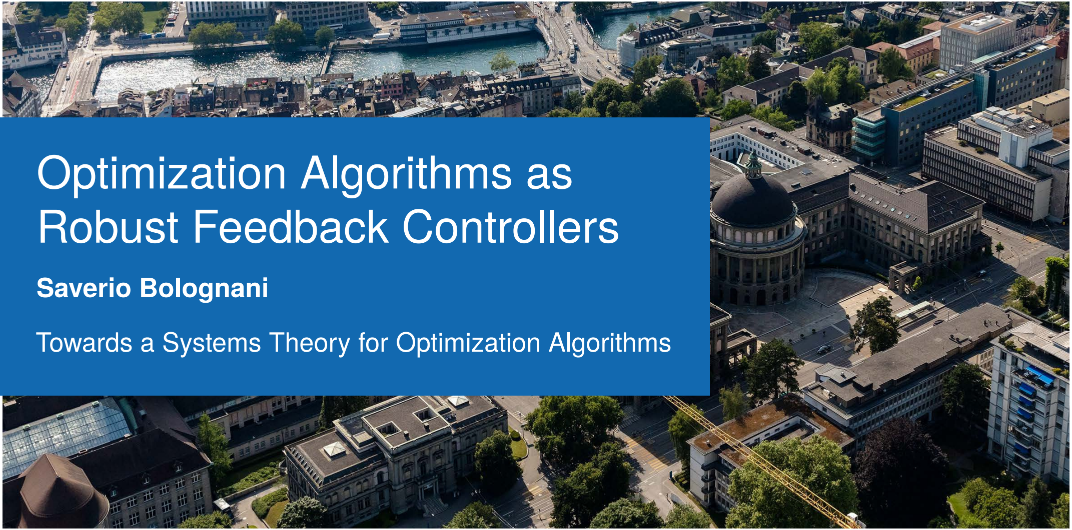
ETH zürich

AUTOMATIC CONTROL LABORATORY



Optimization Algorithms as Robust Feedback Controllers

Saverio Bolognani

Towards a Systems Theory for Optimization Algorithms

Optimal steady-state regulation

- **Stable** or **pre-stabilized** nonlinear plant
- **Steady-state specifications** in terms of optimality (efficiency, economic cost, ...)
- Multiple **constrained inputs** to be set
- Multiple **steady-state constraints** to be satisfied
- **Exogenous factors** (disturbances) affect the plant steady state

Data networks

- Multiple source-destination pairs
- Network utility maxim.
- Coupling via congestion
- Throughput

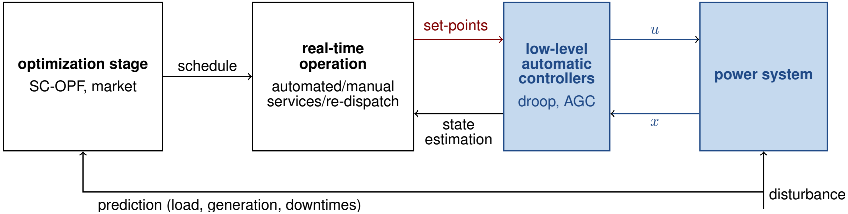
Grid frequency control

- Activation of reserves (generation ramp up/down)
- Exogenous time-varying demand
- Economic cost

Processes (compressors)

- Complex steady-state mapping
- Tunable parameters and set-points
- Efficiency and safety

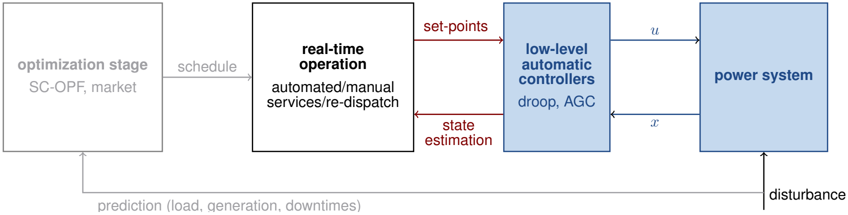
Real-time power system operation

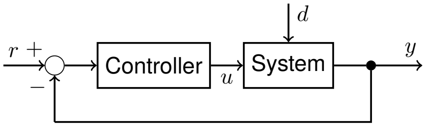


- **Power system dynamics** are considered at **steady state**
- **Disturbances** (renewable generation) require corrections from the predicted schedule
- **Operational limits** on all electrical quantities
- Economic interest defines the **optimal operation** (best generation set-points)

Desired operation: solution of a non-convex **Optimal Power Flow** problem

A control interpretation of real-time grid operation





Goal: The solution of the Optimal Power Flow problem is an asymptotically stable equilibrium for the closed-loop dynamics

→ **Feedback optimization**

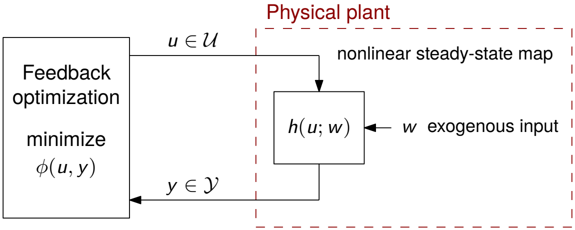
Outline

1. Feedback optimization design
2. Closed-loop analysis
3. Conclusions

Outline

1. Feedback optimization design
2. Closed-loop analysis
3. Conclusions

“Certainty-equivalence” design



Steady state specs

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

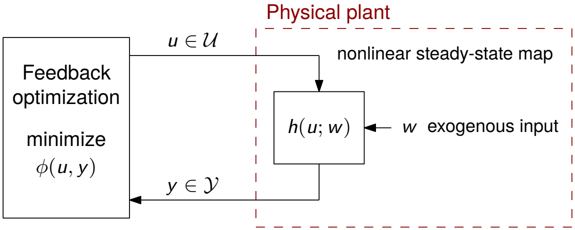
How to design **feedback controllers** that yield the desired **(autonomous) closed-loop behavior**?

Algebraic **input/output steady-state map** $y = h(u; w)$

+

Classical **iterative nonlinear optimization algorithms**

“Certainty-equivalence” design



Steady state specs

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

Optimization perspective

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

Control perspective

Feedback controllers 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, extremum seeking, real-time iteration, modifier adaptation...

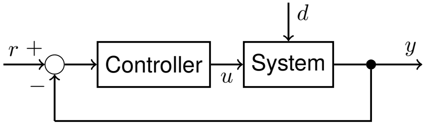
Design of feedback optimizers

Borrow ideas from **iterative optimization algorithms** for **nonlinear 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.

Re-deriving PI controllers



$$\begin{aligned} \text{minimize}_{u,y} \quad & u^2 \\ \text{subject to} \quad & y = r \quad \text{(tracking)} \\ & y = h(u;w) \quad \text{(plant steady-state)} \end{aligned}$$

Dual ascent iteration

$$\mathcal{L}(x, \lambda) = u^2 + \lambda(h(u;w) - r)$$

- Primal minimization $u = -\frac{1}{2}\nabla h(u;w)\lambda$
- Dual ascent $\dot{\lambda} = \alpha\frac{\partial \mathcal{L}}{\partial \lambda} = \alpha(h(u;w) - r)$

I controller

$$\dot{u} = \frac{\alpha}{2}\nabla h(u;w)(r-y)$$

Augmented dual ascent iteration

$$\mathcal{L}(x, \lambda) = u^2 + \lambda(h(u;w) - r) + \rho(h(u;w) - r)^2$$

PI controller

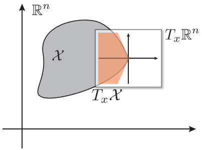
$$\dot{\lambda} = \alpha(r-y)$$

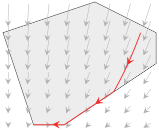
$$u = \frac{1}{2}\nabla h(u;w)\lambda + \rho\nabla h(u;w)(r-y)$$

Examples of feedback optimization design

$$\begin{array}{rll} \text{minimize}_{u,y} & \phi(u,y) & \\ \text{subject to} & y \in \mathcal{Y} & \text{output constraints} \\ & u \in \mathcal{U} & \text{input saturation} \\ & y = h(u;w) & \text{steady state map} \end{array}$$

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





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

Gradient descent flow $\to$ proportional-like feedback law

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

$\to$ **arbitrarily small output constraint violation**

Examples of feedback optimization design

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

Output constraint representation

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

Lagrangian

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

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

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

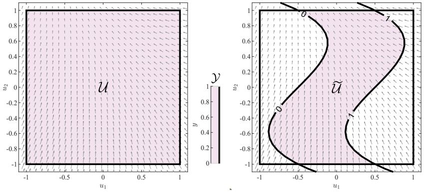
$$
\begin{cases}
\dot{u} = \Pi_{\mathcal{U}} \left[-\nabla_u \phi(u, y) - \underbrace{\nabla h(u; w)'}_{\text{model}} \nabla_y \phi(u, y) - \underbrace{\nabla h(u; w)'}_{\text{model}} \nabla g(y)' \lambda \right] \\
\dot{\lambda} = \Pi_{\geq 0} \left[g(y) \right]
\end{cases}
$$

$\rightarrow$ **asymptotic (exact) constraint satisfaction**

Examples of feedback optimization design

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

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



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

Projection on the input and output constraints

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

→ **any-time constraint satisfaction**

→ **discretization: repeated QP (projection)**
Häberle et al., IEEE Control Systems Letters, 2021

Projected gradient flow via repeated Quadratic Programming

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

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

Discrete-time approximation

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

$$\begin{aligned} \delta u := \quad & \arg\min_v && \| v - (-\nabla_u \phi(u,y) - \nabla h(u;w)' \nabla_y \phi(u,y)) \|^2 \\ & \text{subject to} && A(u + \alpha v) \leq b \\ &&& \underbrace{C(y + \alpha \nabla h(u;w)' v) \leq d}_{\text{1st order approx of } h^{-1}(\mathcal{Y}) \text{ centered at the measurement } y}, \end{aligned}$$

Theorem: (Häberle et al., IEEE Control Systems Letters, 2021)

LICQ + Lipschitz + differentiability + small α $\rightarrow$ global convergence to the set of local minima

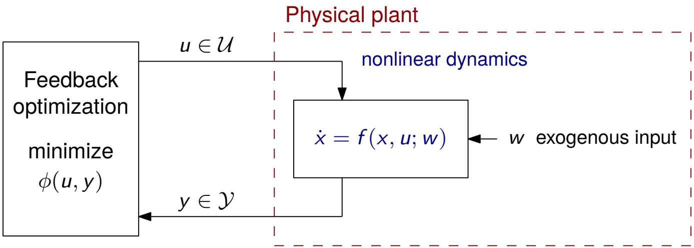
Outline

1. Feedback optimization design

2. Closed-loop analysis

3. Conclusions

Closing the loop



Optimization Dynamics

Example: Generalized gradient descent

$$\dot{u} = -Q(u)\left(-\nabla_u \phi(u, x) - \nabla h(u; w)' \nabla_y \phi(u, x)\right)$$

with $Q(u) \succ 0$

Plant Dynamics

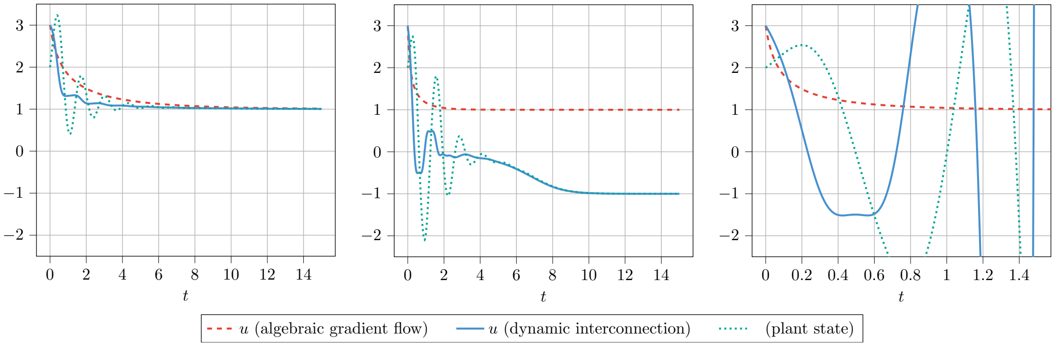
Exponentially stable system

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

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

Closed loop

» Increasing gain $\|Q(u)\|$ »



Singular perturbation analysis

Theorem A. Hauswirth, S. Bolognani, G. Hug, F. Dörfler, IEEE TAC, 2021

Assume

- Physical system **exponentially stable** with Lyapunov function $W(u,x)$ s.t.

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

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

Then, all trajectories converge to the set of KKT points whenever

$$\sup_u \|Q(u)\| < \frac{\gamma}{\zeta L} \,.$$

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

Gradient-based Feedback Optimization

Projected Gradient Descent

Choose $Q(u) = \varepsilon I$.
Stability is guaranteed if

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

$\rightarrow$ **prescription on global control gain**

Saddle-flow

Similar prescription on ε, but

$\rightarrow$ requires **exponential stability** of the saddle flow

[Qu & Li, 2018]

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

And

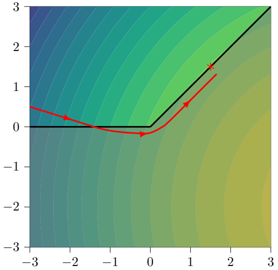
- Higher-order optimization dynamics

But not

- Subgradient methods
- Accelerated gradient method

Design guideline and tradeoffs

	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



Outline

1. Feedback optimization design

2. Closed-loop analysis

3. Conclusions

Conclusions

Take home

- **Feedback optimization:** interconnection of optimization algorithms and physical systems
- Robust regulation of stable plants to **optimal steady states**
- **Flexible design:** tap into nonlinear optimization algorithms
- **Guidelines for stability:** algorithm selection and tuning

Next step: Generalization to **feedback equilibrium seeking**: optima & Nash equilibria

G. Belgioioso, D. Liao-McPherson, M. Hudoba de Badyn, S. Bolognani, J. Lygeros, and F. Dörfler.
Sampled-data online feedback equilibrium seeking: Stability and tracking.
https://arxiv.org/abs/2103.13988

IEEE CDC 2021 **TuB15 Tuesday December 14, 2021** **16:15**

ETH zürich




Adrian Hauswirth, Saverio Bolognani, Gabriela Hug, and Florian Dörfler

Optimization Algorithms as Robust Feedback Controllers

arXiv:2103.11329 [math.OC], 2021.

ETH zürich

Project UNICORN
http://unicorn.control.ee.ethz.ch

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Federal Office of Energy SFOE

Relevant literature

	convex problem	input constraints	output constraints	continuous- or discrete-time	plant model
Gan & Low (2016)	no	yes	none/soft	DT	nonlin. algebraic
Hauswirth et al. (2016)	no	yes	none/soft	CT (non-smooth)	nonlin. algebraic
Tang et al. (2017)	no	yes	none/soft	DT	nonlin. algebraic
Hauswirth et al. (2017)	no	yes	none/soft	DT	nonlin. algebraic
Mazzi et al. (2018)	no	yes	unilateral	DT	nonlin. algebraic
Dall'Anese & Simonetto (2018)	no	yes	unilateral	DT	nonlin. algebraic
Hauswirth et al. (2018)	no	yes	none/soft	CT (non-smooth)	nonlin. algebraic
Tang et al. (2018b)	no	yes	unilateral	DT	nonlin. algebraic
Tang et al. (2018a)	no	yes	unilateral	DT/CT (non-smooth)	nonlin. algebraic
Nelson & Mallada (2018)	yes	no	none/soft	CT (smooth)	LTI
Lawrence et al. (2018)	yes	no	lin. equality	CT (smooth)	LTI
Menta et al. (2018)	yes	no	lin. equality	CT (smooth)	LTI
Zhang et al. (2018)	yes	no	unilateral	CT (non-smooth)	LTI
Colombino et al. (2019b)	no	yes	none/soft	DT	nonlin. algebraic
Bernstein et al. (2019)	yes	yes	unilateral	DT	nonlin. algebraic
Chang et al. (2019)	yes	yes	unilateral	CT (non-smooth)	lin. algebraic
Colombino et al. (2019a)	yes	no	linear eq.	CT (smooth)	LTI
Lawrence et al. (2020)	yes	no	linear eq.	CT (smooth)	LTI
Picallo et al. (2020)	no	yes	none/soft	DT	nonlin. algebraic
Hauswirth et al. (2020e)	yes	yes	none/soft	CT (non-smooth)	nonlin. dynamic
Hauswirth et al. (2020c)	no	yes	unilateral	CT (non-smooth)	nonlin. dynamic
Hauswirth et al. (2020b)	no	yes	unilateral	CT (non-smooth)	nonlin. dynamic
Simpson-Porco (2020)	yes	no	lin. equality	CT (smooth)	nonlin. dynamic