PowerTech, 2017.



	Cost		Constraints				
	$a^{(1)}$	$a^{(2)}$	$\underline{p}_i$	$\overline{p}_i$	$\underline{q}_i$	$\overline{q}_i$	Type
Generator 1	0.1	0.9	0	150	-40	50	slack
Generator 2	0.04	0.5	0	120	-40	50	PV
Condensor 1	0	0	0	0	-40	50	PV
Condensor 2	0	0	0	0	-40	50	PV
Condensor 3	0	0	0	0	-40	50	PV
Solar	0	0	0	$\overline{p}_s(t)$	-40	50	PQ
Wind	0	0	0	$\overline{p}_w(t)$	-40	50	PQ

Output constraints



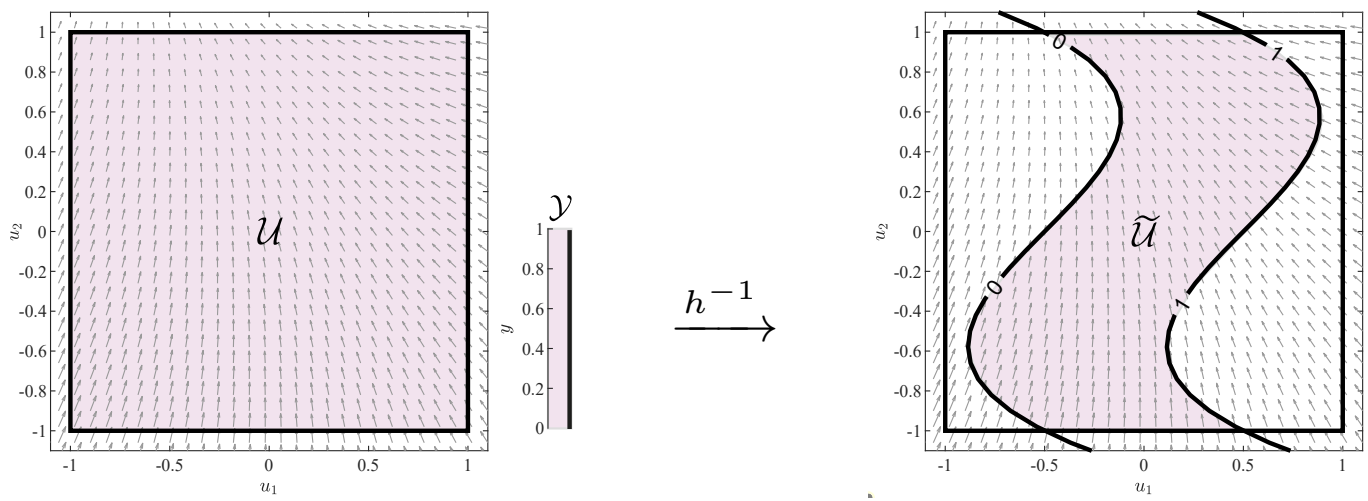
Example 2

$$
\begin{aligned}
\text{minimize}_{u,y} \quad & \phi(u,y) \\
\text{subject to} \quad & y \in \mathcal{Y} \quad \text{output constraints} \\
& u \in \mathcal{U} \quad \text{input saturation} \\
& y = h(u;w) \quad \text{power flow equations}
\end{aligned}
$$

Instead of replacing the output constraint $\mathcal{Y}$ with a penalty function, we consider its **pre-image** in the space of inputs.

$$\tilde{\mathcal{U}} = \mathcal{U} \cap h^{-1}(\mathcal{Y})$$

Example: $u \in \mathbb{R}^2$, $y \in \mathbb{R}$



Example 2

Projected gradient descent (Hauswirth 2016, Häberle 2020, ...)

Projection on the input and output constraints

$$\dot{u} = \Pi_{\tilde{\mathcal{U}}}\Big[-\nabla_u \phi(u,y) - \nabla h(u;w)^\top \nabla_y \phi(u,y)\Big]$$

How do we project on the tangent cone of $\tilde{\mathcal{U}}$?

- do we need to compute $h(u,w)$?
- do we need to invert $h(u,w)$?

Hint: Assume $\mathcal{U} := \{u \in \mathbb{R}^p \mid Au \leq b\}, \mathcal{Y} := \{y \in \mathbb{R}^n \mid Cy \leq d\}$

Consider the direction v. For v to belong to $T_{\tilde{\mathcal{U}}}(u)$ we need

$$\lim_{\alpha \to 0} A(u + \alpha v) \leq b$$

$$\lim_{\alpha \to 0} C(h(u + \alpha v; w)) \leq d$$

which are satisfied when v belongs to the cone

$$\mathcal{V} = \{v \mid Av \leq 0,\ C\nabla h(u;w)^\top v \leq 0\}$$

Projected gradient flow via QP

Continuous-time flow

$$\dot{u} = \Pi_{\tilde{\mathcal{U}}}\Big[-\nabla_u \Phi(u,y) - \underbrace{\nabla h(u;w)'}_{\text{model}} \nabla_y \Phi(u,y)\Big]$$

Assumption

$$\mathcal{U} := \{u \in \mathbb{R}^p \mid Au \le b\}$$
$$\mathcal{Y} := \{y \in \mathbb{R}^n \mid Cy \le d\}$$

Discrete-time approximation

$$u^+ = u + \alpha \delta u$$

$$\begin{aligned} \delta u := \quad & \arg\min_v \quad \| v - (-\nabla_u \Phi(u,y) - \nabla h(u;w)' \nabla_y \Phi(u,y)) \|^2 \\ & \text{subject to} \quad A(u + \alpha v) \le b \\ & \qquad\qquad \underbrace{C(y + \alpha \nabla h(u;w)' v) \le d}, \end{aligned}$$

1st order approx of $h^{-1}(\mathcal{Y})$ centered at the measurement y

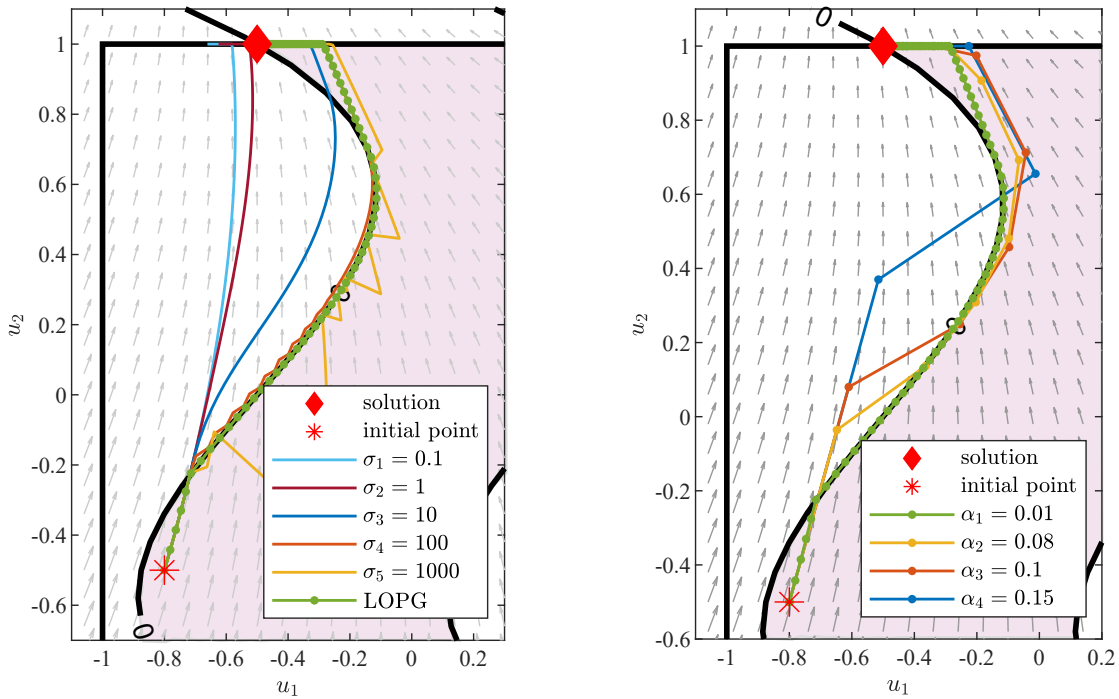
Theorem: (Häberle et al., IEEE Control Systems Letters, 2021)
LICQ + Lipschitz + differentiability + small α $\quad\rightarrow\quad$ global convergence to the set of local minima

Example 2

Projection on the input and output constraints come at no extra cost

- same computational complexity
- same model information
- numerically more stable when **discretized**

Penalty **Fully-projected gradient**



V. Häberle, V., A. Hauswirth, L. Ortmann, S. Bolognani, F. Dörfler
Non-Convex Feedback Optimization with Input and Output Constraints
IEEE Control Systems Letters, 5(1), 2021.

Example 3

$$\begin{aligned} \text{minimize}_{u,y} \quad & \phi(u,y) \\ \text{subject to} \quad & y \in \mathcal{Y} \quad \text{output constraints} \\ & u \in \mathcal{U} \quad \text{input saturation} \\ & y = h(u;w) \quad \text{power flow equations} \end{aligned}$$

Output constraint representation

$$\mathcal{Y} := \{y \mid g(y) \leq 0\}$$

Lagrangian

$$\mathcal{L}(u,y,\lambda) = \phi(u,y) + \lambda^\top g(y)$$

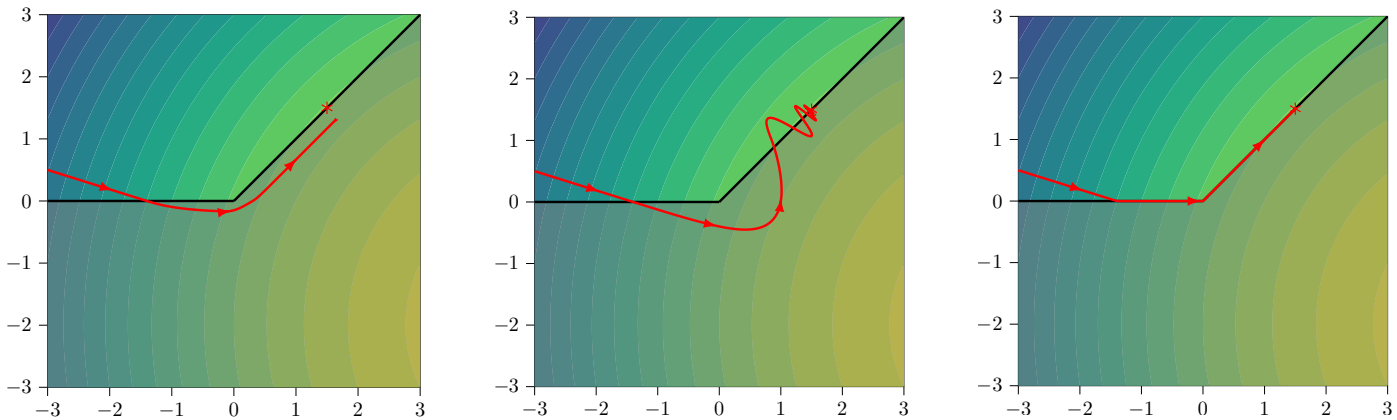
Saddle flow (Bolognani 2015, Dall'Anese 2018, Bernstein 2019, Colombino 2020, ...)

Primal descent / dual ascent $\rightarrow$ proportional-integral feedback law

$$\begin{cases} \dot{u} = \Pi_{\mathcal{U}} \Big[-\nabla_u \phi(u,y) - \underbrace{\nabla h(u;w)^\top}_{\text{model}} \nabla_y \phi(u,y) - \underbrace{\nabla h(u;w)^\top}_{\text{model}} \nabla g(y)^\top \lambda \Big] \\ \dot{\lambda} = \Pi_{\geq 0} \left[g(y) \right] \end{cases}$$

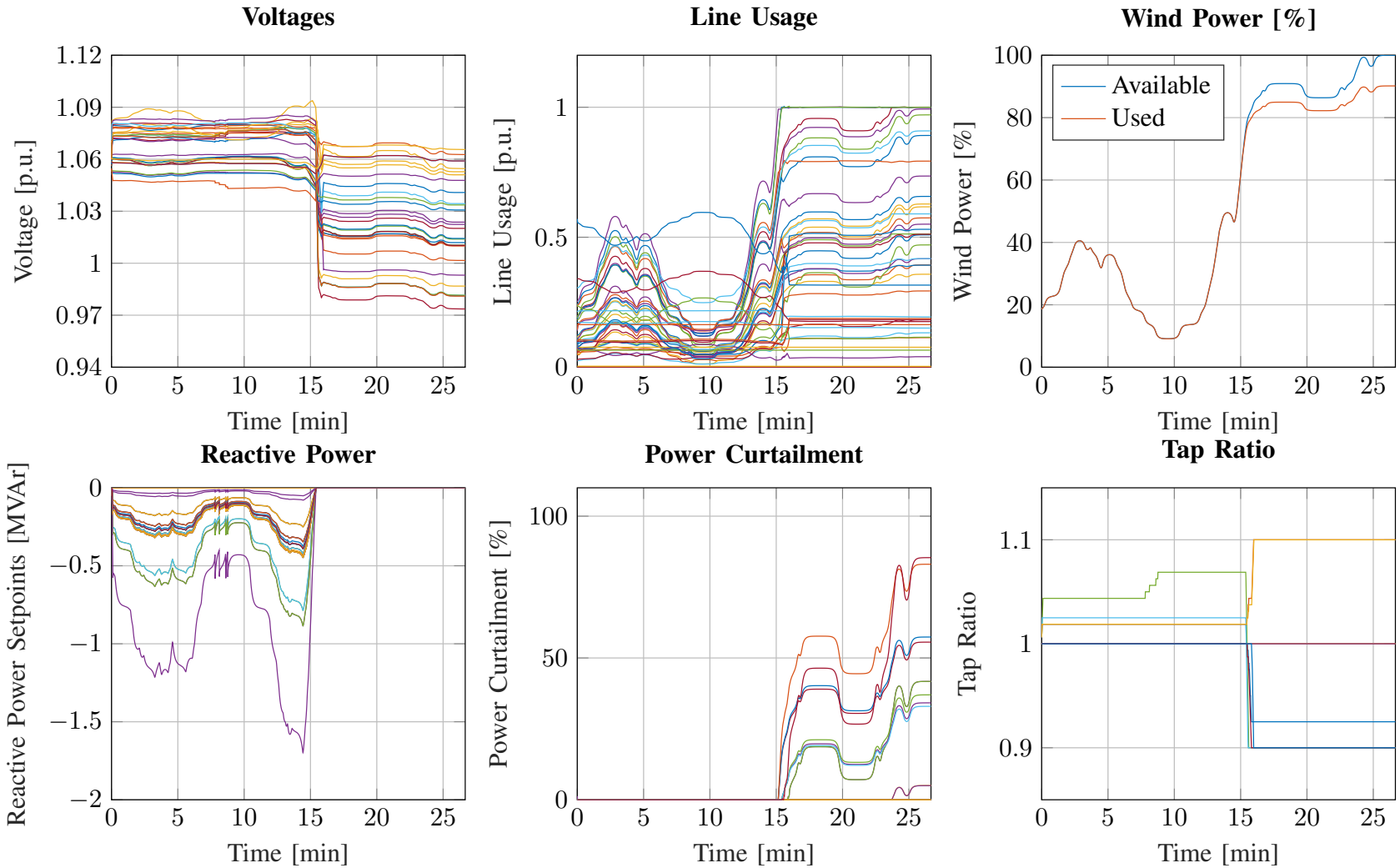
First-order methods

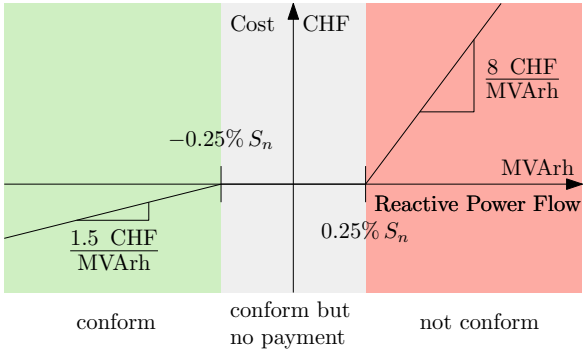
	Penalty	**(Augmented) saddle-flow**	**Projected gradient**
Feasibility	arbitrarily small violation	asymptotic feasibility	any-time feasibility
Controller	“proportional”	“(proportional)-integral”	quadratic programming
Stability	steep penalty limits speed	requires exp. stability	simple gain limit



All these methods require **gradient** information, and therefore **input output sensitivities** ∇h.

Numerical experiment: congestion of the French subtransmission grid



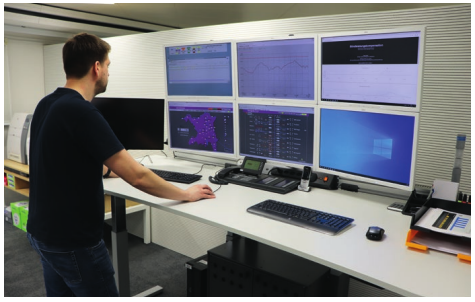


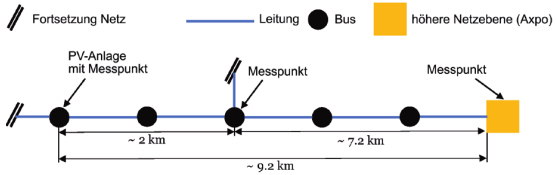
ELSEVIER
Electric Power Systems Research
Volume 189, December 2020, 106782

Experimental validation of feedback optimization in power distribution grids ☆

Lukas Ortmann, Adrian Hauswirth, Ivo Caduff, Florian Dörfler, Saverio Bolognani







TRL 1 >> **7 (Integrated pilot system demonstrated)**

arXiv > eess > arXiv:2305.06702

Electrical Engineering and Systems Science > Systems and Control

[Submitted on 11 May 2023 (v1), last revised 10 Aug 2023 (this version, v2)]

Deployment of an Online Feedback Optimization Controller for Reactive Power Flow Optimization in a Distribution Grid

Lukas Ortmann, Christian Rubin, Alessandro Scozzafava, Janick Lehmann, Saverio Bolognani, Florian Dörfler





Take home

- The available **actuators** and **measurement** determine the **inputs** and **outputs**
- A **chart** of the PFM gives us a convenient **input-output** representation of the grid **steady-state**
- We can tap into the tools of **iterative optimization** in order to
  - design an **optimizing feedback controller**
  - study their **convergence** to stationary points/minima
- In continuous time, we typically look at variations of either **gradient flows** or **saddle flows**, possibly with different metrics (**Newton flow**)
- The most challenging aspect is the projection on the **output constraints**

Outline

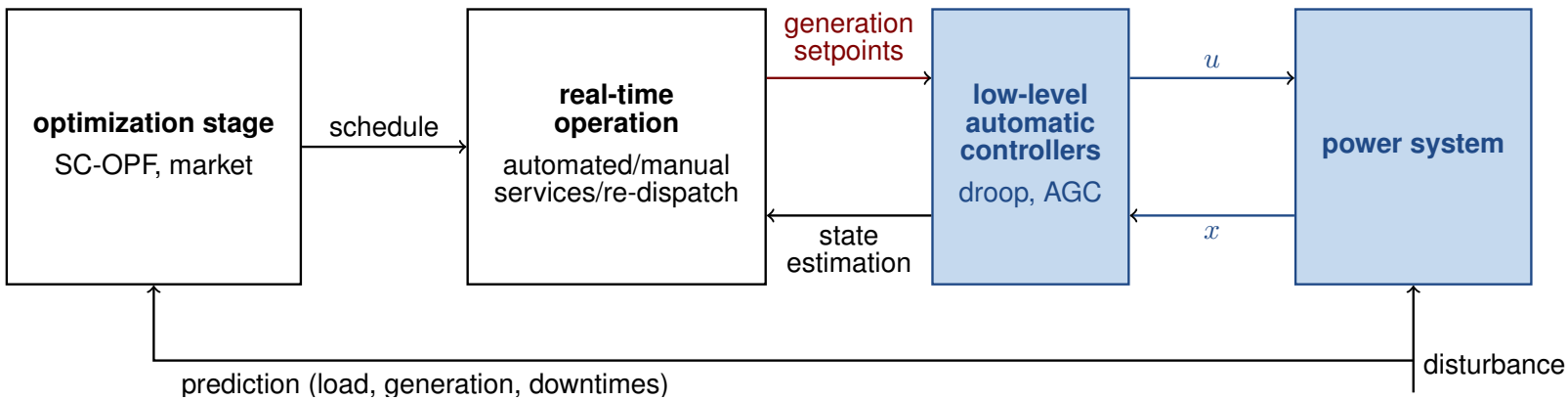
- Input-output steady-state model
- Examples of feedback optimization design
- Stability of the dynamic interconnection

Closing the loop

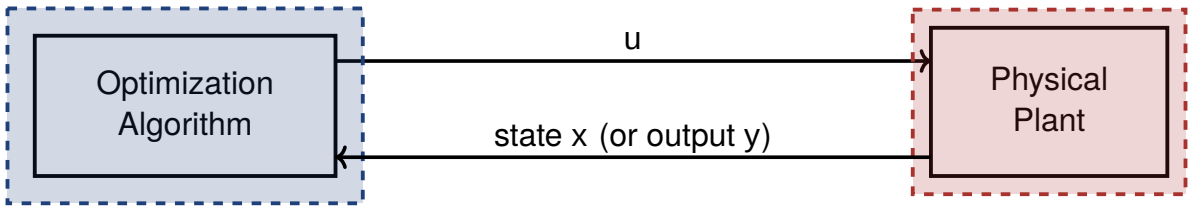
Up until now, we have studied convergence of **feedback optimization strategies** under the assumption that the power system is at **steady-state**, i.e.

$$y = h(u; w)$$

Is this going to work once it's connected to the power-system dynamics?



Example: unconstrained gradient flow



Optimization Dynamics

Gradient descent

$$\dot{u} = -\epsilon \left(\nabla_u \phi(u, x) + \nabla h(u)^\top \nabla_x \phi(u, x) \right)$$

Plant Dynamics

Exponentially stable system

$$\dot{x} = f(x, u)$$

with steady-state map $x = h(u)$

Assumption on the physical system

Assumption

For any input u there exist a Lyapunov function $W(x,u)$ and parameters α, β, γ, ζ such that

$$\begin{aligned} \alpha\|x-h(u)\|^2 \leq W(x,u) \leq \beta\|x-h(u)\|^2 \\ \dot{W}(x,u) \leq -\gamma\|x-h(u)\|^2 \\ \|\nabla_u W(x,u)\| \leq \zeta\|x-h(u)\| \end{aligned}$$

- The first two conditions are standard for exponentially stable systems
- The third condition is a Lipschitz-type condition

Interconnected system

$$\dot{x} = f(x, u)$$

$$\dot{u} = -\epsilon H(u)^\top \nabla \phi(x, u) \quad H := \begin{bmatrix} \nabla h(u) \\ I \end{bmatrix}$$

Proposition

Every **minimizer** $(x^\star, u^\star)$ of

$$\begin{aligned} \text{minimize}_u \quad & \phi(x, u) \\ \text{subject to} \quad & x = h(u) \end{aligned}$$

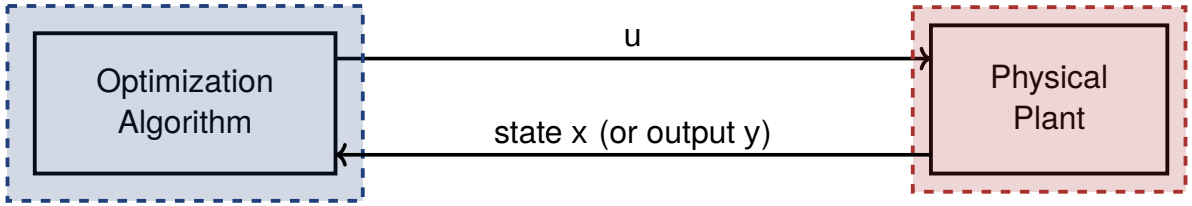
is an **equilibrium** for the interconnected system, and every equilibrium of the interconnected system is a critical point.

Proof:

In other words, the interconnection with the physical system preserves the same equilibria of the continuous-time optimization scheme in u.

How about stability?

Time-scale separation analysis



Key idea: Certify stability for a "sufficiently slow" optimization algorithms

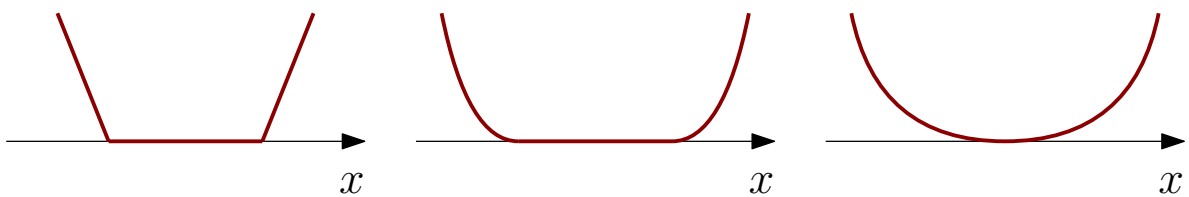
What is the "gain" of the gradient descent $\dot{u} = -\epsilon H(u)\nabla\phi(x,u)$?

Lipschitz condition

There exists an $L > 0$ such that

$$\|H(u)^\top(\nabla\phi(x,u) - \nabla\phi(z,u))\| \leq L\|x - z\|$$

for all x, z, u.



Main interconnected stability result

A. Hauswirth, S. Bolognani, G. Hug, F. Dörfler
Timescale Separation in Autonomous Optimization
IEEE Transactions on Automatic Control, 2021. → preprint

Theorem (Hauswirth 2019)

Assume

- **Exponentially stable** physical system with Lyapunov function W s.t.

$$\dot{W}(u,x) \leq -\gamma \|x - h(u)\|^2 \qquad \|\nabla_u W(u,x)\| \leq \zeta \|x - h(u)\| .$$

- $\phi(u,x)$ has **compact level sets** and L**-Lipschitz** gradient.

Then, all gradient descent trajectories converge to the set of KKT points if

$$\epsilon < \frac{\gamma}{\zeta L}$$

- Asymptotically stable equilibrium $\Rightarrow$ strict local minimizer
- Strict local minimizer $\Rightarrow$ stable equilibrium

If ϕ convex and $h(u)$ linear, then convergence to set of global minimizers.

Other gradient flows

Projected GD

$$\dot{u} = \Pi_{\mathcal{U}}[-\epsilon H(u)^\top \nabla \phi(u, x)]$$

$\rightarrow$ same bound:

$$\epsilon \leq \frac{\gamma}{\zeta L}$$

Other gradient flows

Projected GD

$$\dot{u} = \Pi_{\mathcal{U}}[-\epsilon H(u)^\top \nabla \phi(u, x)]$$

$\rightarrow$ same bound:

$$\epsilon \leq \frac{\gamma}{\zeta L}$$

General metric $Q(u) \geq 0$

$$\dot{u} = -Q(u) H(u)^\top \nabla \phi(u, x)$$

Stability is guaranteed if

$$\sup_{u \in \mathbb{R}^p} \|Q(u)\| < \frac{\gamma}{\zeta L}$$

Other gradient flows

Projected GD

$$\dot{u} = \Pi_{\mathcal{U}}[-\epsilon H(u)^\top \nabla \phi(u, x)]$$

$\rightarrow$ same bound:

$$\epsilon \leq \frac{\gamma}{\zeta L}$$

General metric $Q(u) \geq 0$

$$\dot{u} = -Q(u) H(u)^\top \nabla \phi(u, x)$$

Stability is guaranteed if

$$\sup_{u \in \mathbb{R}^p} \|Q(u)\| < \frac{\gamma}{\zeta L}$$

Newton GD

Choose $Q(u) = (\nabla^2 \phi(h(u; w), u))^{-1}$
(if ϕ μ-strongly convex and twice differentiable)
Stability is guaranteed if

$$\frac{L}{\mu} \leq \frac{\gamma}{\zeta}$$

$\rightarrow$ invariant under scaling of ϕ

Other gradient flows

Projected GD

$$\dot{u} = \Pi_{\mathcal{U}}[-\epsilon H(u)^\top \nabla \phi(u, x)]$$

$\rightarrow$ same bound:

$$\epsilon \leq \frac{\gamma}{\zeta L}$$

General metric $Q(u) \geq 0$

$$\dot{u} = -Q(u) H(u)^\top \nabla \phi(u, x)$$

Stability is guaranteed if

$$\sup_{u \in \mathbb{R}^p} \|Q(u)\| < \frac{\gamma}{\zeta L}$$

Newton GD

Choose $Q(u) = (\nabla^2 \phi(h(u; w), u))^{-1}$ (if ϕ μ-strongly convex and twice differentiable)
Stability is guaranteed if

$$\frac{L}{\mu} \leq \frac{\gamma}{\zeta}$$

$\rightarrow$ invariant under scaling of ϕ

The same analysis can be specialized to your preferred optimization flow

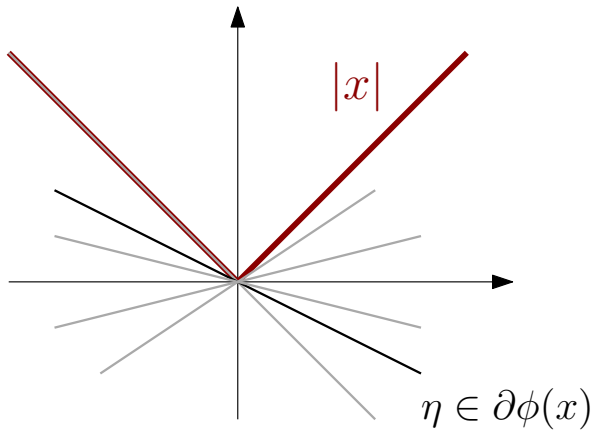
Same prescription: the optimization needs to be sufficiently slow compared to the physical system

In constrast to stability *tests*: e.g. IQC
Nelson & Mallada 2017

Subgradient flows

Quiz: does subgradient work?

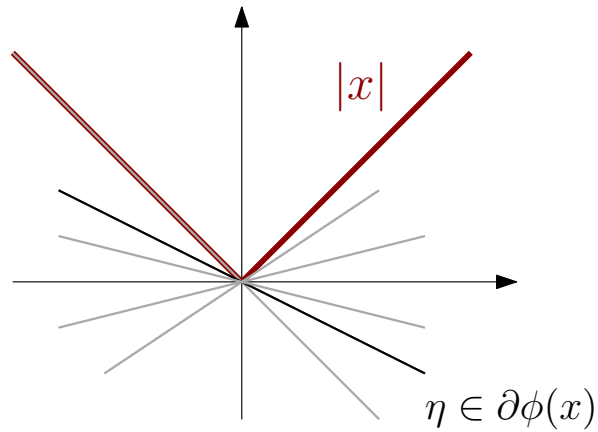
$$\dot{u} \in -H(u)^\top \partial\phi(x) = -H(u) \begin{cases} -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$



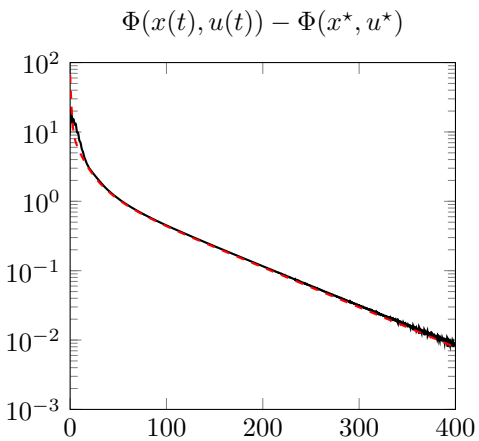
Subgradient flows

Quiz: does subgradient work?

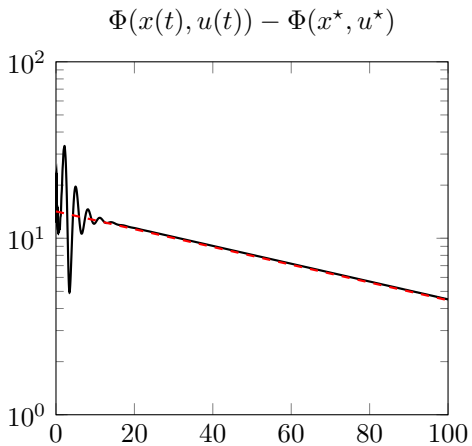
$$\dot{u} \in -H(u)^\top \partial\phi(x) = -H(u) \begin{cases} -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$



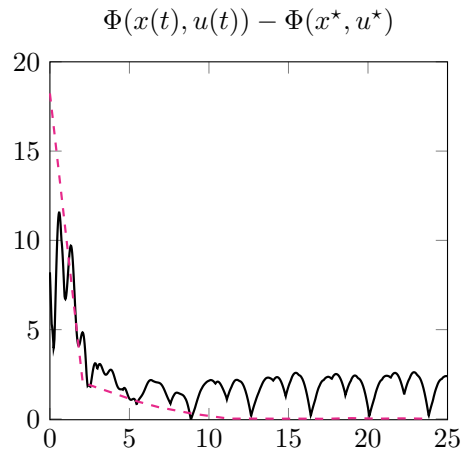
gradient flow



Newton flow



subgradient flow



Momentum-based controllers

See recent use in **machine learning** and **stochastic gradient descent**.
$\rightarrow$ continuous time:

$$\dot{u} = Q(u)z \qquad\qquad z: \text{ momentum variable}$$
$$\dot{z} = -\underbrace{D(u)}_{\text{damping}} z - Q(u)H(u)^\top \nabla\phi(x,u)$$

Same procedure:

- Lyapunov function for the optimization scheme + algebraic plant
  Try $V(u,z) = \phi(u,h(u)) + \frac{1}{2}z^\top z$
- Construct Lyapunov function for the interconnected system
  $(1-\delta)V(u,z) + \delta W(x,u)$
- Bound different terms of the Lie derivative using the assumptions on W
  $\dot{W}(x,u) \leq -\gamma\|x-h(u)\|^2, \quad \|\nabla_u W(x,u)\| \leq \zeta\|x-h(u)\|$
- Obtain a 2×2 quadratic form for the Lie derivative
- Find conditions to make this form negative definite
  Damping "bounded away" from zero $\Rightarrow$ no acceleration $D = \frac{d}{t}$*!*

Momentum-based controllers

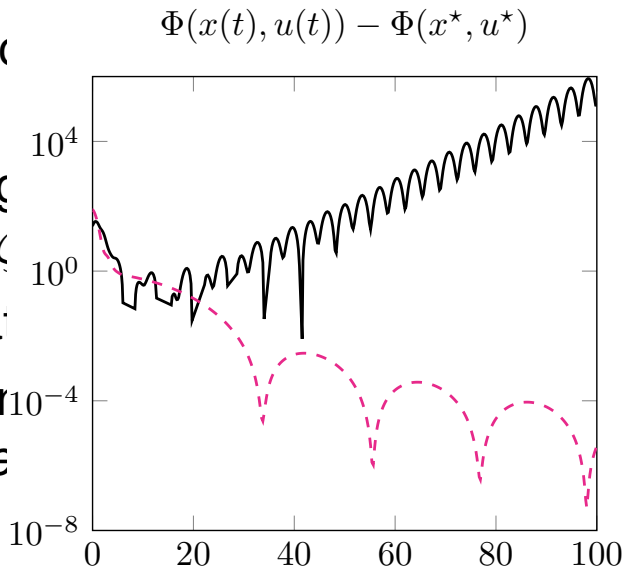
See recent use in **machine learning** and **stochastic gradient descent**.
$\rightarrow$ continuous time:

$$\dot{u} = Q(u)z \qquad\qquad z: \text{ momentum variable}$$
$$\dot{z} = -\underbrace{D(u)}_{\text{damping}} z - Q(u)H(u)^\top \nabla\phi(x,u)$$

Same procedure:

- Lyapunov function for the optimization scheme
  Try $V(u,z) = \phi(u,h(u)) + \frac{1}{2}z^\top z$
- Construct Lyapunov function for the interconnec
  $(1-\delta)V(u,z) + \delta W(x,u)$
- Bound different terms of the Lie derivative usin
  $\dot{W}(x,u) \leq -\gamma\|x-h(u)\|^2, \quad \|\nabla_u W(x,u)\| \leq$
- Obtain a 2×2 quadratic form for the Lie deriva
- Find conditions to make this form negative defi
  Damping "bounded away" from zero $\Rightarrow$ no acce

Accelerated gradient flow



General feedback optimization controllers

$$\dot{u} = \varepsilon g(h(u; w), u, z)$$
$$\dot{z} = \varepsilon k(h(u; w), u, z)$$

Theorem

- $(g(x, u, z), k(x, u, z))$ is L-Lipschitz in x
- $(g(h(u; w), u, z), k(h(u; w), u, z))$ is ℓ-Lipschitz
- $\exists$ **Lyapunov function $V(u, z)$ for the slow dynamics**

$$\dot{V}(u, z) \leq -\mu \|e(u, z)\|^2 \quad \|\nabla V(u, z)\| \leq \kappa \|e(u, z)\|$$

- $\exists$ Lyapunov function $W(x, u)$ for the plant

$$\dot{W}(x, u) \leq -\gamma \|x - h(u; w)\|^2, \quad \|\nabla_u W(x, u)\| \leq \zeta \|x - h(u; w)\|$$

Then, asymptotic stability is guaranteed if

$$\epsilon < \frac{\gamma}{\zeta L(1 + \frac{\kappa \ell}{\mu})}.$$

Saddle flow

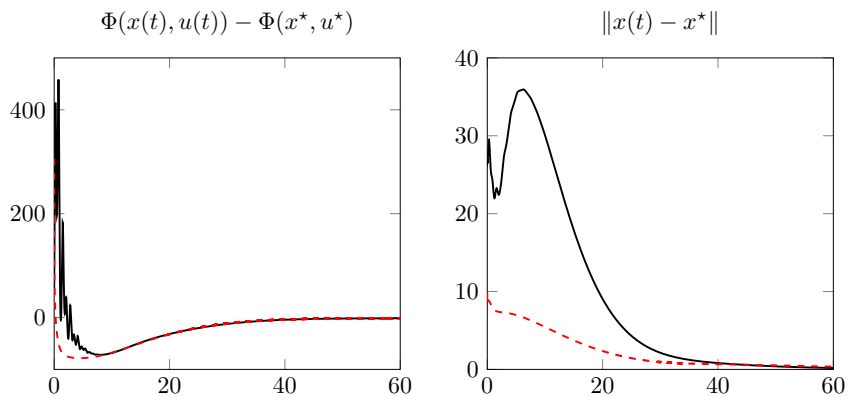
$$\begin{aligned} \text{minimize}_{x,u} \quad & \phi(x,u) \\ \text{subject to} \quad & Ax - b = 0 \\ & x = h(u) \end{aligned}$$

In this case, the **augmented saddle flow**

$$\dot{u} = -\epsilon \nabla_u \mathcal{L}(u,\lambda) \qquad \dot{\lambda} = \epsilon \nabla_\lambda \mathcal{L}(u,\lambda)$$

$$\mathcal{L}(u,\lambda) = \phi(h(u),u) + \lambda^\top (Ah(u) - b) + \frac{\sigma}{2} \|Ah(u) - b\|^2$$

is **exponentially stable** $\rightarrow$ stable interconnecton for **sufficiently small** ϵ



Exponential stability of general saddle-flows still an open problem! [Qu & Li, 2018]

Numerical example

Dynamic model:

- linearized **swing dynamics** (with primary frequency control)
- 1st-order **turbine-governor**
- DC power flow approximation (Kron-reduced)

$$\left.\begin{aligned} \dot{\theta} &= \omega \\ \dot{\omega} &= -M^{-1}\left(D\omega + \mathbf{B}\theta - p + p^L(t)\right) \\ \dot{p} &= -K\left(R^{-1}\omega + p - p^C\right) \end{aligned}\right\} \quad \begin{aligned} \dot{x} &= Ax + Bu + Qw \quad \text{where} \\ x &= \begin{bmatrix} \theta \\ \omega \\ p \end{bmatrix}, \; u = p^C, \; w = p^L(t) \end{aligned}$$

Outputs: frequency at node 1 + line flows

$$y = Cx \quad \text{where} \quad C = \begin{bmatrix} \mathbb{O} & 1 & 0 & \dots & 0 & \mathbb{O} \\ \mathbf{B}^\ell & & & \mathbb{O} & & \mathbb{O} \end{bmatrix}$$

Numerical example

Optimization problem:

- minimization of economic cost (DC OPF)
- constraints on power generation, line flows, and frequency deviation
- engineering limits treated as penalty functions

$$\begin{aligned} \underset{y,u}{\text{minimize}} \quad & \Phi(y,u) \\ \text{subject to} \quad & y = Hu + Qw \end{aligned}$$

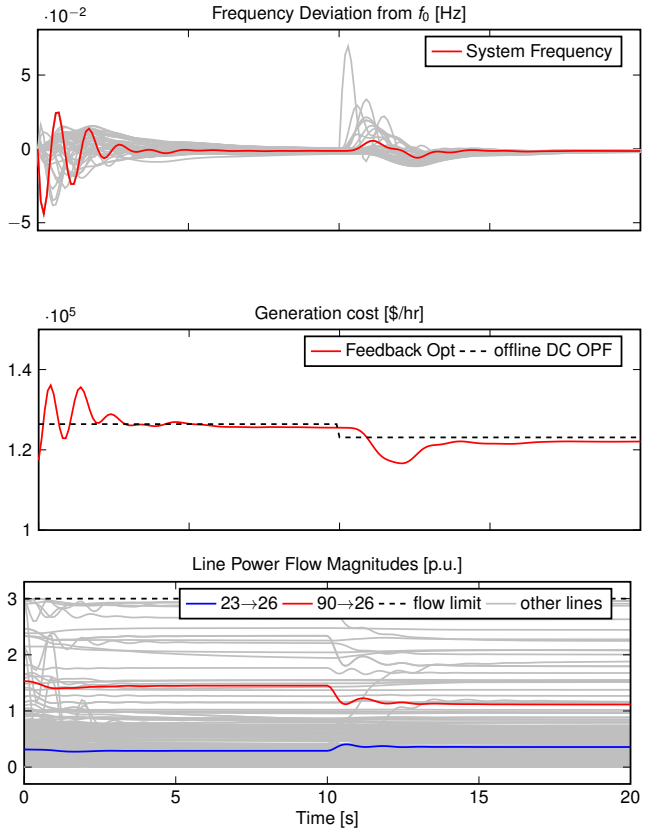
where $y = Hu + Qw$ is the steady-state input-output map and

$$\Phi(y,u) = \text{cost}(u) + \tfrac{1}{2}\| \max\{0, \underline{y} - y\}\|_{\Xi}^2 + \tfrac{1}{2}\| \max\{0, y - \overline{y}\}\|_{\Xi}^2$$

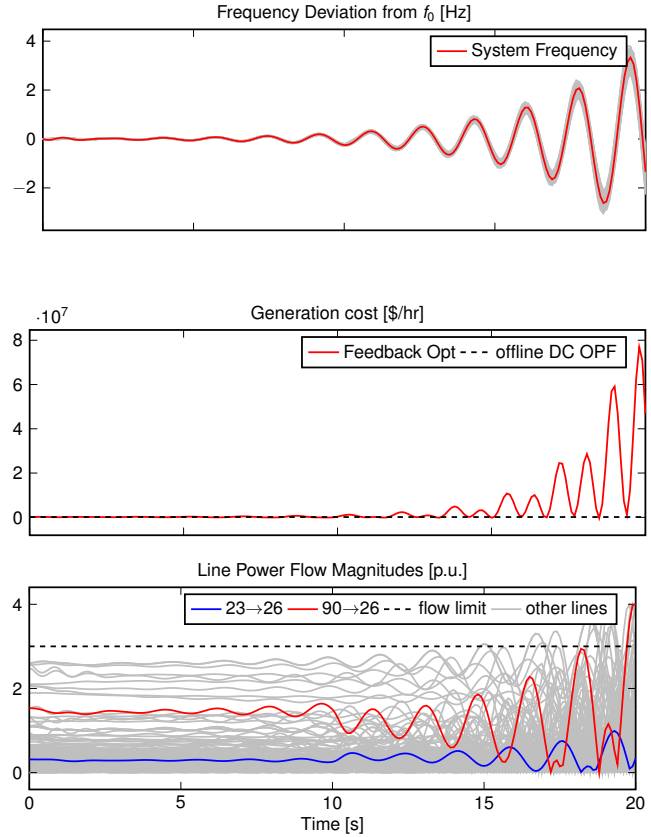
Numerical example

Simulation on IEEE 118-bus test case

still **stable for** $\varepsilon = 2\varepsilon^\star$



unstable for $\varepsilon = 10\varepsilon^\star$



Observed **factors of conservativeness** from **1.2 to 1000**, depending on penalty scaling.

Take home

- **Iterative optimization algorithms** can be implemented in closed loop with the physical system **(feedback optimization)**
- **Stability of the interconnection** of
  - feedback optimizer
  - physical system

  is not guaranteed, even if both of them are individually stable
- if the optimization scheme is **exponentially stable** and **sufficiently slow**, interconnected stability can be certified
- the **same analysis** can be repeated for many optimization schemes and follows similar steps

A. Hauswirth, Z. He, S. Bolognani, G. Hug, F. Dörfler
Optimization algorithms as robust feedback controllers
Annual Reviews in Control, 2024.