# Embedded Learning and Optimization Methods for Nonlinear Systems

Moritz Diehl
Systems Control and Optimization Laboratory
Department of Microsystems Engineering and Department of Mathematics
University of Freiburg

joint work with Florian Messerer, Katrin Baumgärtner, and Adrian Bürger

KU Leuven, ELO-X Spring School, March 24, 2022

# Overview

- Model Predictive Control - Examples and Optimization Problems

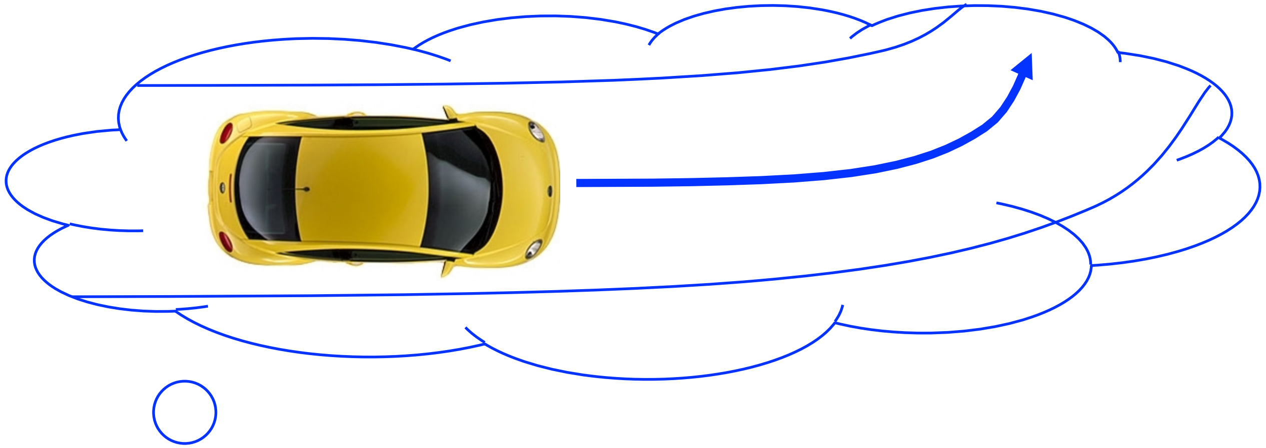
- Convexity Exploiting Newton-Type Optimization
  - Sequential Convex Programming (SCP)
  - Generalized Gauss-Newton (GGN)
  - + SLP, CGN, SCQP, SQCQP

- Zero-Order Optimization-based Iterative Learning Control
  - Tutorial Example
  - Bounding the Loss of Optimality and Exactness

- Mixed Integer Optimal Control
  - Problem Statement
  - Three Step Algorithm
  - Application to Renewable Energy System in Karlsruhe

# Model Predictive Control (MPC)

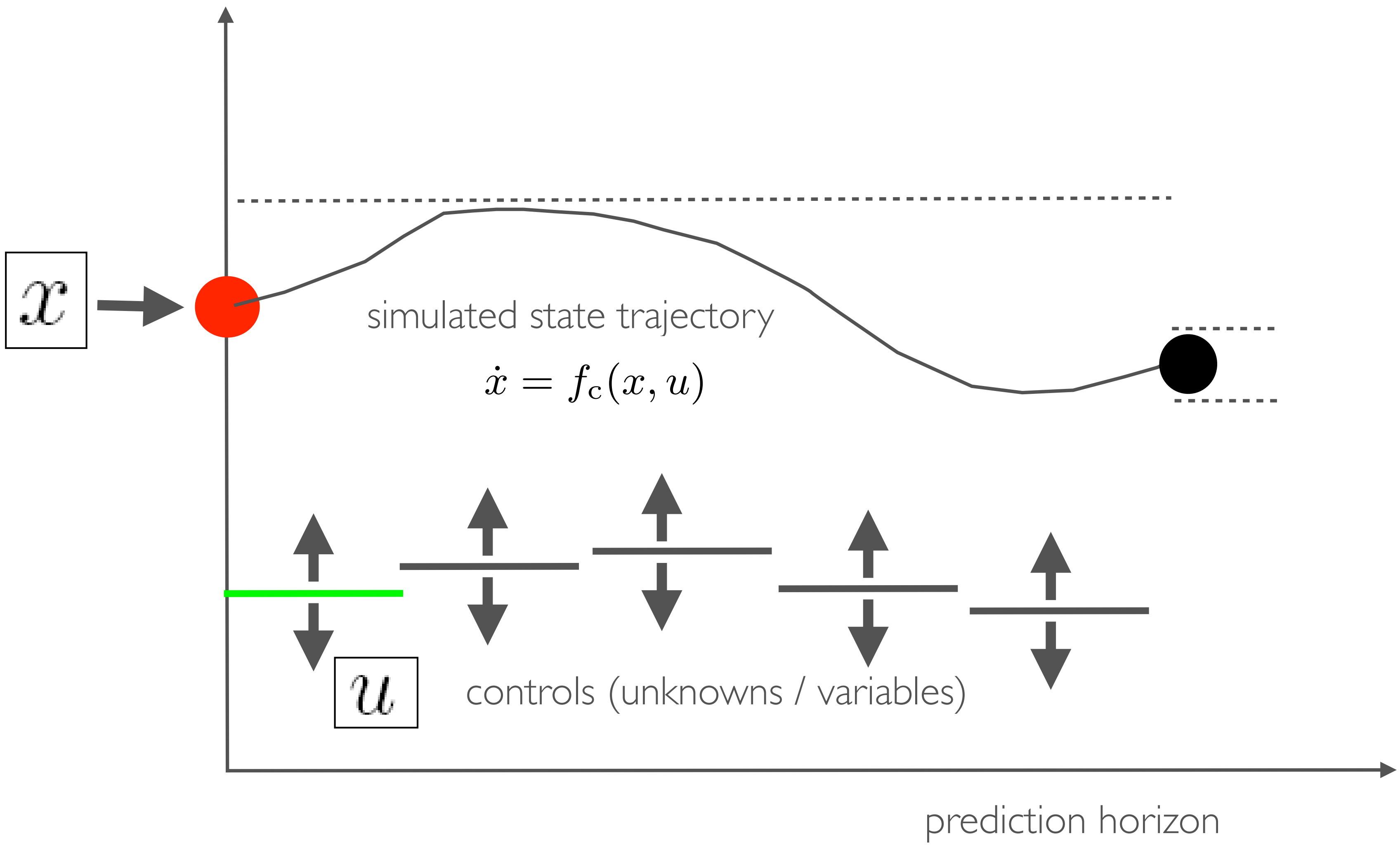
Always look a bit into the future



Example: driver predicts and optimizes, and therefore slows down before a curve

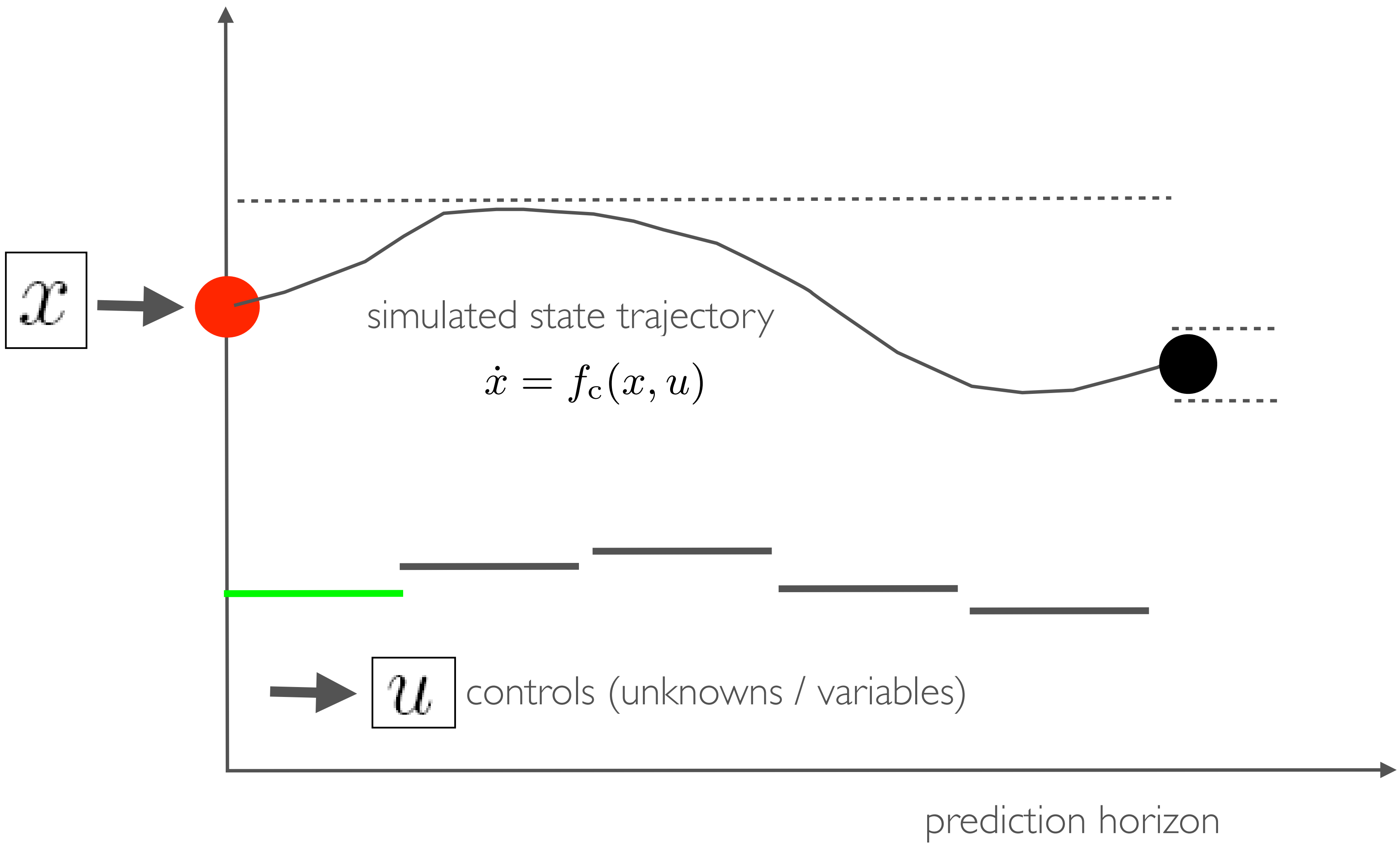
# Optimal Control Problem in MPC

For given system state $\boldsymbol{x}$, which controls $\boldsymbol{u}$ lead to the best objective value without violation of constraints ?

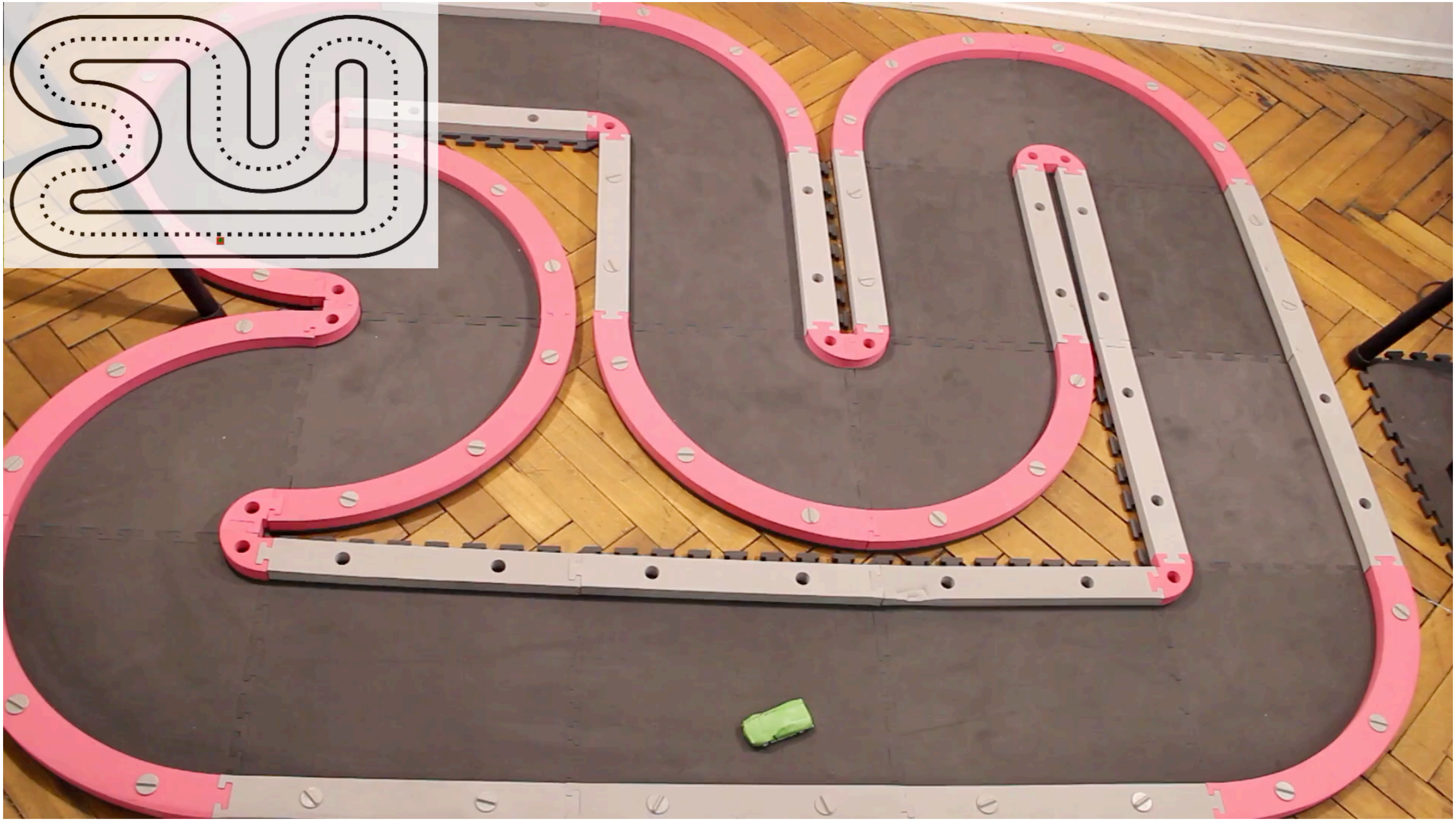


# Optimal Control Problem in MPC

For given system state $\boldsymbol{x}$, which controls $\boldsymbol{u}$ lead to the best objective value without violation of constraints ?



# Model Predictive Control of RC Race Cars (in Freiburg)



Minimize least squares distance to centerline, respect constraints. Use nonlinear embedded optimization software *acados* coupled to ROS, sample at 100 Hz.

[Kloeser et al. 2020]

# MPC needs numerical simulation

## e.g. Runge Kutta (RK) methods (for one sampling interval with piecewise constant control)

$$\boxed{\dot{x} = f_{\mathrm{c}}(x, u)}$$

**Exact ODE solution**

$$\begin{aligned} x(0) &= s, \\ \dot{x}(t) &= v(t) \\ v(t) &= f_{\mathrm{c}}(x(t), u), \\ &\quad \text{for} \quad t \in [0, \Delta t] \\ S(s, u) &:= x(\Delta t) \end{aligned}$$

**$N$ steps of general RK method with $S$ stages**

$$\begin{aligned} x_0 = s, \quad x_{k+1} &= x_k + h \textstyle\sum_{j=1}^{S} b_j v_{k,j} \\ x_{k,i} &= x_k + h \textstyle\sum_{j=1}^{S} a_{ij} v_{k,j} \\ v_{k,i} &= f_{\mathrm{c}}(x_{k,i}, u), \\ &\quad \text{for} \quad i = 1, \ldots, S, \quad k = 0, \ldots, N-1 \\ S(s, u) &:= x_N \end{aligned}$$

- $a_{ij}$ and $b_j$ are **Butcher tableau entries** of (potentially implicit) Runge Kutta method
- step length $h := \Delta t / N$; intermediate states $x_k, x_{k,i}, v_{k,i} \in \mathbb{R}^{n_s}$ with integration step index $k \in \{0, 1, \ldots, N\}$ and RK stage index $i, j \in \{1, \ldots, S\}$
- $N$ nonlinear equation systems with each $2Sn_s$ equations in $2Sn_s$ unknowns $(x_{k,i}, v_{k,i})$
- solved by Newton's method (or imposed as equality constraints in optimization)

# MPC needs numerical optimization
## e.g., using Nonlinear Programming (NLP)

**Continuous Time NMPC Problem**

$$\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & \int_0^T L(x,u)\mathrm{d}t + E(x(T)) \\
\text{s.t.} \quad & x(0) = \bar{x}_0 \\
& \dot{x}(t) = f(x(t),u(t)) \\
& 0 \geq h(x(t),u(t)),\ t \in [0,T] \\
& 0 \geq r(x(T))
\end{aligned}$$

Assume smooth convex $L, E, h, r$.
Nonlinear $f$ makes problem nonconvex.
Direct methods discretize, then optimize.
E.g. collocation or multiple shooting.

**Discretized NMPC Problem (an NLP)**

$$\begin{aligned}
\min_{x,z,u} \quad & \textstyle\sum_{k=0}^{N-1} \Phi_L(x_k,z_k,u_k) + E(x_N) \\
\text{s.t.} \quad & x_0 = \bar{x}_0 \\
& x_{k+1} = \Phi_f^{\mathrm{dif}}(x_k,z_k,u_k) \\
& 0 = \Phi_f^{\mathrm{alg}}(x_k,z_k,u_k) \\
& 0 \geq \Phi_h(x_k,z_k,u_k),\ k = 0,\ldots,N-1 \\
& 0 \geq r(x_N)
\end{aligned}$$

Again, smooth convex $\Phi_L, E, \Phi_h, r$.
Variables $x = (x_0, \ldots)$ and $z = (z_0, \ldots)$ and $u = (u_0, \ldots, u_{N-1})$ can be summarized in vector $w \in \mathbb{R}^{n_w}$.

# MPC Example: Point-To-Point Motions [PhD Vandenbrouck 2012]


Fast oscillating systems (cranes, plotters, wafer steppers, …)

Control aims:

- reach end point as fast as possible
- do not violate constraints
- no residual vibrations

Idea: formulate as embedded optimization problem
in form of Model Predictive Control (MPC)

# Time Optimal MPC of a Crane

SENSORS

- line angle
- cart position

→

MPC

→

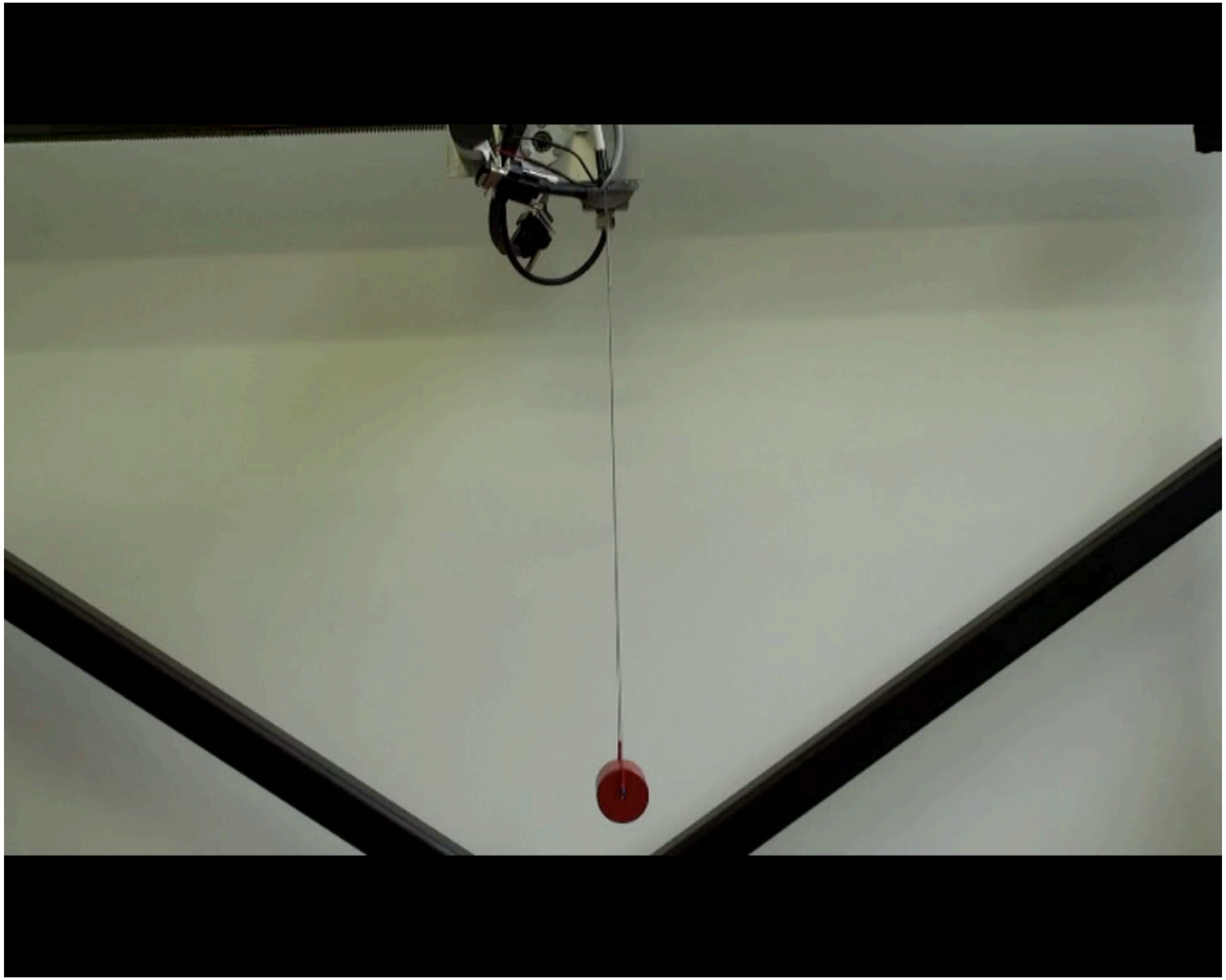
ACTUATOR

- cart motor

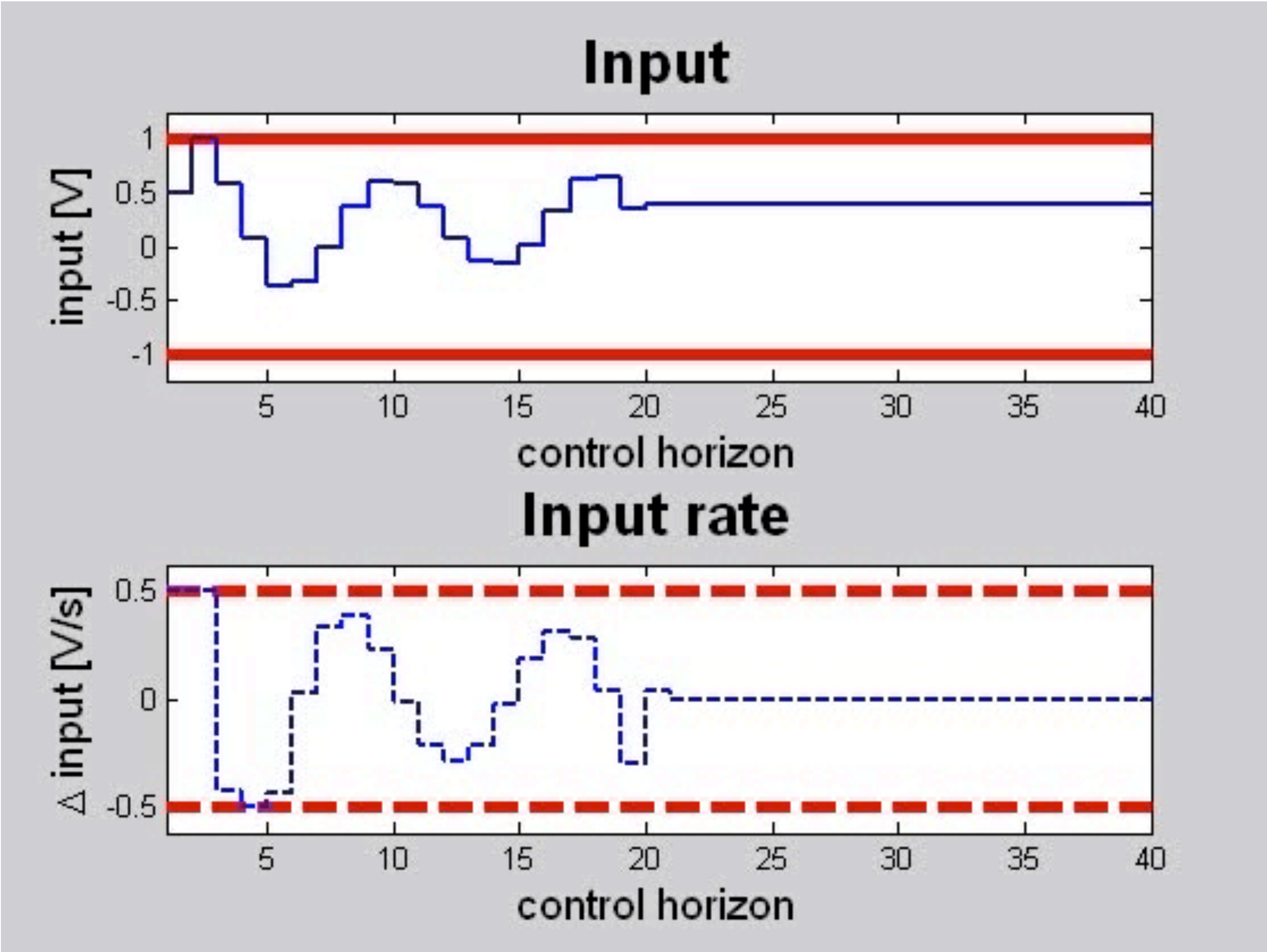
Hardware: xPC Target.    Software: qpOASES [Ferreau, D., Bock, 2008]

# Time Optimal MPC of a Crane

KU Leuven [Vandenbrouck, Swevers, D.]



# Optimal solutions varying in time (inequalities matter)




Solver qpOASES [PhD H.J. Ferreau, 2011], [Ferreau, Kirches, Potschka, Bock, D. , A parametric active-set algorithm for quadratic programming, Mathematical Programming Computation, 2014]

# eco4wind: MPC for wind turbine control

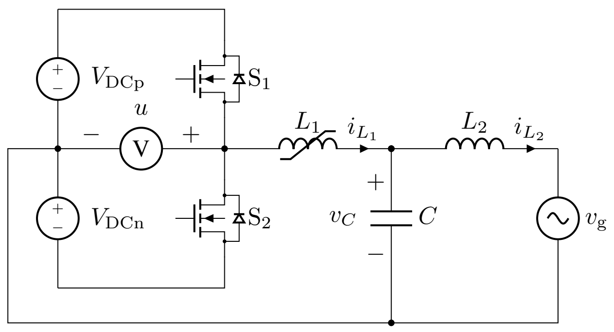

Industrial partners: IAV, SENVION (now bankrupt)

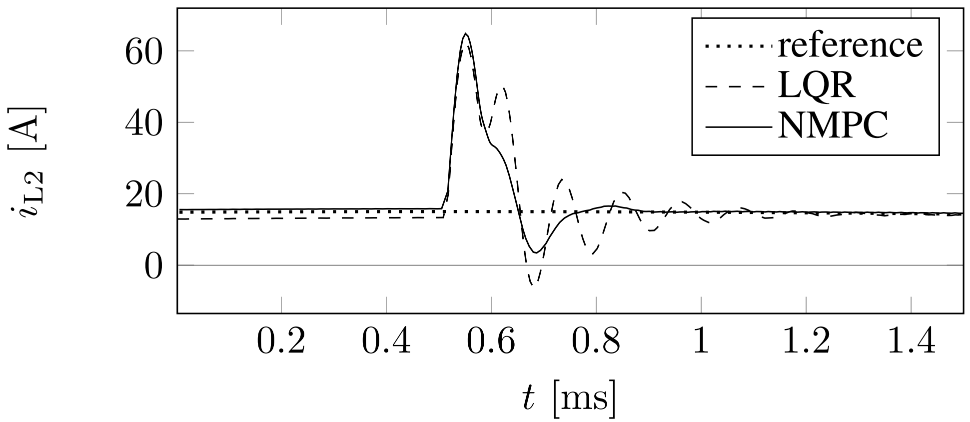
Aim: minimise fatigue and oscillations, respect constraints.

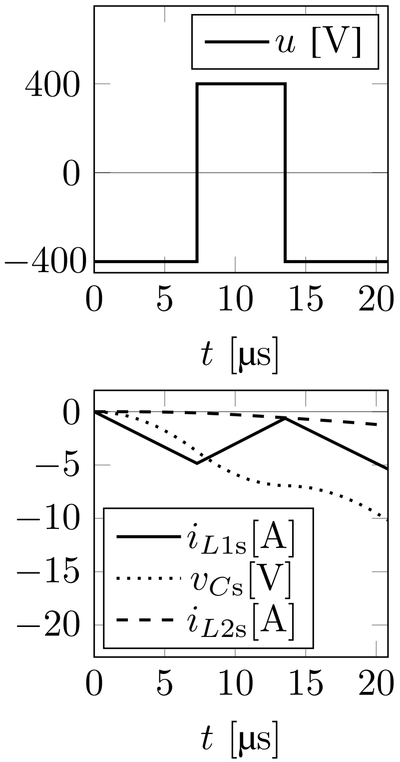
Nonlinear MPC with about 40 states based on ACADO/acados with QP solver HPIPM running on industrial hardware at IAV.

# Switched NMPC for Electric DC-AC Power Converter

## NMPC 48 kHz [Stickan, Frison et al. ACC 2022]



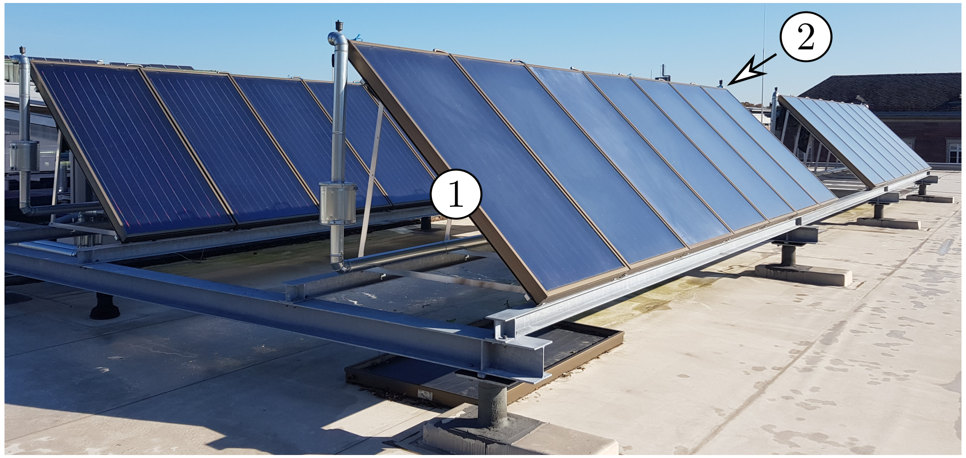



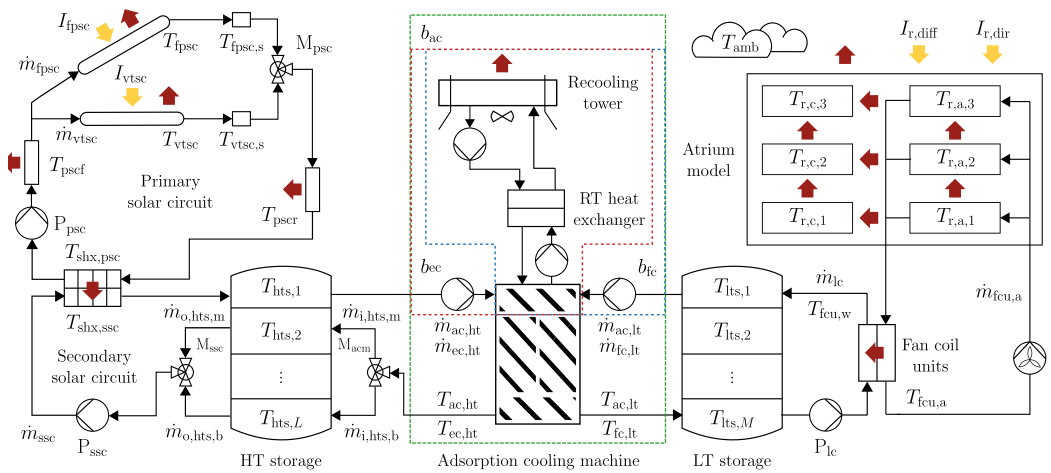



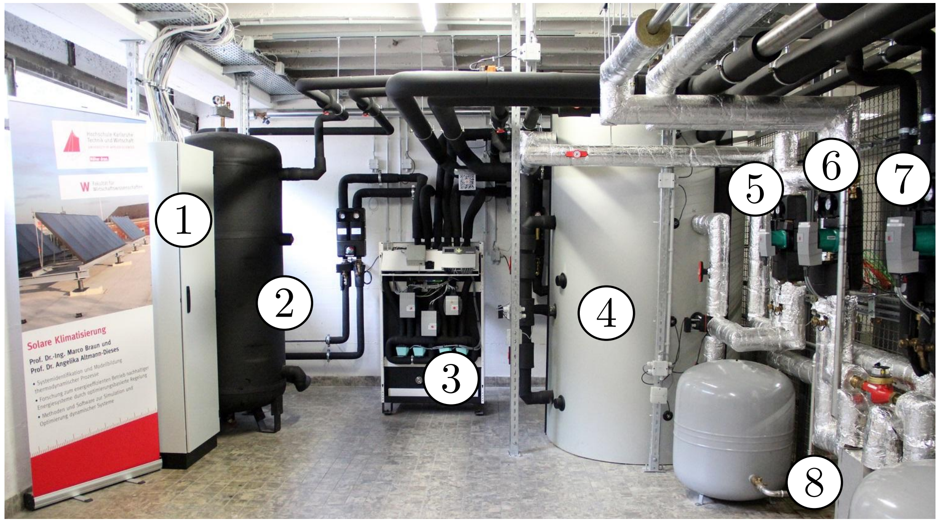

- NMPC aim: follow sinusoidal reference, react fast to grid failures
- 3 states, **1 binary input, 1 state dependent switch due to diodes (in blanking time)**
- sampling time: 25 microseconds, ARM A53@1.1GHz, horizon $N = 2$
- switching integrator, 3 RK4 and 4 Euler steps, generated as C code via CasADi
- hand tailored SQP real-time iteration, on track to be applied on industrial photovoltaic power converter (in DyConPV project).

# Nonlinear Mixed-Integer MPC of a Solar Adsorptive Cooling Machine

[Bürger et al., 2019]




Nonlinear ODE with 39 states, 6 continuous and 2 binary inputs. Contains combinatorial constraints such as minimum uptime, minimum downtime, …

Predict 24 hours. Aim: minimise electricity consumption.

# MPC needs System Identification and State Estimation

Prior to implementing an MPC controller, one needs to address two tasks:

- **System Identification (offline):**

  use a long sequence of recorded input and output data, $(a_0, \ldots, a_N)$ and $(y_0, \ldots, y_N)$, to identify parameters $p$ using e.g. least squares optimization or subspace identification

- **State Estimation (online):**

  estimate the state $s_k$ by using the previous control actions $(..., a_{k-2}, a_{k-1})$ and the past measurements $(..., y_{k-2}, y_{k-1})$ using e.g. Extended Kalman Filter (EKF) or moving horizon estimation (MHE) (MHE uses a fixed window of past data for fitting)

*Learning-based MPC* typically refers to an online model adaptation, i.e., to estimating parameters online (for which MHE is particularly suitable) ("learning a model" $=$ "system identification")

# Convex-over-linear Structure of MPC Optimization Problems

## (MPC itself, but also System Identification and Moving Horizon Estimation)

$$
\begin{aligned}
\underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad & \sum_{i=0}^{N} \varphi_i(F_i(s_i, u_i)) + \varphi_N(F_N(s_N)) \\
\text{subject to} \quad & s_0 = x, \\
& s_{i+1} = S_i(s_i, u_i), \quad i = 0, \dots, N-1, \\
& H_i(s_i, u_i) \in \Omega_i, \quad i = 0, \dots, N-1, \\
& H_N(s_N) \in \Omega_N
\end{aligned}
$$

- variables $w = (s, u)$ with $s = (s_0, \dots, s_N)$ and $u = (u_0, \dots, u_{N-1})$
- convexities in $\varphi_i$ (e.g. quadratic) and $\Omega_i$ (e.g. polyhedral, ellipsoidal)
- nonlinearities in dynamic system $S_i$ and constraint functions $F_i$, $H_i$
- often: $S_i$ result of time integration (direct multiple shooting)

# Overview

- Model Predictive Control - Examples and Optimization Problems
- Convexity Exploiting Newton-Type Optimization
  - Sequential Convex Programming (SCP)
  - Generalized Gauss-Newton (GGN)
  - + SLP, CGN, SCQP, SQCQP
- Zero-Order Optimization-based Iterative Learning Control
  - Tutorial Example
  - Bounding the Loss of Optimality and Exactness
- Mixed Integer Optimal Control
  - Problem Statement
  - Three Step Algorithm
  - Application to Renewable Energy System in Karlsruhe

# Nonlinear optimization with convex substructure

$$
\begin{aligned}
\underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad & \phi_0(F_0(w)) \\
\text{subject to} \quad & F_i(w) \in \Omega_i \quad i = 1, \dots, m, \\
& G(w) = 0
\end{aligned}
$$

Assumptions:

- twice continuously differentiable functions $G : \mathbb{R}^{n_w} \to \mathbb{R}^{n_g}$ and $F_i : \mathbb{R}^{n_w} \to \mathbb{R}^{n_{F_i}}$ for $i = 0, 1, \dots, m$.
- outer function $\phi_0 : \mathbb{R}^{n_{F_0}} \to \mathbb{R}$ convex.
- sets $\Omega_i \subset \mathbb{R}^{n_{F_i}}$ convex for $i = 1, \dots, m$,
  (possibly $z \in \Omega_i \Leftrightarrow \phi_i(z) \leq 0$ with smooth convex $\phi_i$)

Idea:
exploit convex substructure via *iterative convex approximations*.

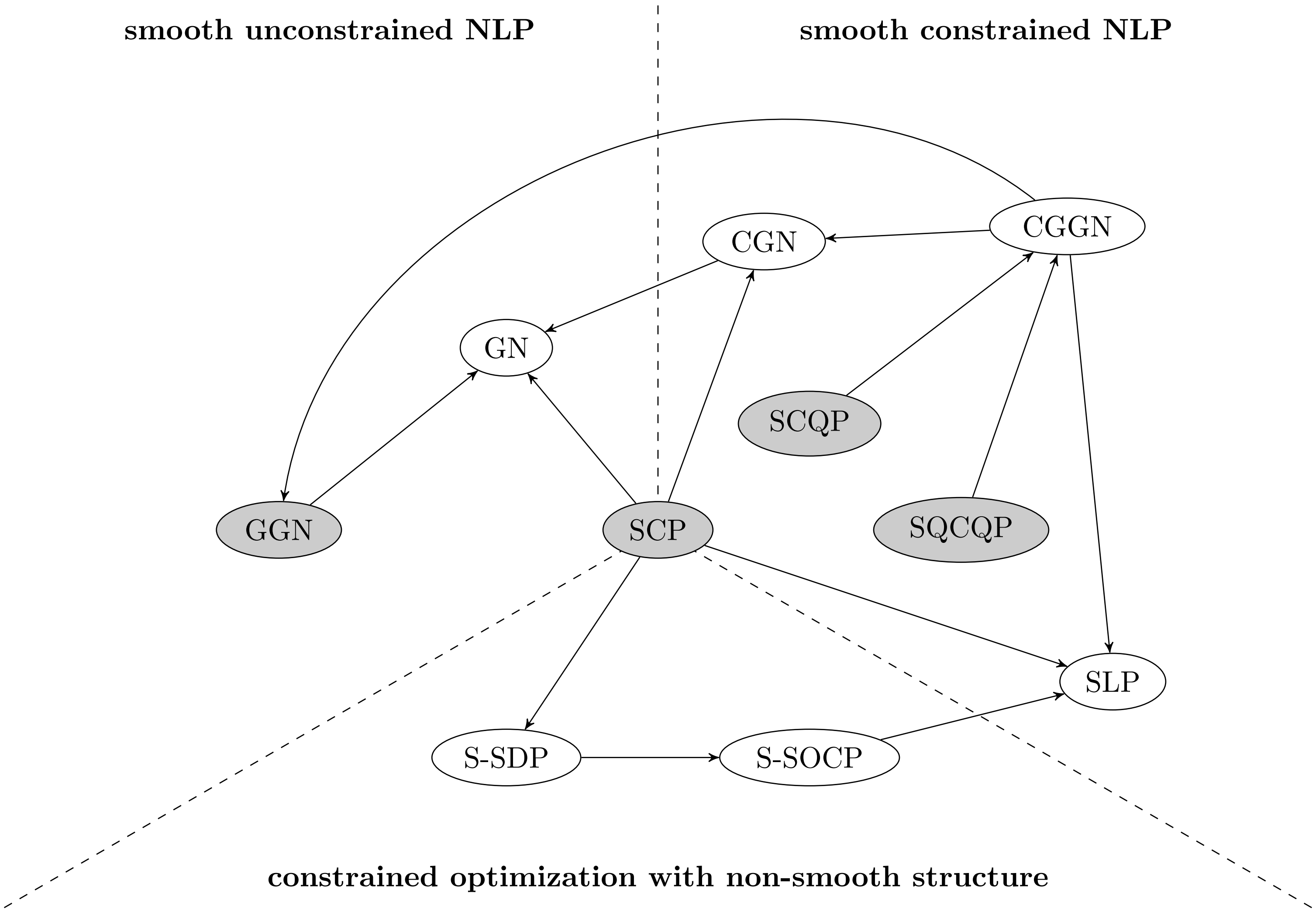
# Why is this class of problems and algorithms interesting ?

- many optimization problems have "convex-over-nonlinear" structure
- standard NLP solvers cannot address all non-smooth convex constraints
- there exist many mature and efficient convex optimization solvers

Some application areas:

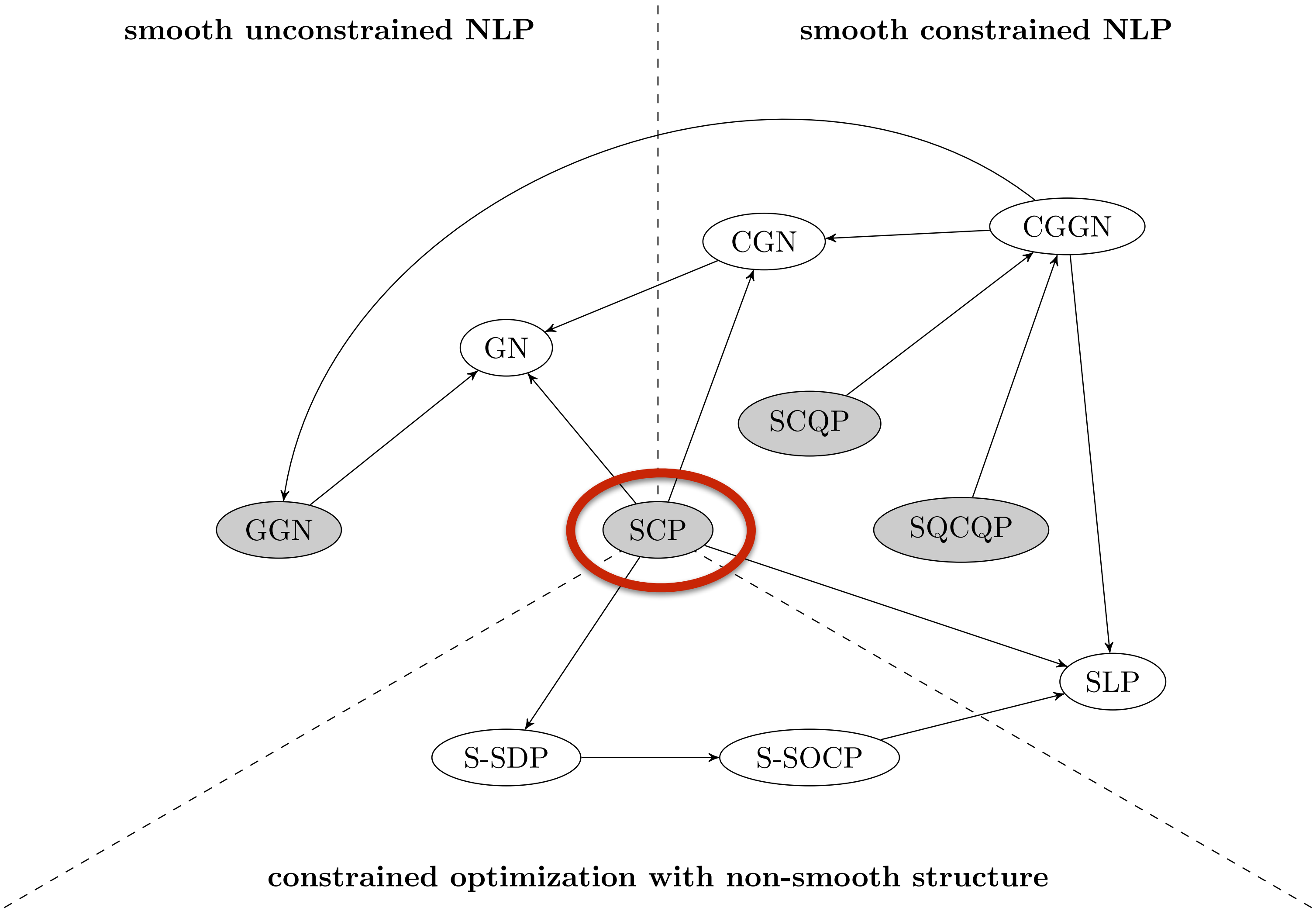
- nonlinear least squares for estimation and tracking
  [Gauss 1809; Bock 1983; Li and Biegler 1989; Sideris and Bobrow 2004]
- nonlinear matrix inequalities for reduced order controller design
  [Fares, Noll, Apkarian 2002; Tran-Dinh et al. 2012]
- ellipsoidal terminal regions in nonlinear model predictive control
  [Chen and Allgöwer 1998; Verschueren 2016]
- robustified inequalities in nonlinear optimization
  [Nagy and Braatz 2003; D., Bock, Kostina 2006]
- tube-following optimal control problems [Van Duijkeren 2019]
- non-smooth composite minimization [Apkarian et al. 2008; Lewis and Wright 2016]
- deep neural network training with convex loss functions
  [Schraudolph 2002; Martens 2016]

# Class Picture of Iterative Convex Approximation Methods



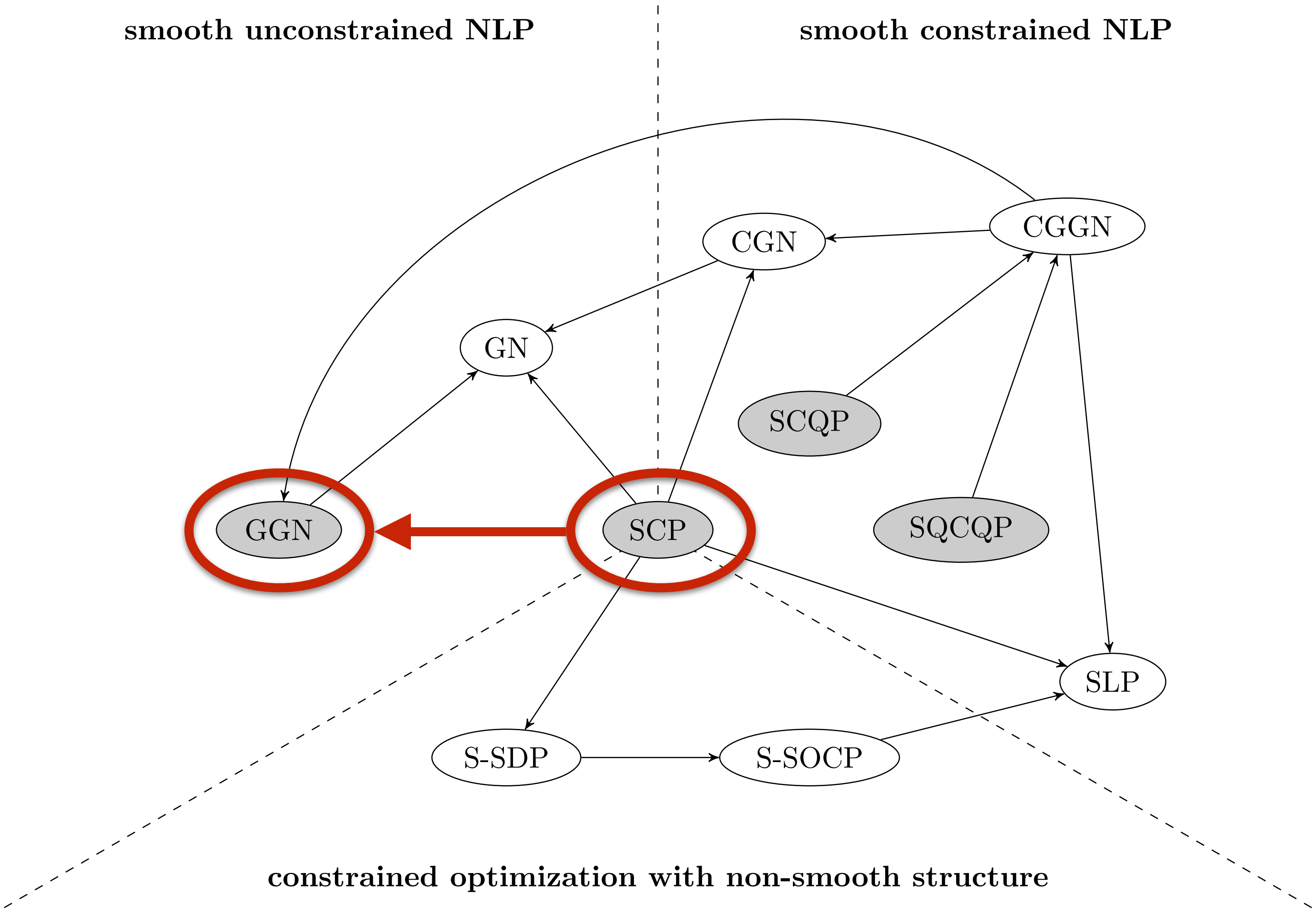

[Messerer et al., ESAIM 2021]

# Class Picture of Iterative Convex Approximation Methods



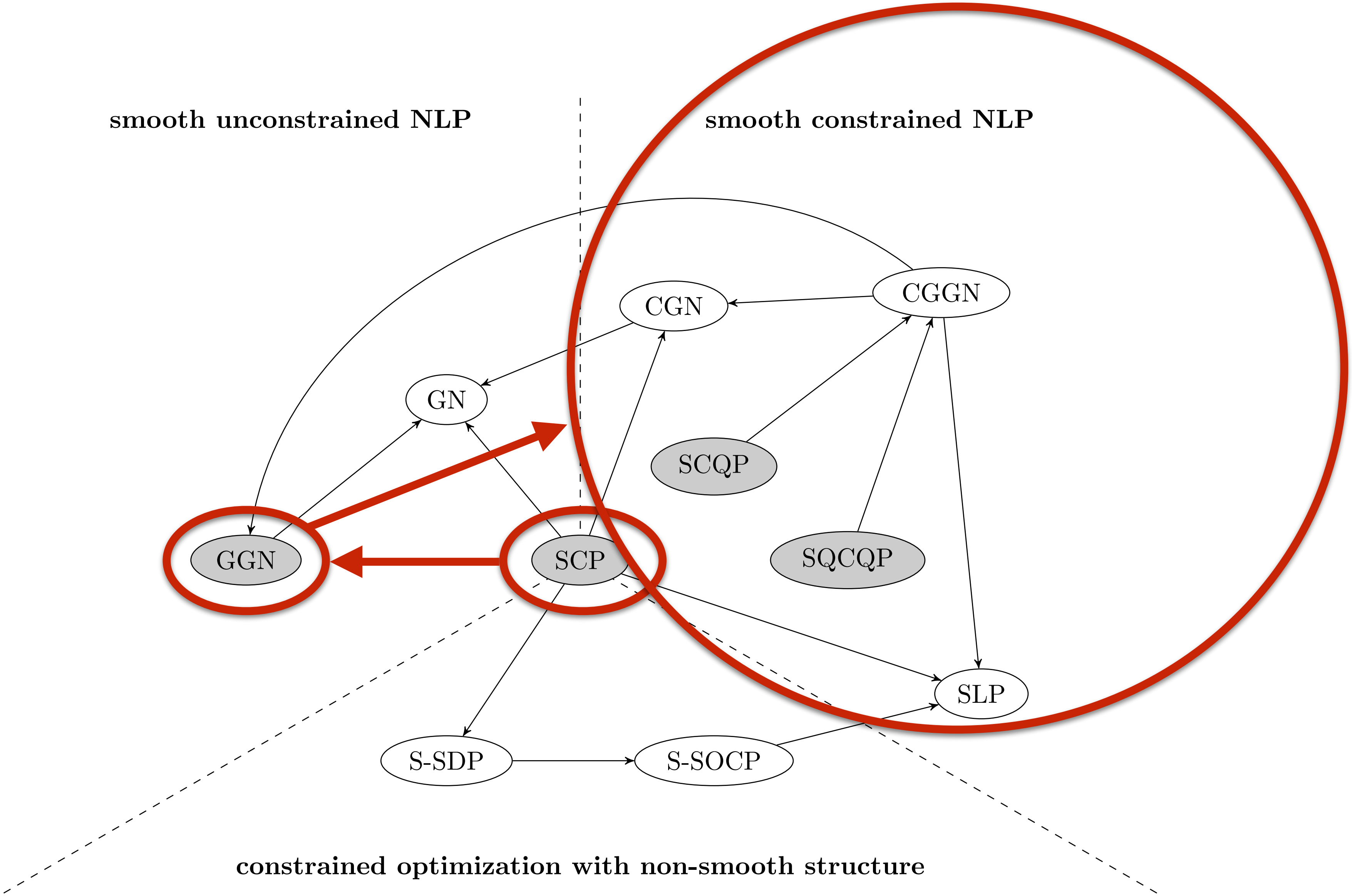

[Messerer et al., ESAIM 2021]

# Class Picture of Iterative Convex Approximation Methods



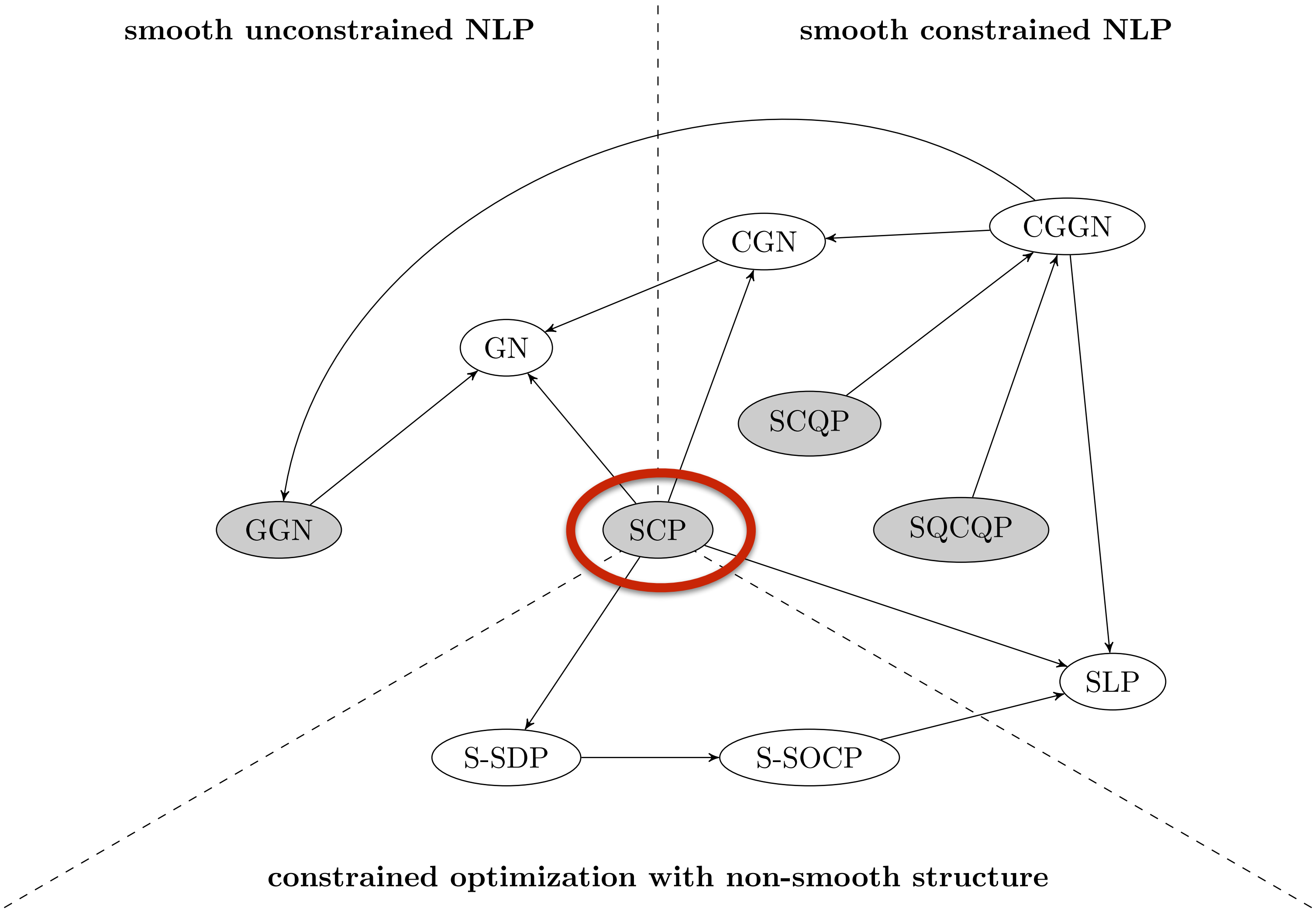

[Messerer et al., ESAIM 2021]

# Class Picture of Iterative Convex Approximation Methods




[Messerer et al., ESAIM 2021]

# Class Picture of Iterative Convex Approximation Methods




[Messerer et al., ESAIM 2021]

# Sequential Convex Programming (SCP)

- linearize $\boxed{F_i^{\mathrm{lin}}(w;\bar{w}) := F_i(\bar{w}) + J_i(\bar{w})\ (w - \bar{w})}$ with $J_i(\bar{w}) := \frac{\partial F_i}{\partial w}(\bar{w})$
- formulate convex subproblems:

$$
\begin{aligned}
\underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad & \phi_0(F_0^{\mathrm{lin}}(w;\bar{w})) \\
\text{subject to} \quad & F_i^{\mathrm{lin}}(w;\bar{w}) \in \Omega_i, \quad i = 1, \ldots, m, \\
& G^{\mathrm{lin}}(w;\bar{w}) = 0
\end{aligned}
$$

- start at $w_0$ with $k = 0$
- solve convex subproblem at $\bar{w} = w_k$ to obtain next iterate $w_{k+1}$

# Simplest case: smooth unconstrained problems

Unconstrained minimization of "convex over nonlinear" function

$$\underset{w \in \mathbb{R}^n}{\text{minimize}} \quad \underbrace{\phi(F(w))}_{=:f(w)}$$

Assumptions:
- Inner function $F : \mathbb{R}^n \to \mathbb{R}^N$ of class $C^2$
- Outer function $\phi : \mathbb{R}^N \to \mathbb{R}$ of class $C^2$ and convex

# Simplest case: smooth unconstrained problems

Unconstrained minimization of "convex over nonlinear" function

$$\underset{w \in \mathbb{R}^n}{\text{minimize}} \quad \underbrace{\phi(F(w))}_{=:f(w)}$$

Assumptions:
- Inner function $F : \mathbb{R}^n \to \mathbb{R}^N$ of class $C^2$
- Outer function $\phi : \mathbb{R}^N \to \mathbb{R}$ of class $C^2$ and convex

SCP subproblem becomes

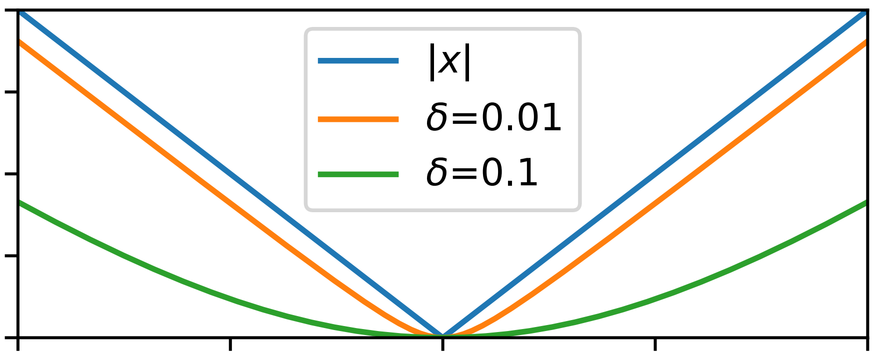
$$\underset{w \in \mathbb{R}^n}{\text{minimize}} \quad \underbrace{\phi\left(F^{\text{lin}}(w;\bar{w})\right)}_{=:f_{\text{SCP}}(w;\bar{w})} \tag{1}$$

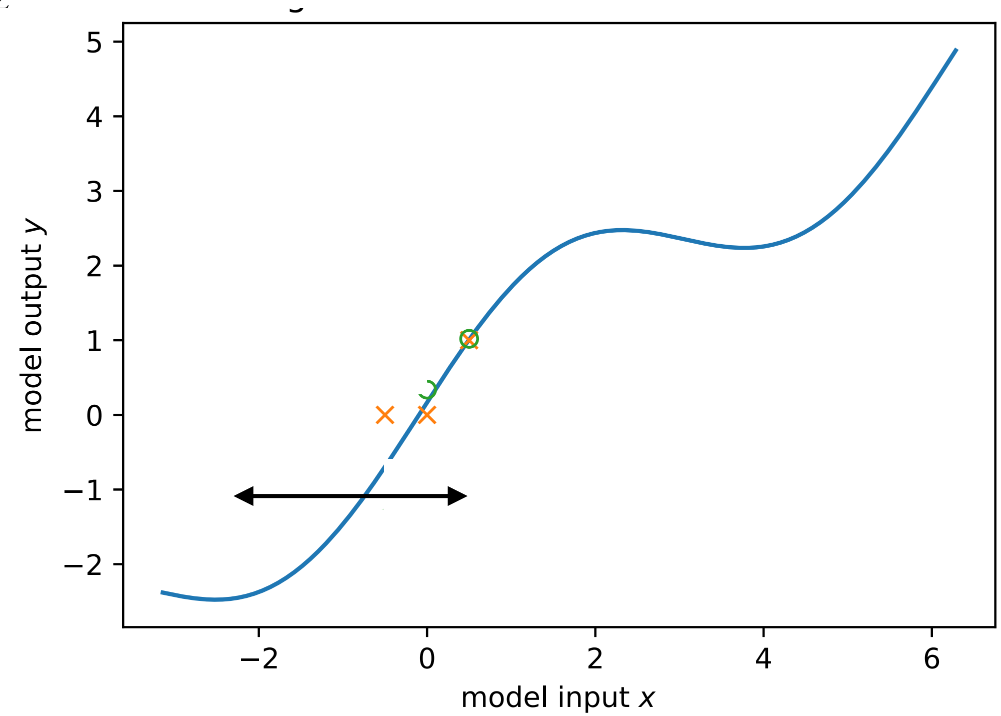
# Tutorial Example: Pseudo Huber Loss Minimization

Experiments conducted by Florian Messerer

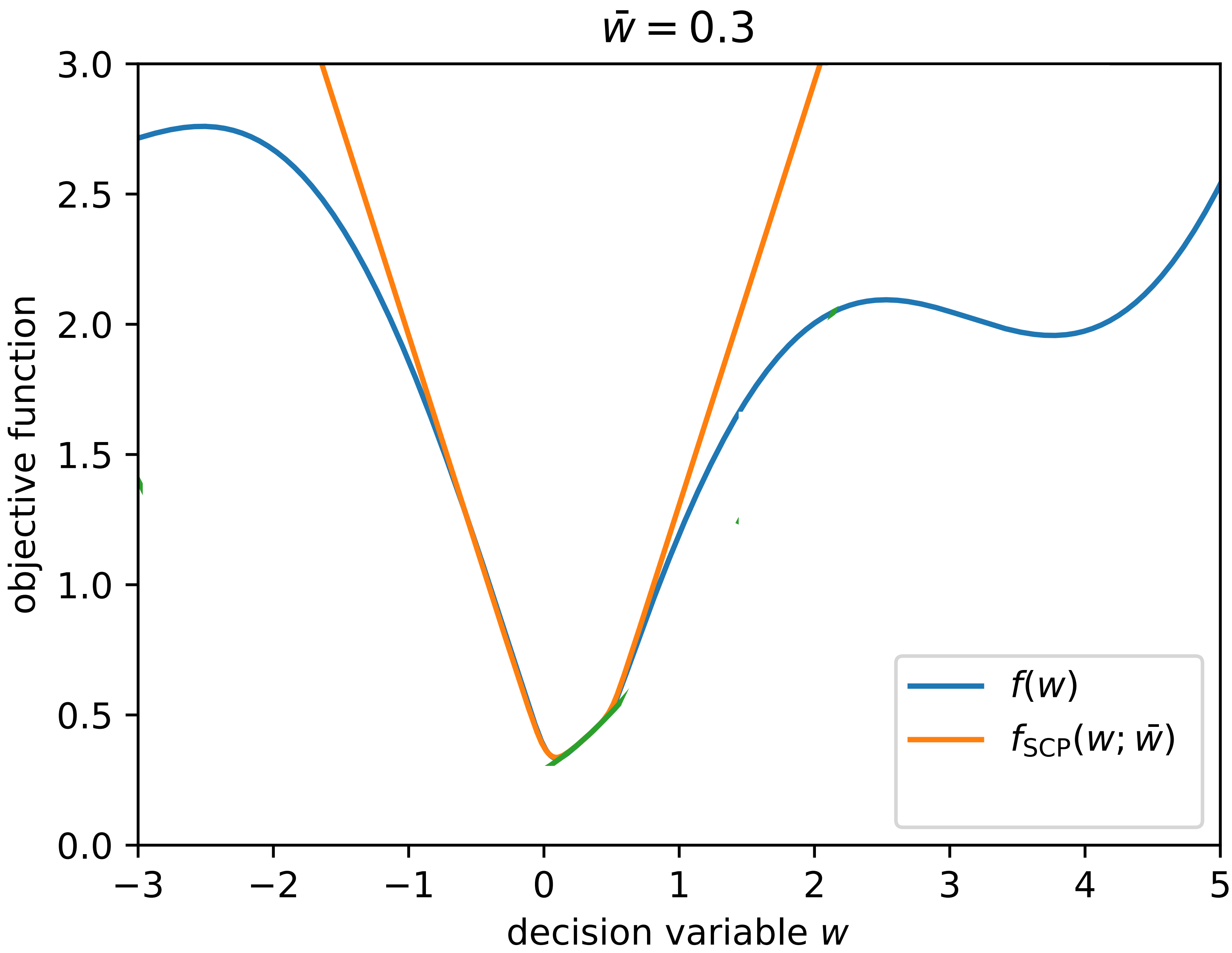
$$\underset{w \in \mathbb{R}}{\text{minimize}} \quad \underbrace{\sum_{i=1}^{n} \phi_\delta(y_i - m(x_i + w))}_{=:f(w)}$$

Aim: fit $n$=3 measurements $y_i$ to a model $m(w + x_i)$ with $m(x) = \frac{3}{4}x + \sin(x)$ using the pseudo Huber loss $\phi_\delta(x) := \sqrt{\delta^2 + x^2}$



# Cost function and SCP approximation



# SCP for Least Squares = Gauss-Newton

With quadratic $\phi(z) = \frac{1}{2}\|z\|_2^2 = \frac{1}{2}z^\top z$, SCP subproblems become

$$\underset{w \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|F(w_k) + J(w_k)(w - w_k)\|_2^2 \tag{2}$$

If $\operatorname{rank}(J) = n$ this is uniquely solvable, giving

$$w_{k+1} = w_k - \Big(\underbrace{J(w_k)^\top J(w_k)}_{=:B_{\mathrm{GN}}(w_k)}\Big)^{-1} \underbrace{J(w_k)^\top F(w_k)}_{=\nabla f(w_k)}$$

SCP applied to LS = Newton method with "Gauss-Newton Hessian"

$$B_{\mathrm{GN}}(w) \approx \nabla^2 f(w)$$

# Generalized Gauss-Newton (GGN) [Schraudolph 2002]

For general convex $\phi(\cdot)$ we have for $f(w) = \phi(F(w))$

$$\nabla^2 f(w) = \underbrace{J(w)^\top \, \nabla^2 \phi(F(w)) \, J(w)}_{=: B_{\mathrm{GGN}}(w)} + \underbrace{\textstyle\sum_{j=1}^{N} \nabla^2 F_j(w) \, \nabla_{z_j} \phi(F(w))}_{=: E_{\mathrm{GGN}}(w)}$$

"GGN Hessian" "Error matrix"

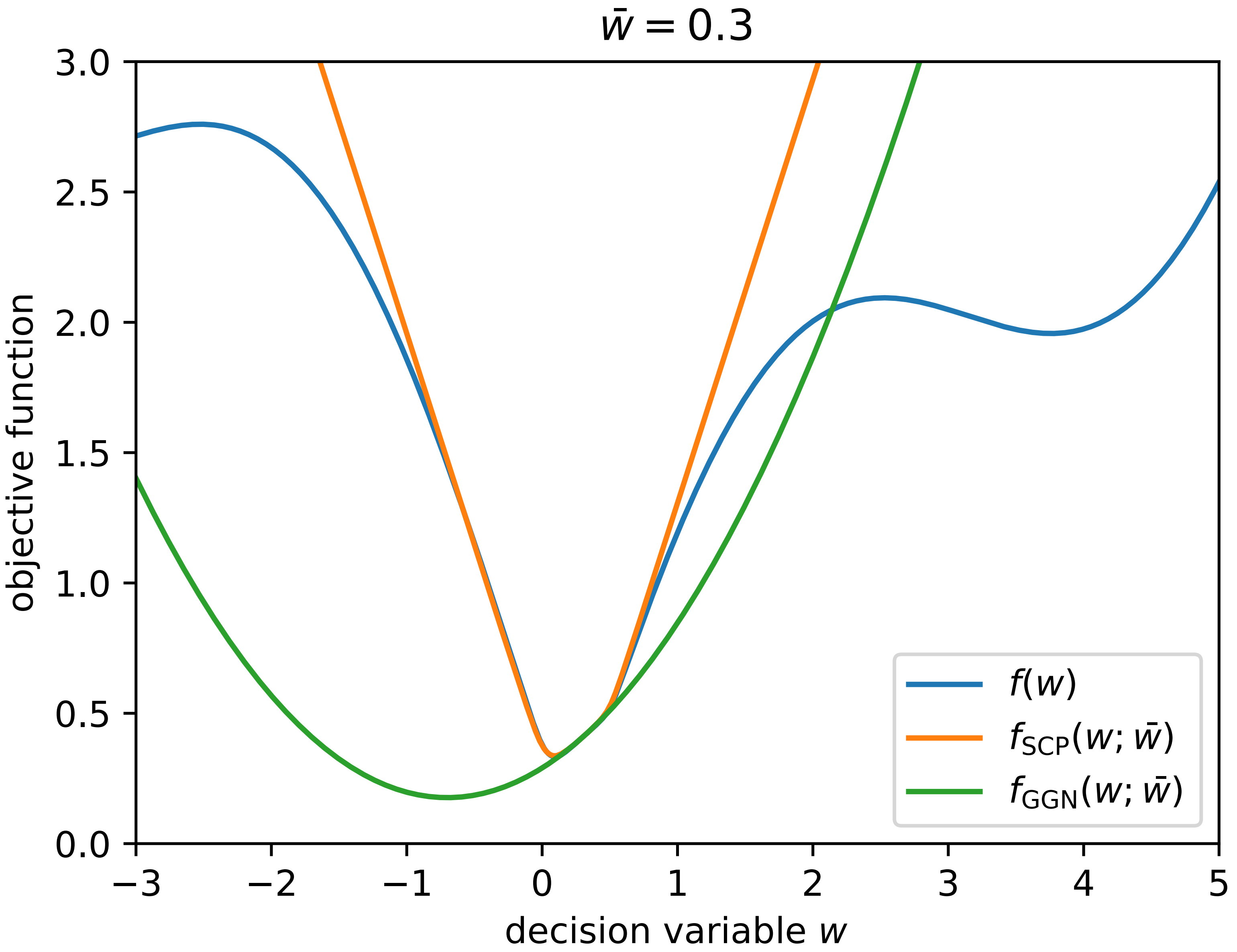
Generalized Gauss-Newton (GGN) method iterates according to

$$w_{k+1} = w_k - B_{\mathrm{GGN}}(w)^{-1} \nabla f(w_k)$$

Note: GGN solves convex quadratic subproblems

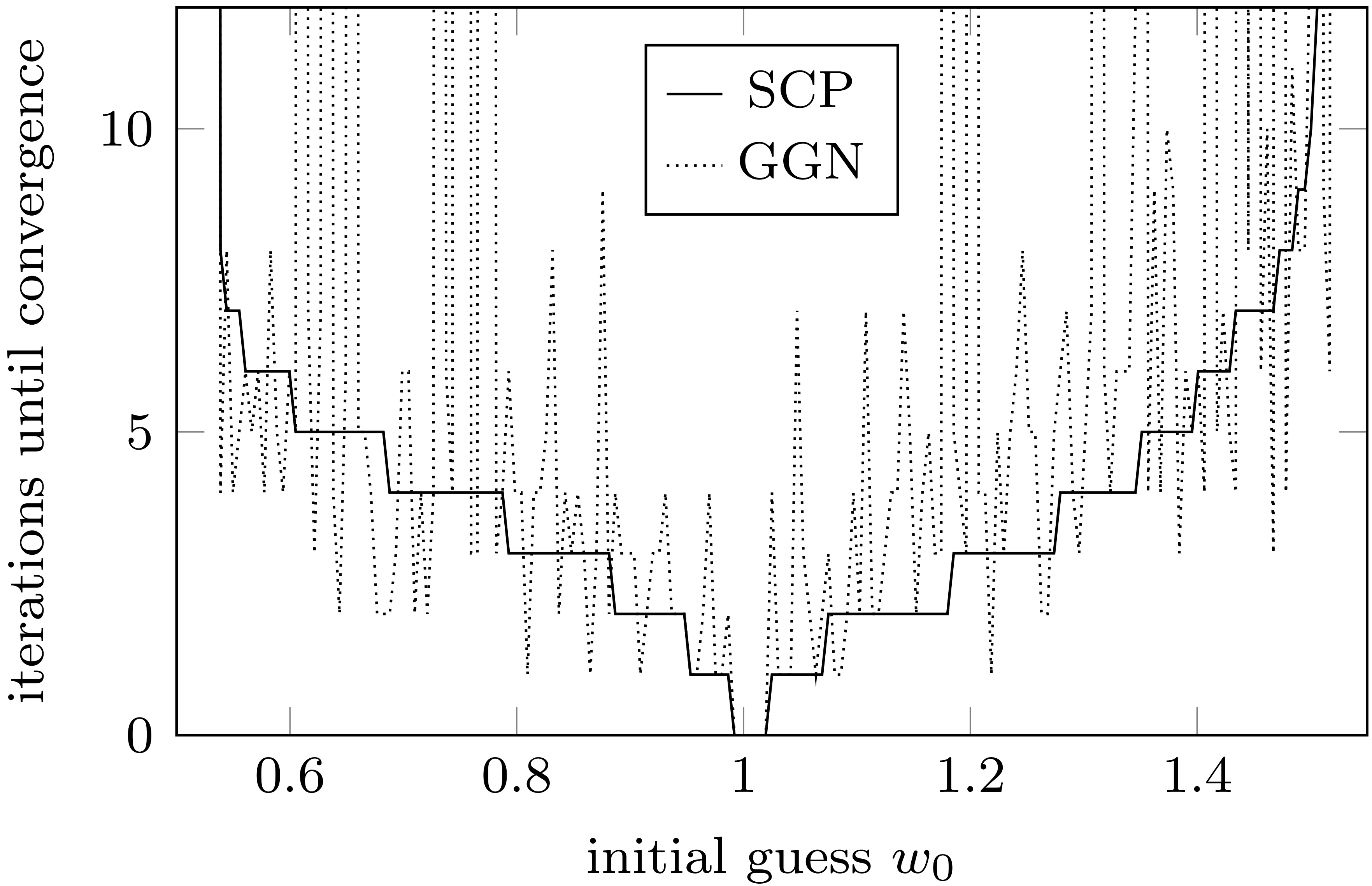
$$\min_{w \in \mathbb{R}^n} \; \underbrace{f(w_k) + \nabla f(w_k)^\top (w - w_k) + \frac{1}{2} (w - w_k)^\top B_{\mathrm{GGN}}(w_k)(w - w_k)}_{=: f_{\mathrm{GGN}}(w; w_k)}$$

# Tutorial Example: SCP and GGN Approximation



# Iteration count: SCP more predictable than GGN

(on a similar example)



# General smooth NLP formulation with constraints

Now regard an NLP with smooth convex $\phi_0, \phi_1, \ldots, \phi_m$

$$\begin{aligned} \underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad & \underbrace{\phi_0(F_0(w))}_{=:f_0(w)} \\ \text{subject to} \quad & \underbrace{\phi_i(F_i(w))}_{=:f_i(w)} \leq 0, \quad i = 1, \ldots, m, \\ & G(w) = 0 \end{aligned}$$

SCP subproblem becomes

$$\begin{aligned} \underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad & \phi_0(F_0^{\text{lin}}(w; \bar{w})) \\ \text{subject to} \quad & \phi_i(F_i^{\text{lin}}(w; \bar{w})) \leq 0, \quad i = 1, \ldots, m, \\ & G^{\text{lin}}(w; \bar{w}) = 0 \end{aligned}$$

(SCP algorithm is expensive, but multiplier-free and affine-invariant)

# Sequential Linear Programming (SLP)

If functions $\phi_0, \ldots, \phi_m$ are linear, SCP just solves linear programs (LP)

$$\begin{aligned} \underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad & f_0^{\text{lin}}(w; \bar{w}) \\ \text{subject to} \quad & f_i^{\text{lin}}(w; \bar{w}) \leq 0, \quad i = 1, \ldots, m, \\ & G^{\text{lin}}(w; \bar{w}) = 0 \end{aligned}$$

- might be called **Sequential Linear Programming (SLP)** ("Method of Approximation Programming" by Griffith & Stewart, 1961)
- equivalent to standard SQP with zero Hessian
- SLP only attracted to NLP solutions in vertices of feasible set
- works very well for L1-estimation [Bock, Kostina, Schlöder 2007]
- converges **quadratically** once correct active set is discovered

# Constrained Gauss-Newton [Bock 1983]

Use $B_{\mathrm{CGN}}(w) := J_0(w)^\top \nabla^2 \phi_0(F_0(w)) J_0(w)$ and solve convex quadratic program (QP)

$$\begin{aligned} \underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad & f_0^{\mathrm{lin}}(w;\bar{w}) + \frac{1}{2}(w-\bar{w})^\top B_{\mathrm{CGN}}(\bar{w})(w-\bar{w}) \\ \text{subject to} \quad & f_i^{\mathrm{lin}}(w;\bar{w}) \leq 0, \quad i = 1,\ldots,m, \\ & G^{\mathrm{lin}}(w;\bar{w}) = 0 \end{aligned}$$

- like SCP, the method is multiplier free and affine invariant
- QPs are potentially cheaper to solve
- but CGN diverges on some problems where SCP converges

Remark: for least-squares objectives, this method is due to [Bock 1983]. In many papers, Bock's method is called "the Generalized Gauss-Newton (GGN) method". To avoid a notation clash with Schraudolph and the computer science literature, we prefer to call Bock's method "the Constrained Gauss-Newton (CGN) method".

# Sequential Convex Quadratic Programming (SCQP)
[Verschueren et al 2016]

$$B_{\mathrm{SCQP}}(w,\mu) := J_0(w)^\top \nabla^2 \phi_0(F_0(w)) J_0(w) + \sum_{i=1}^{m} \mu_i J_i(w)^\top \nabla^2 \phi_i(F_i(w)) J_i(w)$$

$$\begin{aligned}
\underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad & f_0^{\mathrm{lin}}(w;\bar{w}) + \frac{1}{2}(w-\bar{w})^\top B_{\mathrm{SCQP}}(\bar{w},\bar{\mu})(w-\bar{w}) \\
\text{subject to} \quad & f_i^{\mathrm{lin}}(w;\bar{w}) \leq 0, \quad i = 1,\ldots,m, \quad | \quad \mu^+, \\
& G^{\mathrm{lin}}(w;\bar{w}) = 0
\end{aligned}$$

- obtain pair $(w_{k+1}, \mu_{k+1})$ from solution at $(\bar{w},\bar{\mu}) = (w_k, \mu_k)$
- "optimizer state" contains both, $\bar{w}$ and inequality multipliers $\bar{\mu}$
- again, only a QP needs to be solved in each iteration
- again, affine invariant
- $B_{\mathrm{SCQP}}(w,\mu) \succeq B_{\mathrm{CGN}}(w)$ (more likely to converge than CGN)
- for unconstrained problems, SCQP becomes GGN
- in fact, SCQP has same contraction rate as SCP [Messerer &D., ECC 2020]

# Sequential Quadratically Constrained Quadratic Programming (SQCQP)

[Messerer et al 2021]

$$B_i(w) := J_i(w)^\top \nabla^2 \phi_i(F_i(w)) J_i(w), \quad i = 0, 1, \ldots, m$$

$$\begin{aligned}
\underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad & f_0^{\text{lin}}(w; \bar{w}) + \frac{1}{2}(w - \bar{w})^\top B_0(\bar{w})(w - \bar{w}) \\
\text{subject to} \quad & f_i^{\text{lin}}(w; \bar{w}) + \frac{1}{2}(w - \bar{w})^\top B_i(\bar{w})(w - \bar{w}) \leq 0, \quad i = 1, \ldots, m, , \\
& G^{\text{lin}}(w; \bar{w}) = 0
\end{aligned}$$

- obtain $w_{k+1}$ from solution at $\bar{w} = w_k$
- multiplier free, optimizer state contains only $\bar{w}$
- a quadratically constrained quadratic program (QCQP) needs to be solved in each iteration, e.g. via HPIPM [Frison et al. 2022]
- again, affine invariant
- also SQCQP has same contraction rate as SCP [Messerer et al. 2021]

# Identical Local Convergence of SCP, SCQP, SQCQP

**Theorem 1: Local Convergence of SCP, SCQP, SQCQP** [Messerer et al. 2021, ESAIM Survey]

Regard KKT point $z^* := (w^*, \mu^*, \lambda^*)$ with LICQ and strict complementarity. Denote the reduced Hessian by $\tilde{\Lambda}_*$, the reduced SCQP Hessian by $\tilde{B}_*$ (*) and assume that $\tilde{B}_* \succ 0$. Then

- $z^*$ is a fixed point for all three methods SCP, SCQP and SQCQP
- all three methods are well-defined in a neighborhood of $z^*$
- their linear contraction rates are equal and given by the smallest $\alpha \in \mathbb{R}$ that satisfies the linear matrix inequality

$$-\alpha\tilde{B}_* \preceq \tilde{\Lambda}_* - \tilde{B}_* \preceq \alpha\tilde{B}_* \qquad (3)$$

(*) $\tilde{\Lambda}_* := Z^\top \nabla^2 \mathcal{L}(w^*, \mu^*, \lambda^*) Z$ and $\tilde{B}_* := Z^\top B_{\text{SCQP}}(w^*, \mu^*) Z$ with $Z$ a fixed nullspace basis of the Jacobian of active constraints

# Identical Local Convergence of SCP, SCQP, SQCQP

**Theorem 1: Local Convergence of SCP, SCQP, SQCQP** [Messerer et al. 2021, ESAIM Survey]

Regard KKT point $z^* := (w^*, \mu^*, \lambda^*)$ with LICQ and strict complementarity. Denote the reduced Hessian by $\tilde{\Lambda}_*$, the reduced SCQP Hessian by $\tilde{B}_*$ (*) and assume that $\tilde{B}_* \succ 0$. Then

- $z^*$ is a fixed point for all three methods SCP, SCQP and SQCQP
- all three methods are well-defined in a neighborhood of $z^*$
- their linear contraction rates are equal and given by the smallest $\alpha \in \mathbb{R}$ that satisfies the linear matrix inequality

$$-\alpha \tilde{B}_* \preceq \tilde{\Lambda}_* - \tilde{B}_* \preceq \alpha \tilde{B}_* \qquad (3)$$

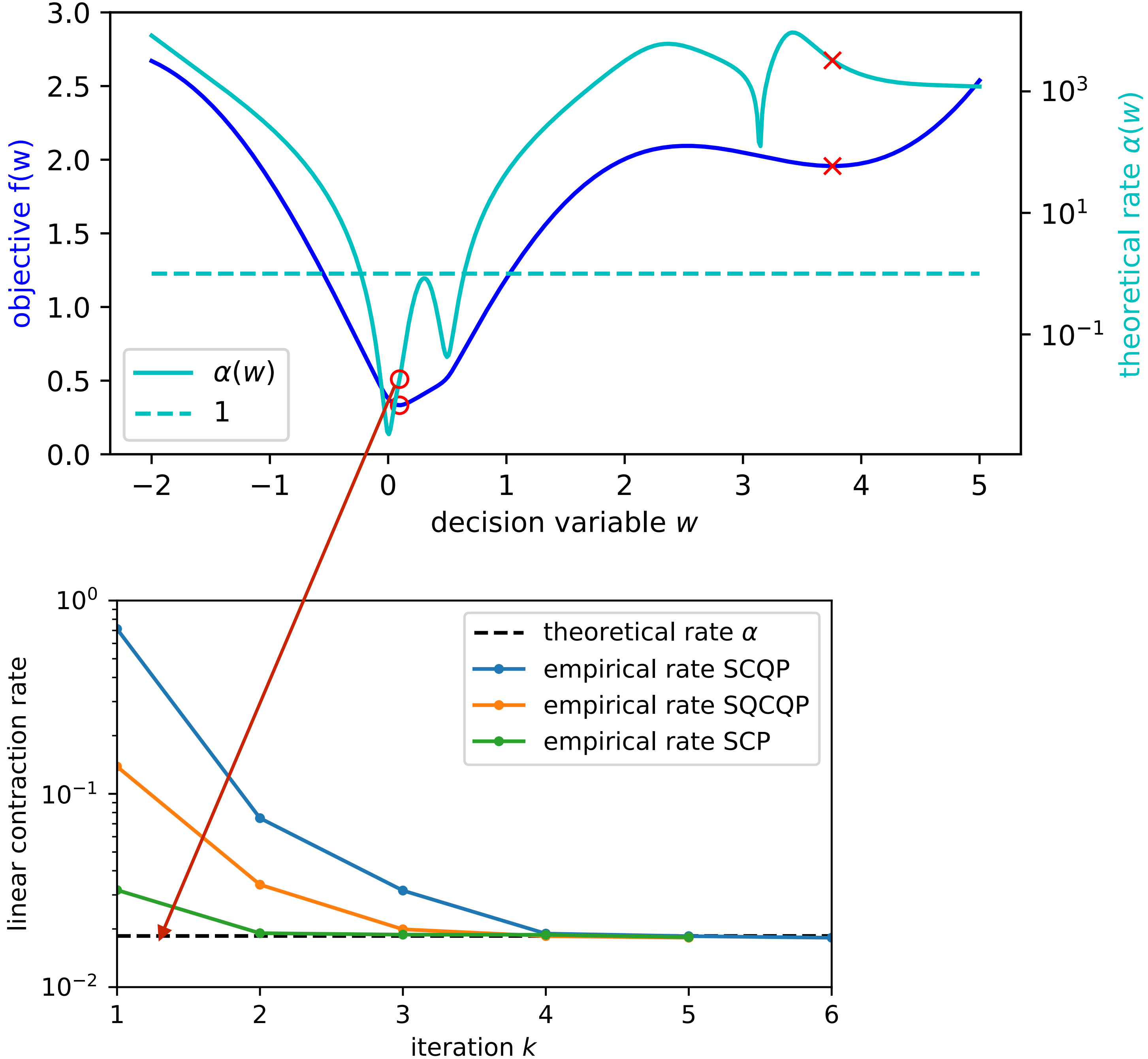
(*) $\tilde{\Lambda}_* := Z^\top \nabla^2 \mathcal{L}(w^*, \mu^*, \lambda^*) Z$ and $\tilde{B}_* := Z^\top B_{\mathrm{SCQP}}(w^*, \mu^*) Z$ with $Z$ a fixed nullspace basis of the Jacobian of active constraints

**Corollary**

Necessary condition for local convergence of all methods is $\tilde{B}_* \succeq \frac{1}{2}\tilde{\Lambda}_* \succeq 0$

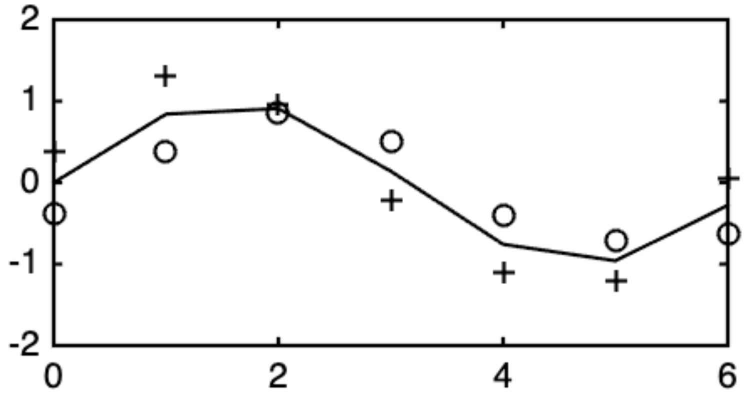
Proof of corollary: Set $\alpha = 1$ in (3).

# Tutorial Example: Objective and Local Contraction Rate



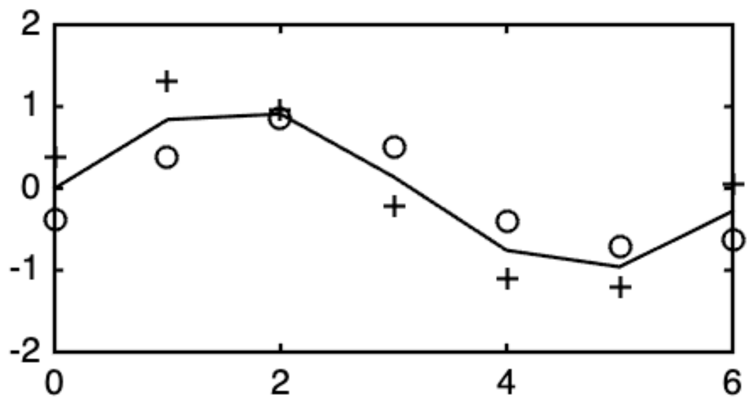
# Desirable Divergence and Mirror Problem [cf. Bock 1987]

SCP and GGN do not converge to every local minimum. This can help to avoid "bad" local minima, as discussed next.




Regard maximum likelihood estimation problem $\boxed{\min_w \phi(M(w) - y)}$ with nonlinear model $M : \mathbb{R}^n \to \mathbb{R}^N$ and measurements $y \in \mathbb{R}^N$. Assume penalty $\phi$ is symmetric with $\phi(-z) = \phi(z)$ as is the case for symmetric error distributions. At a solution $w^*$, we can generate "mirror measurements" $y_{\mathrm{mr}} := 2M(w^*) - y$ obtained by reflecting the residuals.
From a statistical point of view, $y_{\mathrm{mr}}$ should be as likely as $y$.

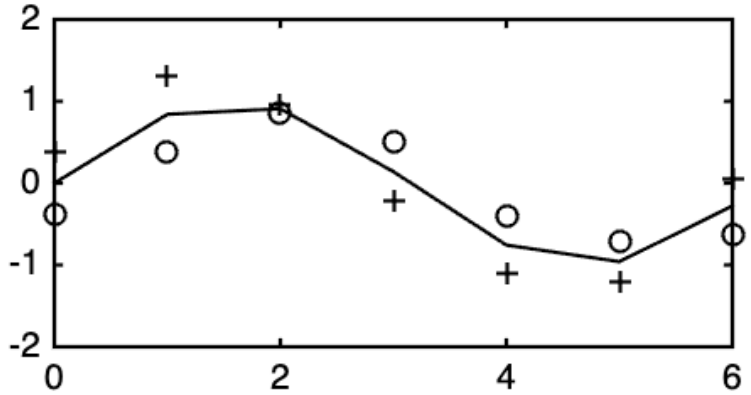
# SCP divergence ⇔ minimum unstable under mirroring




**Theorem** [Messerer and D., 2019/2020] generalizing [Bock 1987]

Regard a local minimizer $w^*$ of $\phi(M(w) - y)$ that satisfies SOSC. If the necessary SCP convergence condition $\tilde{B}_* \succeq \frac{1}{2}\tilde{\Lambda}_*$ does not hold, then $w^*$ is a stationary point of the mirror problem but **not** a local minimizer.

# SCP divergence ⇔ minimum unstable under mirroring

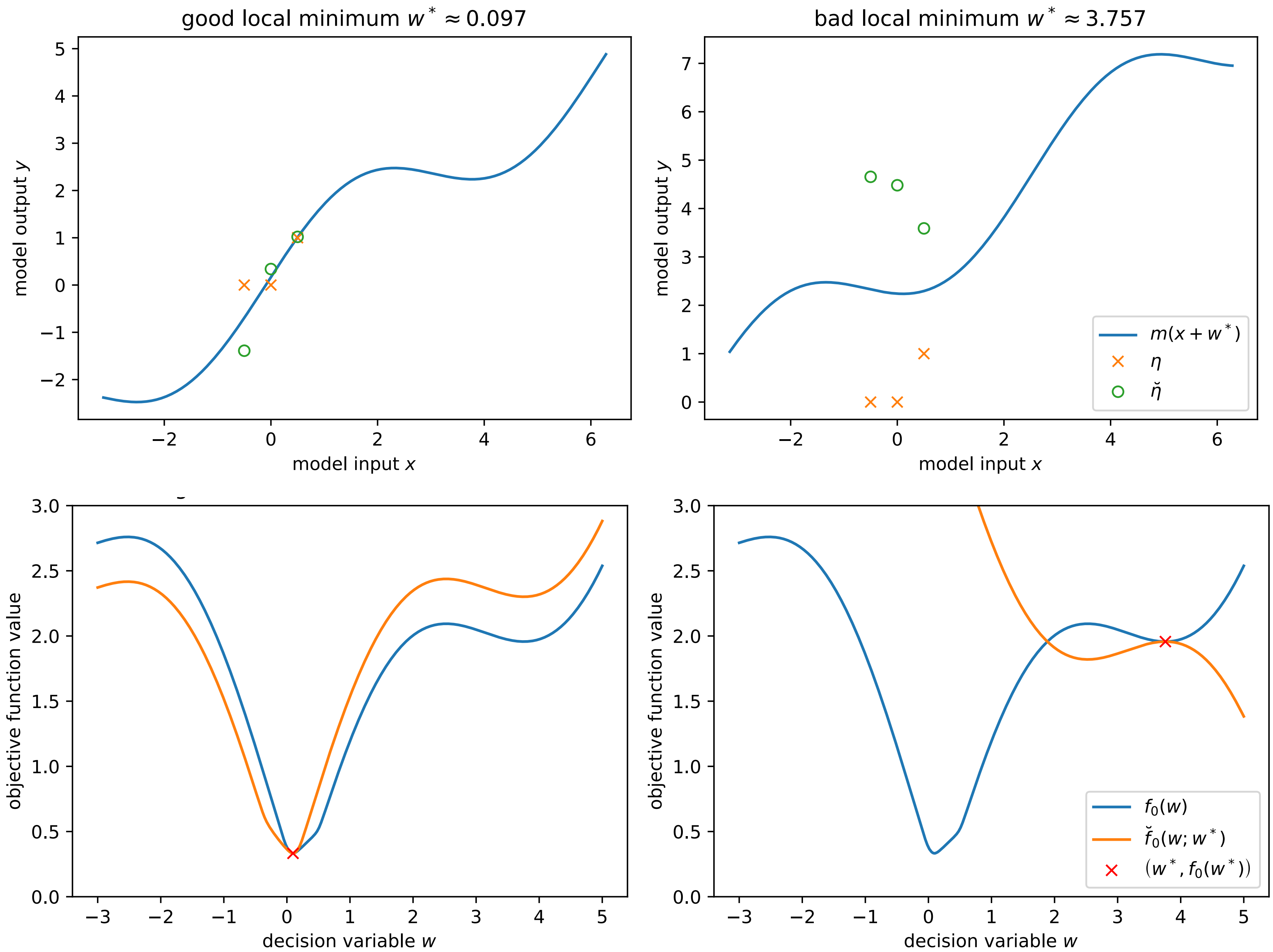



**Theorem** [Messerer and D., 2019/2020] generalizing [Bock 1987]

Regard a local minimizer $w^*$ of $\phi(M(w) - y)$ that satisfies SOSC. If the necessary SCP convergence condition $\tilde{B}_* \succeq \frac{1}{2}\tilde{\Lambda}_*$ does not hold, then $w^*$ is a stationary point of the mirror problem but **not** a local minimizer.

*Sketch of proof (unconstrained): use $M(w^*) - y_{\mathrm{mr}} = y - M(w^*)$ to show that $\nabla f_{\mathrm{mr}}(w^*) = J(w^*)^\top (y - M(w^*)) = 0$ and
$\nabla^2 f_{\mathrm{mr}}(w^*) = B_{\mathrm{GGN}}(w^*) - E_{\mathrm{GGN}}(w^*) = 2B_{\mathrm{GGN}}(w^*) - \nabla^2 f(w^*) \not\succeq 0$

# Tutorial Example and Mirror Problems at Different Local Minima



# Overview

- Model Predictive Control - Examples and Optimization Problems
- Convexity Exploiting Newton-Type Optimization
  - Sequential Convex Programming (SCP)
  - Generalized Gauss-Newton (GGN)
  - + SLP, CGN, SCQP, SQCQP
- Zero-Order Optimization-based Iterative Learning Control
  - Tutorial Example
  - Bounding the Loss of Optimality and Exactness
- Mixed Integer Optimal Control
  - Problem Statement
  - Three Step Algorithm
  - Application to Renewable Energy System in Karlsruhe

# Two Ingredients of Newton-Type Optimization

The convexity exploiting algorithms presented so far need two ingredients:

1. a good nonlinear model and its linearisation, and
2. convex substructure in objective and constraints

# Two Ingredients of Newton-Type Optimization

The convexity exploiting algorithms presented so far need two ingredients:

1. a good nonlinear model and its linearisation, and
2. convex substructure in objective and constraints

**Which of the two is more important for success in data-driven optimization?**

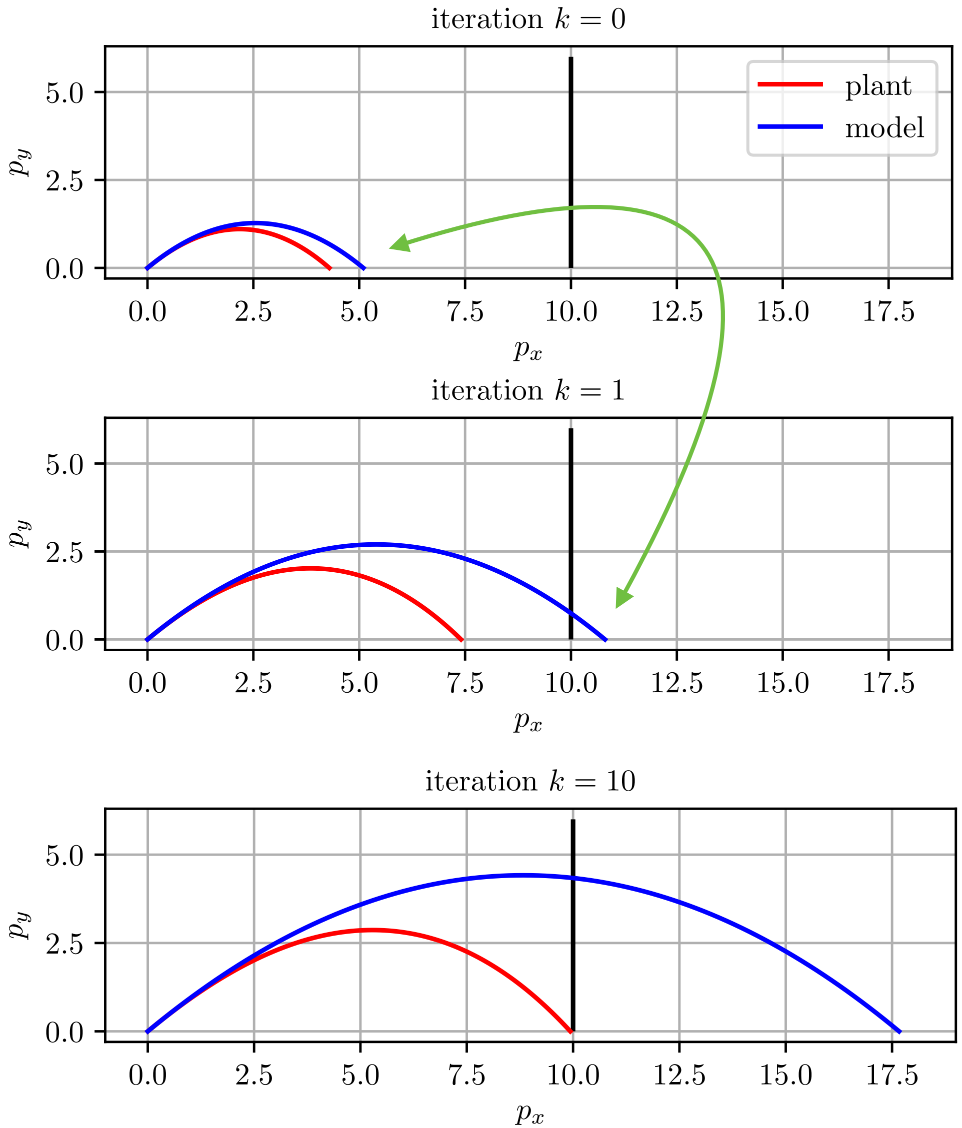
# Iterative Learning Control for Lemon-Ball Throwing

# Iterative Learning of Ball Throwing with Minimal Energy

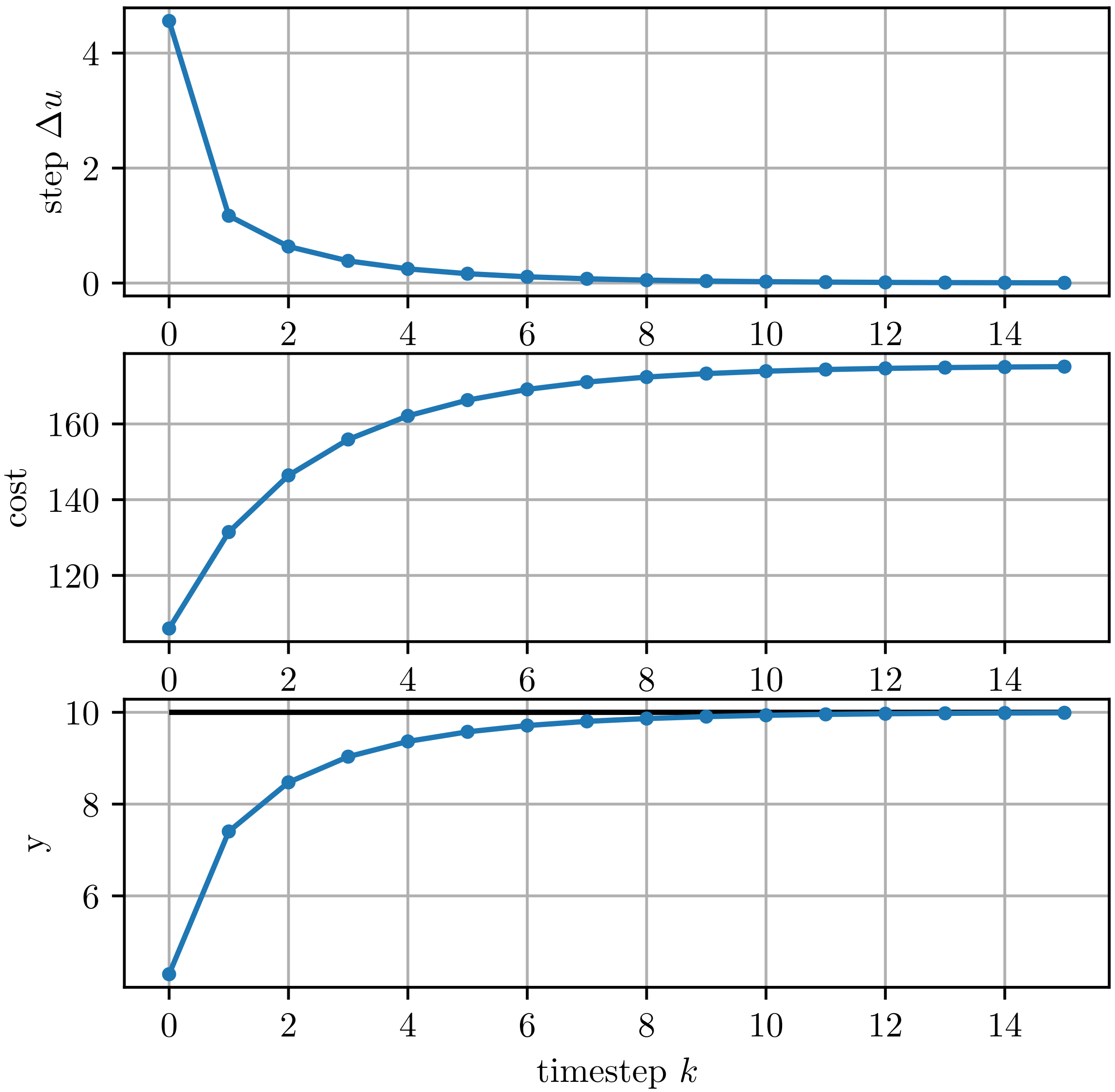
## Experiments conducted by Katrin Baumgärtner

- Model $F_\mathrm{M}(u)$ maps initial velocity $u \in \mathbb{R}^2$ to landing position $y \in \mathbb{R}$
- Aim: throw ball further than $y \geq 10$ with minimal initial velocity
- Experiments with "real plant" give pairs $(u_k, y_k)$ [shorter distance than predicted]
- We can use $(u_k, y_k)$ to correct the model, and iteratively obtain $u_{k+1}$ by solving the following optimization problem:

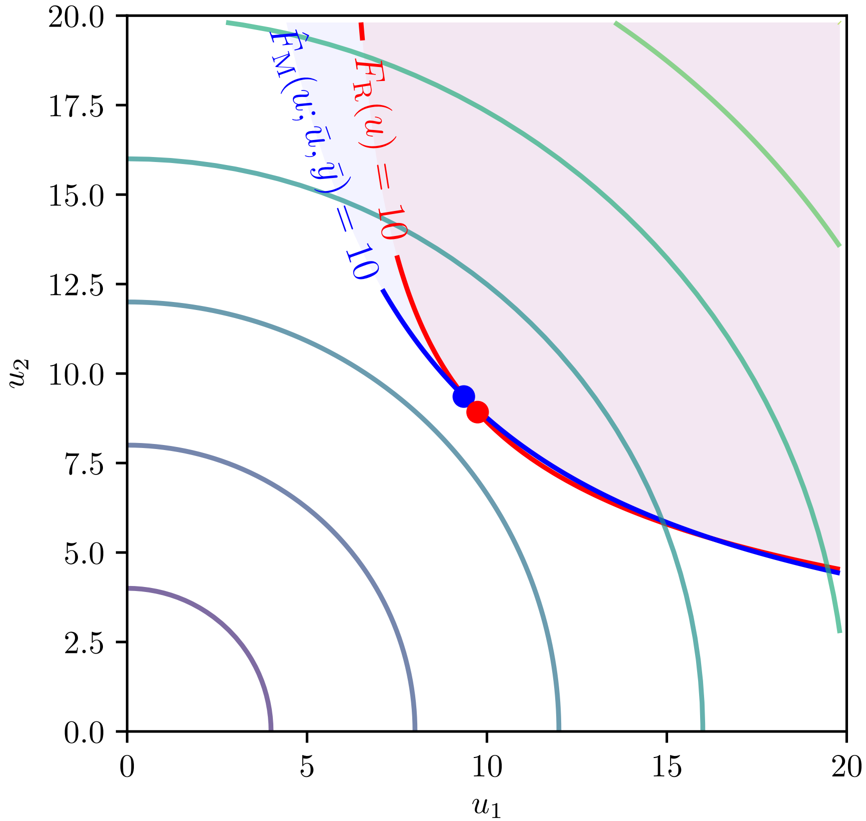
$$\begin{aligned} \underset{u \in \mathbb{R}^2,\, y \in R}{\text{minimize}} \quad & \|u\|_2^2 \\ \text{subject to} \quad & y = \underbrace{F_\mathrm{M}(u) - F_\mathrm{M}(u_k) + y_k}_{=:\tilde{F}_M(u;u_k,y_k)}, \\ & y \geq 10 \end{aligned}$$



# Iterations of Algorithm and Reduced Problem Visualization




$$\begin{array}{ll} \underset{u \in \mathbb{R}^2}{\text{minimize}} & \|u\|_2^2 \\ \text{subject to} & \tilde{F}_M(u; u_k, y_k) \geq 10 \end{array}$$



# Zero Order Optimization-based Iterative Learning Control (ZOO-ILC)

Aim: optimization with unknown input-output system $y = F_{\mathrm{R}}(u)$ ("reality"):

$$\begin{aligned} \underset{u,\,y}{\text{minimize}} \quad & \phi(u,y) \\ \text{subject to} \quad & F_{\mathrm{R}}(u) - y = 0, \\ & H(u,y) \leq 0 \end{aligned} \tag{4}$$

# Zero Order Optimization-based Iterative Learning Control (ZOO-ILC)

Aim: optimization with unknown input-output system $y = F_{\mathrm{R}}(u)$ ("reality"):

$$\begin{aligned} \underset{u,\,y}{\text{minimize}} \quad & \phi(u,y) \\ \text{subject to} \quad & F_{\mathrm{R}}(u) - y = 0, \\ & H(u,y) \leq 0 \end{aligned} \tag{4}$$

ZOO-ILC idea [cf. Schöllig, Volkaert, Zeilinger]: use trial input $u_k$ with output $y_k$ and a model $F_{\mathrm{M}}$ to obtain new trial input $u_{k+1}$ from solution of

$$\begin{aligned} \underset{u,\,y}{\text{minimize}} \quad & \phi(u,y) \\ \text{subject to} \quad & \color{red}{F_{\mathrm{M}}(u)} - y = F_{\mathrm{M}}(u_k) - y_k, \\ & H(u,y) \leq 0 \end{aligned} \tag{5}$$

Questions: Does this method converge? What is its loss of optimality?

# Feasibility and Loss of Optimality of ZOO-ILC

**Theorem 2** [Baumgärtner et al., in preparation]

For any fixed point $(\bar{u}, \bar{y})$ of the ZOO-ILC algorithm with multipliers $(\bar{\lambda}, \bar{\mu})$ holds under mild conditions:

- $(\bar{u}, \bar{y})$ is feasible for the real problem
- the loss of optimality compared to a real solution $(u_\mathrm{R}, y_\mathrm{R})$ is bounded by:

$$\phi(\bar{u}, \bar{y}) - \phi(u_\mathrm{R}, y_\mathrm{R}) \leq \bar{\lambda}^\top \left(J_\mathrm{M}(\bar{u}) - J_\mathrm{R}(\bar{u})\right)(u_\mathrm{R} - \bar{u})$$

Here, the Lagrangian of the model problem is given by

$$\mathcal{L}(u, y, \lambda, \mu) = \phi(u, y) + \lambda^\top (F_\mathrm{M}(u) - y - b_k) + \mu^\top H(u, y)$$

and $J_\mathrm{M}(u)$ and $J_\mathrm{R}(u)$ are the Jacobians of $F_\mathrm{M}(u)$ and $F_\mathrm{R}(u)$.

# Special cases where ZOO-ILC delivers a lossless solution

$$\boxed{\phi(\bar{u},\bar{y}) - \phi(u_{\mathrm{R}}, y_{\mathrm{R}}) \leq \bar{\lambda}^\top \left(J_{\mathrm{M}}(\bar{u}) - J_{\mathrm{R}}(\bar{u})\right)(u_{\mathrm{R}} - \bar{u})}$$

ZOO-ILC delivers lossless solution in the following three cases:

1. Tracking ILC with zero residual (standard ILC):

$$\bar{\lambda} = 0$$

2. Model and real Jacobian coincide at solution (rarely the case):

$$J_{\mathrm{M}}(\bar{u}) - J_{\mathrm{R}}(\bar{u}) = 0$$

3. Constrained problems where solution $u_{\mathrm{R}}$ is in vertex of the reduced feasible set:

$$u_{\mathrm{R}} - \bar{u} = 0$$

(if the Jacobian error is small enough, LICQ and strict complementarity hold)

# Special cases where ZOO-ILC delivers a lossless solution

$$\boxed{\phi(\bar{u}, \bar{y}) - \phi(u_{\mathrm{R}}, y_{\mathrm{R}}) \leq \bar{\lambda}^\top \left(J_{\mathrm{M}}(\bar{u}) - J_{\mathrm{R}}(\bar{u})\right)(u_{\mathrm{R}} - \bar{u})}$$

ZOO-ILC delivers lossless solution in the following three cases:

1. Tracking ILC with zero residual (standard ILC):

   $$\bar{\lambda} = 0$$

2. Model and real Jacobian coincide at solution (rarely the case):

   $$J_{\mathrm{M}}(\bar{u}) - J_{\mathrm{R}}(\bar{u}) = 0$$

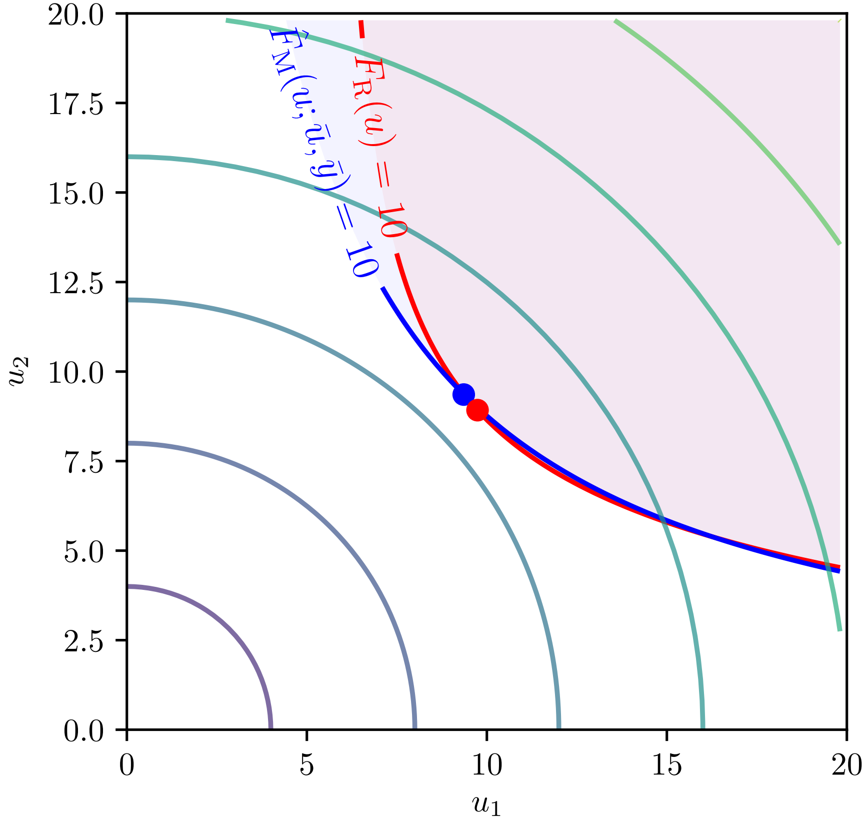
3. Constrained problems where solution $u_{\mathrm{R}}$ is in vertex of the reduced feasible set:

   $$u_{\mathrm{R}} - \bar{u} = 0$$

   (if the Jacobian error is small enough, LICQ and strict complementarity hold)

# Solutions for $L_2$- and $L_\infty$-norm minimisation

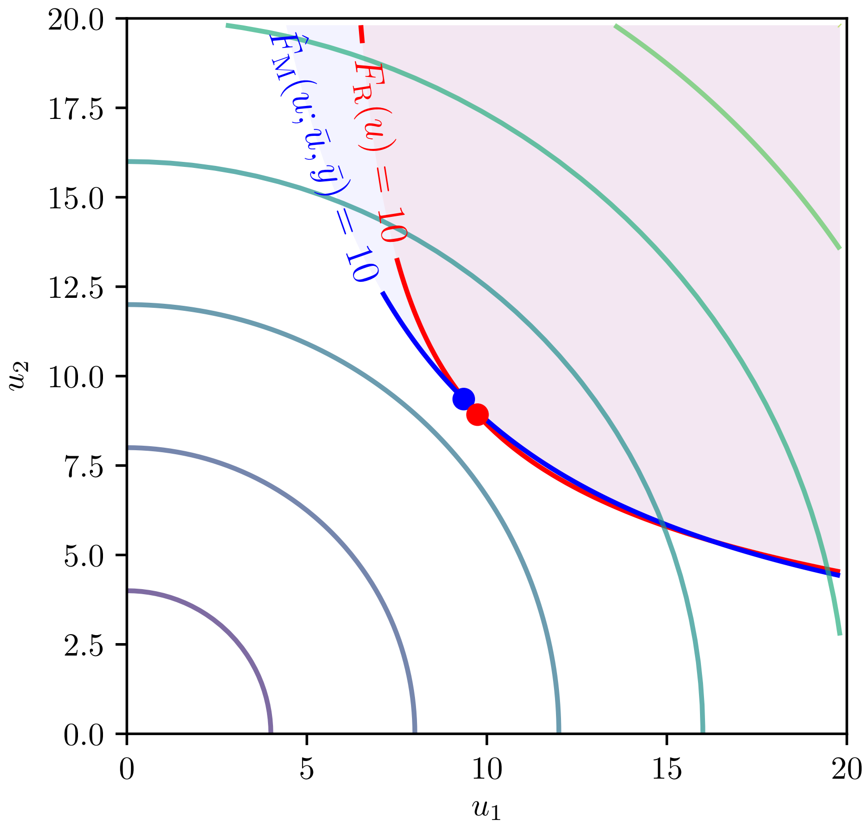
$$\begin{aligned} \underset{u \in \mathbb{R}^2}{\text{minimize}} \quad & \|u\|_2^2 \\ \text{subject to} \quad & \tilde{F}_M(u; u_k, y_k) \geq 10 \end{aligned}$$




suboptimality: $0.874 \leq 1.377$ (bound)

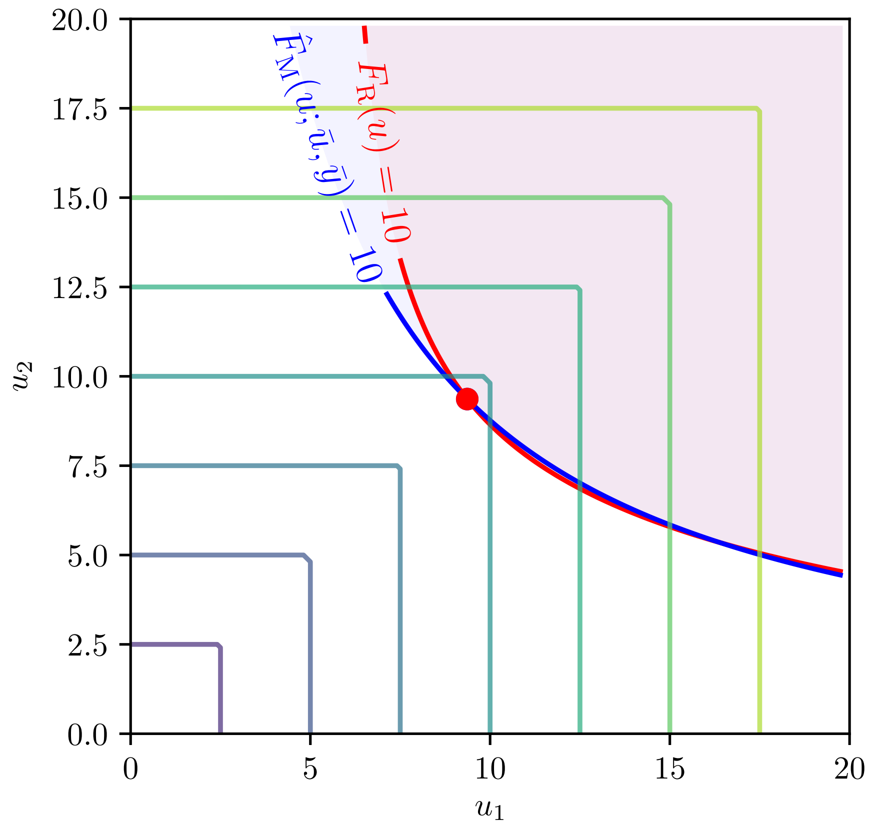
# Solutions for $L_2$- and $L_\infty$-norm minimisation

$$\underset{u \in \mathbb{R}^2}{\text{minimize}} \quad \|u\|_2^2 \qquad \text{subject to} \quad \tilde{F}_M(u; u_k, y_k) \geq 10$$




suboptimality: $0.874 \leq 1.377$ (bound)

$$\underset{u \in \mathbb{R}^2}{\text{minimize}} \quad \|u\|_\infty \qquad \text{subject to} \quad \tilde{F}_M(u; u_k, y_k) \geq 10$$




**solution in vertex, no loss of optimaliy**

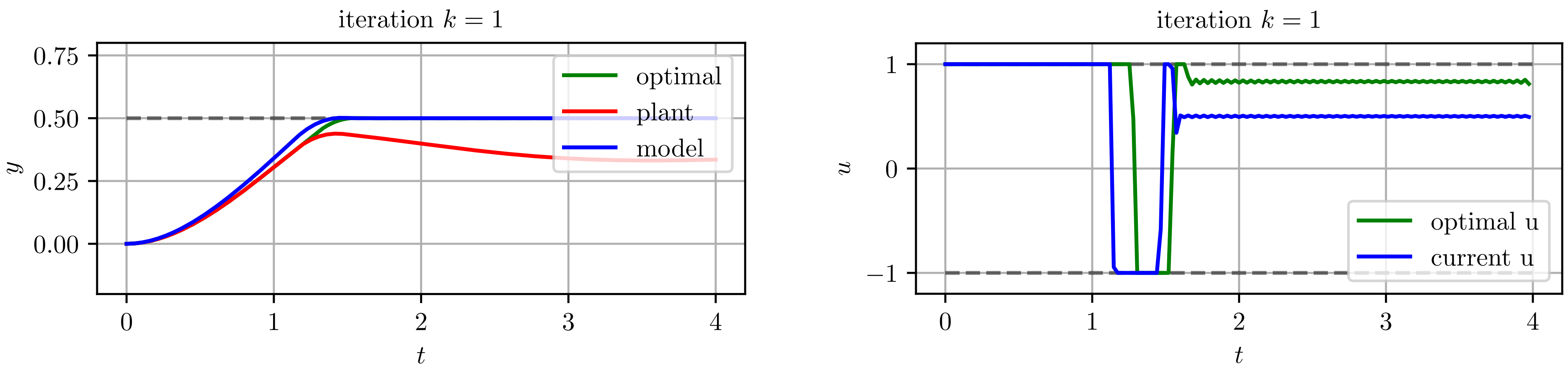
# Time-Optimal Motion of an oscillator (L1-tracking)

**Real plant:** $T^2\ddot{y} + 2Td\dot{y} + y + \beta y^3 = K_\mathrm{R} u$
with $T = 1$, $d = 0.5$, $\beta = 2$, $K_\mathrm{R} = 0.9$

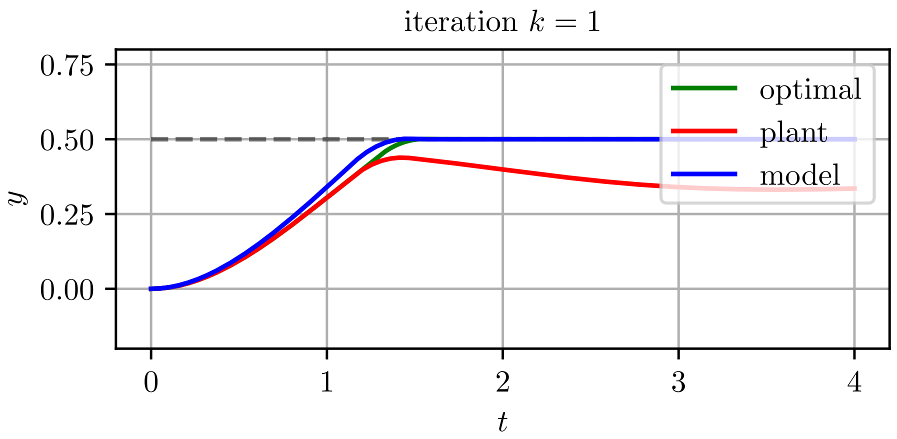
**Model:** $T^2\ddot{y} + 2Td\dot{y} + y = K_\mathrm{M} u$
with $K_\mathrm{M} = 1$

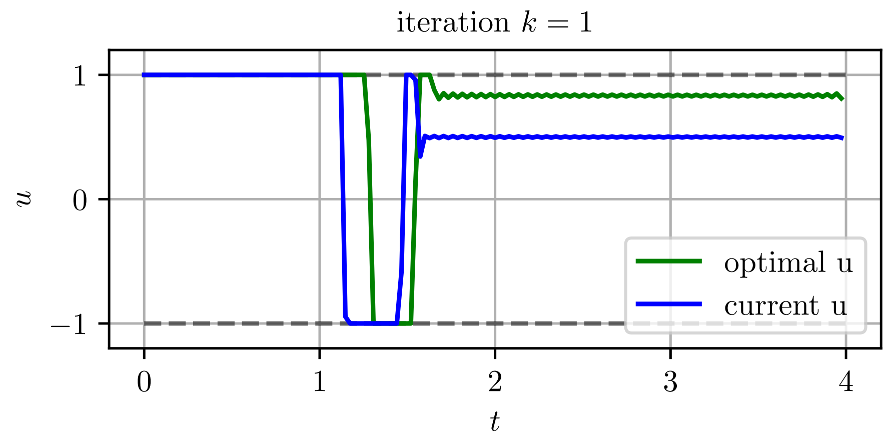
$$\begin{aligned}
\underset{y(\cdot),\,u(\cdot)}{\text{minimize}} \quad & \int_0^{T_\mathrm{H}} |y(t) - y_\mathrm{ref}| + \alpha u(t)^2 \,\mathrm{d}t \\
\text{subject to} \quad & y(t) = F_\mathrm{M}(t;u) + y_k(t) - F_\mathrm{M}(t;u_k), \\
& |u(t)| \le 1, \quad t \in [0, T_\mathrm{H}]
\end{aligned}$$

$\alpha = 10^{-4}$

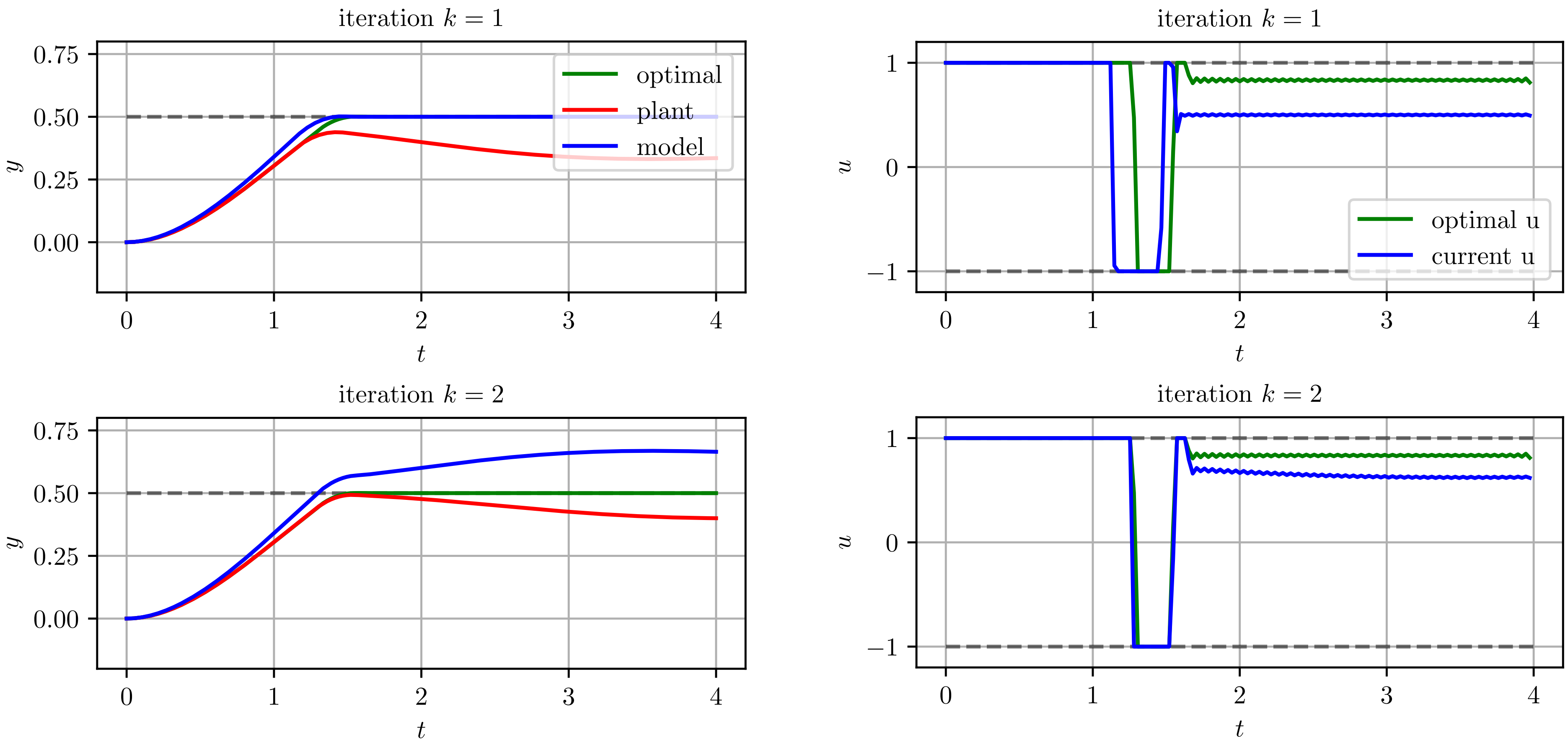


# Time-Optimal Motion of an oscillator (L1-tracking)

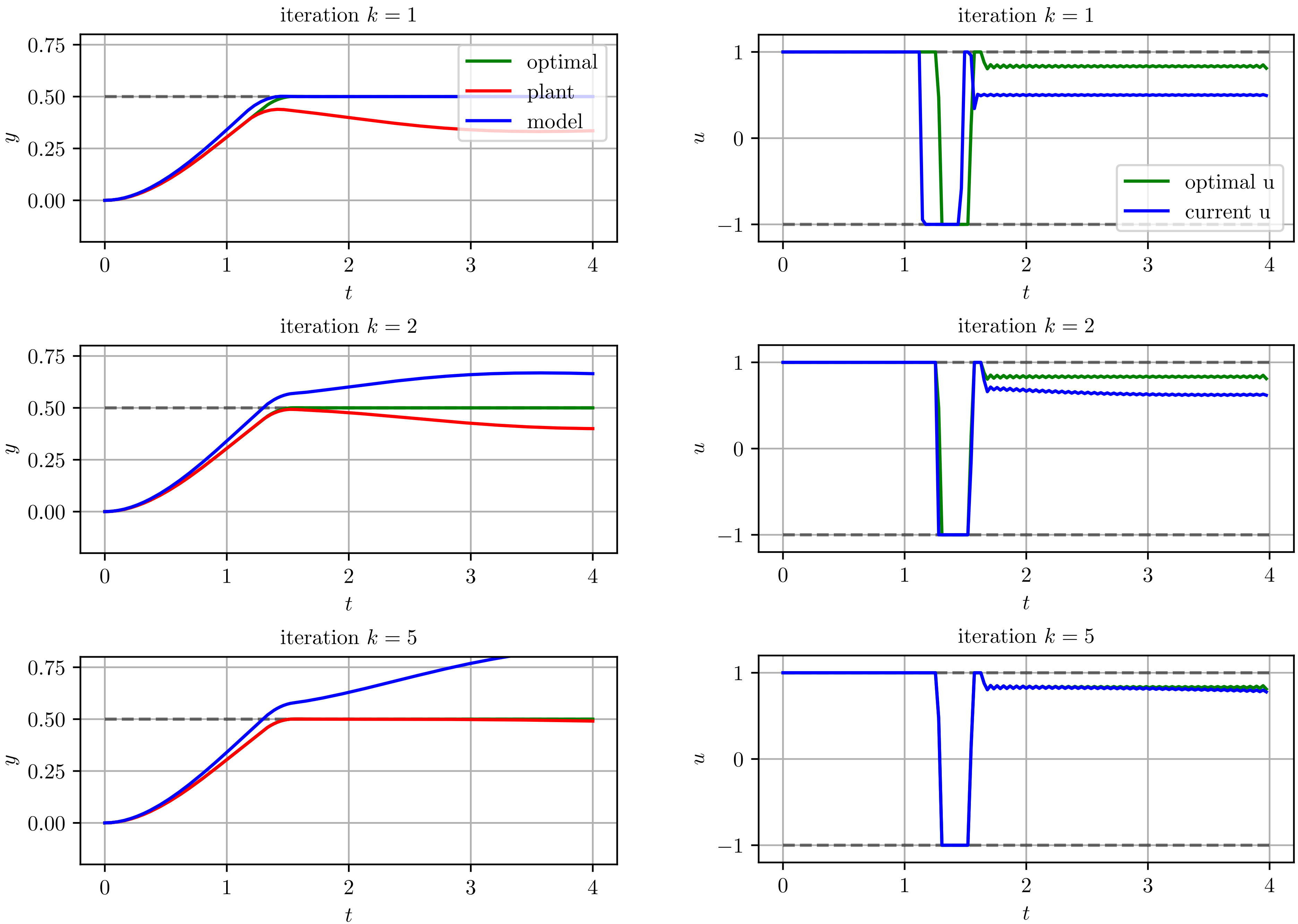


# Time-Optimal Motion of an oscillator (L1-tracking)



# Time-Optimal Motion of an oscillator (L1-tracking)



# When does the ZOO-ILC method converge?

**Theorem 3 (Convergence of ZOO-ILC)** [Baumgärtner et al., in preparation]

Regard a fixed point $\bar{z} = (\bar{u}, \bar{y}, \bar{\lambda}, \bar{\mu}_{\mathcal{A}})$ of ZOO-ILC and assume it satisfies LICQ, SOSC and strict complementarity in the model problem. Then the local contraction rate is given by the spectral radius $\rho(A)$ of the matrix

$$A \quad := \quad [\mathbb{I}_{n_u} \quad 0 \quad 0 \quad 0] \left( \frac{\partial R}{\partial z}(\bar{z}; \bar{u}, \bar{y}) \right)^{-1} \begin{bmatrix} 0 \\ 0 \\ J_{\mathrm{M}}(\bar{u}) - J_{\mathrm{R}}(\bar{u}) \\ 0 \end{bmatrix}$$

The ZOO-ILC method converges if $\rho(A) < 1$ and diverges if $\rho(A) > 1$.

Here, $\mu_{\mathcal{A}}$ are the active constraint multipliers and $R(z; u', y')$ is defined by

$$R(z; u', y') := \begin{bmatrix} \nabla_u \mathcal{L}_{\mathrm{M}}(u, y, \lambda, \mu_{\mathcal{A}}; u', y') \\ \nabla_y \mathcal{L}_{\mathrm{M}}(u, y, \lambda, \mu_{\mathcal{A}}; u', y') \\ F_{\mathrm{M}}(u) - y + y' - F_{\mathrm{M}}(u') \\ H_{\mathcal{A}}(u, y) \end{bmatrix}$$

where the Lagrangian of the model problem is given by

$$\mathcal{L}_{\mathrm{M}}(u, y, \lambda, \mu_{\mathcal{A}}; u', y') = \phi(u, y) + \lambda^\top (F_{\mathrm{M}}(u) - y + y' - F_{\mathrm{M}}(u')) + \mu_{\mathcal{A}}^\top H_{\mathcal{A}}(u, y)$$

and $J_{\mathrm{M}}(u)$ and $J_{\mathrm{R}}(u)$ are the Jacobians of $F_{\mathrm{M}}(u)$ and $F_{\mathrm{R}}(u)$.

# When does the ZOO-ILC method converge?

$$\begin{bmatrix} 0 \\ 0 \\ J_{\mathrm{M}}(\bar{u}) - J_{\mathrm{R}}(\bar{u}) \\ 0 \end{bmatrix}$$

# When does the ZOO-ILC method converge?

$$\begin{bmatrix} 0 \\ 0 \\ J_{\mathrm{M}}(\bar{u}) - J_{\mathrm{R}}(\bar{u}) \\ 0 \end{bmatrix}$$

Contraction rate grows with distance between model and real Jacobian.

# Overview

- Model Predictive Control - Examples and Optimization Problems
- Convexity Exploiting Newton-Type Optimization
  - Sequential Convex Programming (SCP)
  - Generalized Gauss-Newton (GGN)
  - + SLP, CGN, SCQP, SQCQP
- Zero-Order Optimization-based Iterative Learning Control
  - Tutorial Example
  - Bounding the Loss of Optimality and Exactness
- Mixed Integer Optimal Control
  - Problem Statement
  - Three Step Algorithm
  - Application to Renewable Energy System in Karlsruhe

# Mixed Integer Optimal Control Problem with Binary Inputs (in outer convexified form)

$$\underset{x(\cdot),u(\cdot),b(\cdot),s(\cdot)}{\text{minimize}} \quad \int_0^T L\big(x,u,b,s\big)\;\mathrm{d}t + M\big(x(T)\big) \tag{2a}$$

$$\text{subject to} \quad x(0) = \bar{x}_0 \tag{2b}$$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sum_{i=1}^{n_b} b_i \cdot f_i\big(x,u,c\big), \quad \sum_{i=1}^{n_b} b_i(t) = 1, \tag{2c}$$

$$b_i(t) \in \{0,1\} \quad \Big[\Leftrightarrow 0 \le b_i(t) \perp (1 - b_i(t)) \ge 0\Big] \quad \text{for } \; i = 1, \dots n_b, \tag{2d}$$

$$-s + r_\mathrm{l} \le r\big(x,u,b,c\big) \le r_\mathrm{u} + s, \qquad \text{for } \; t \in \big[0,T\big] \tag{2e}$$

( + additional combinatorial constraints) (2f)

$x(t)$: states, $u(t)$: continuous controls, $b(t)$: binary controls, $s(t)$: slack variables

$c(t)$: time-varying parameters, $f$: system dynamics, $r_\mathrm{l} \le r \le r_\mathrm{u}$: path constraints

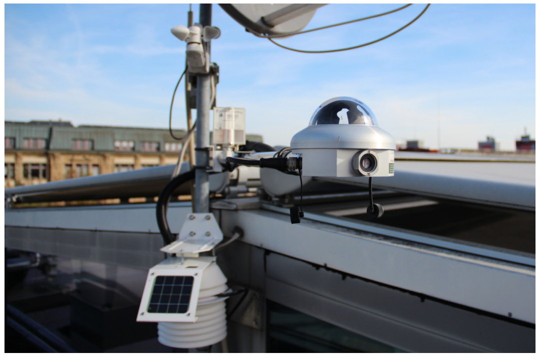
# NMPC for solar thermal test plant at Karlsruhe University of Applied Sciences [Bürger et al. 2019]


Control cabinet, cold storage, ACM, hot storage, pumps (cellar)

Plate collectors (roof)

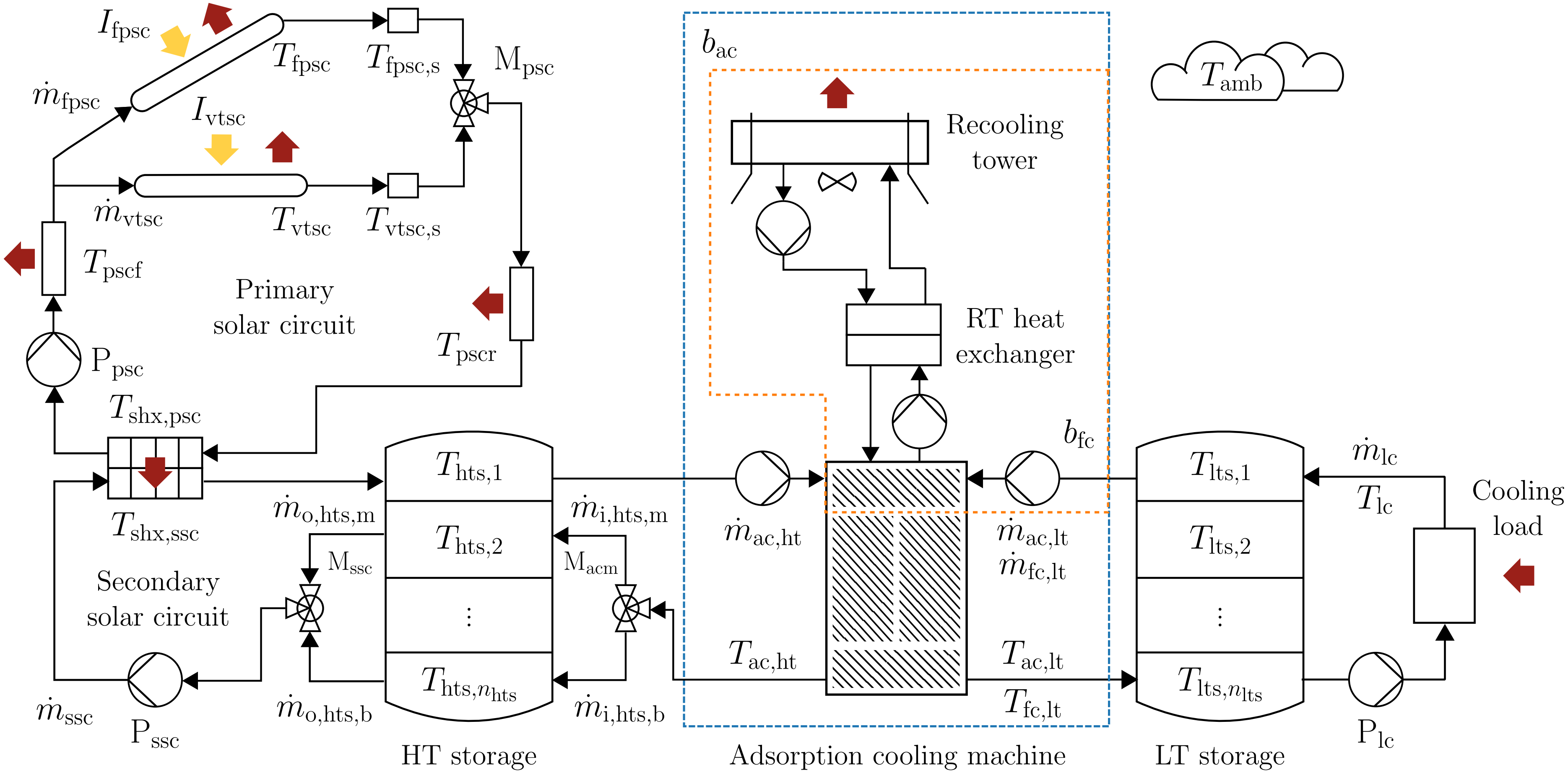
Vacuum tube collectors (roof)

Recooling unit (roof)



Ambient sensors (roof)

# Control-oriented modeling of the solar thermal system



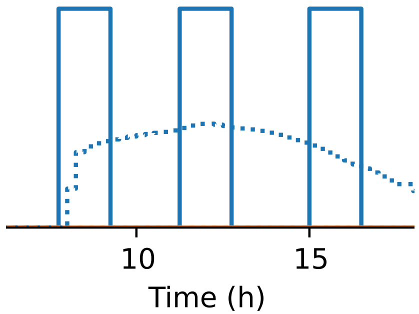

Nonlinear switched system ODE model with $n_x = 20$, $n_b = 2$, $n_u = 5$, and $n_c = 4$,

differentiable in all arguments within the domain of interest

# Three Step Decomposition with CIA Norm
[Sager et al., 2011]

1. Solve Nonlinear Optimal Control Problem with Relaxed Integer Controls, using direct collocation or multiple shooting and a nonlinear programming (NLP) solver.

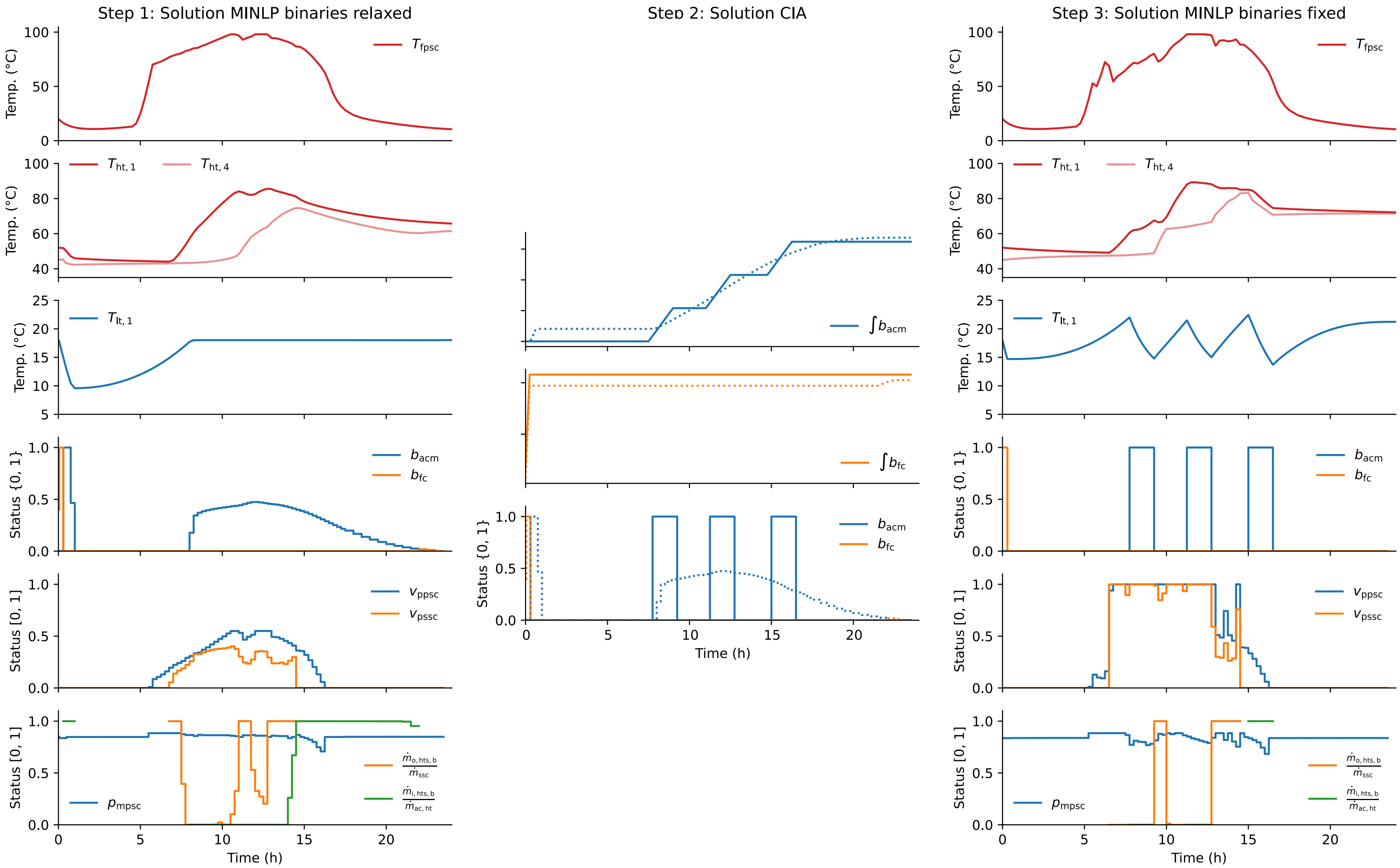
2. Find the "combinatorial integral approximation (CIA) input trajectory that
   (a) satisfies all combinatorial constraints and
   (b) minimises the integrated difference to the relaxed input trajectory




(**pycombina** algorithm is 10-100x times faster than standard MILP solver)

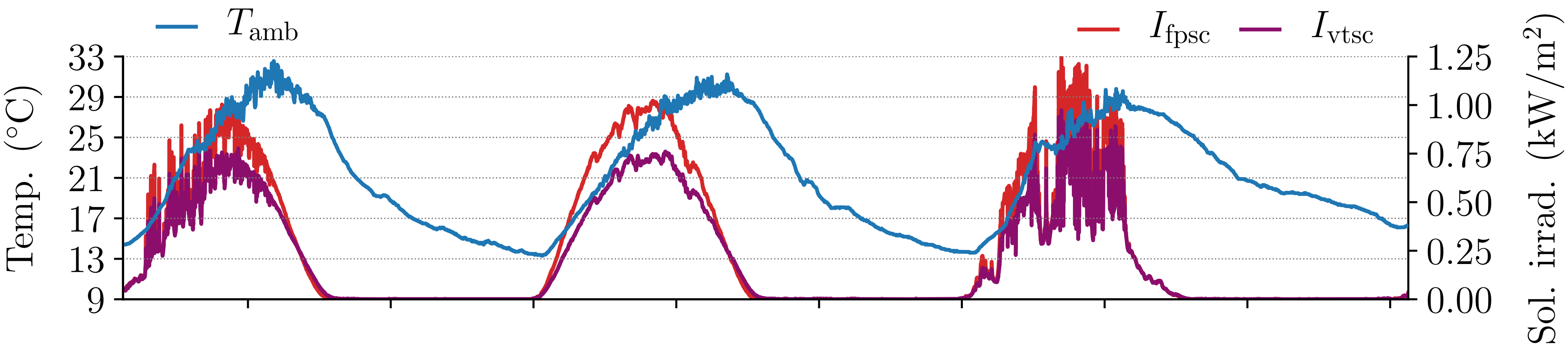
3. Fix the integer inputs and reoptimize over all remaining variables by solving another NLP.

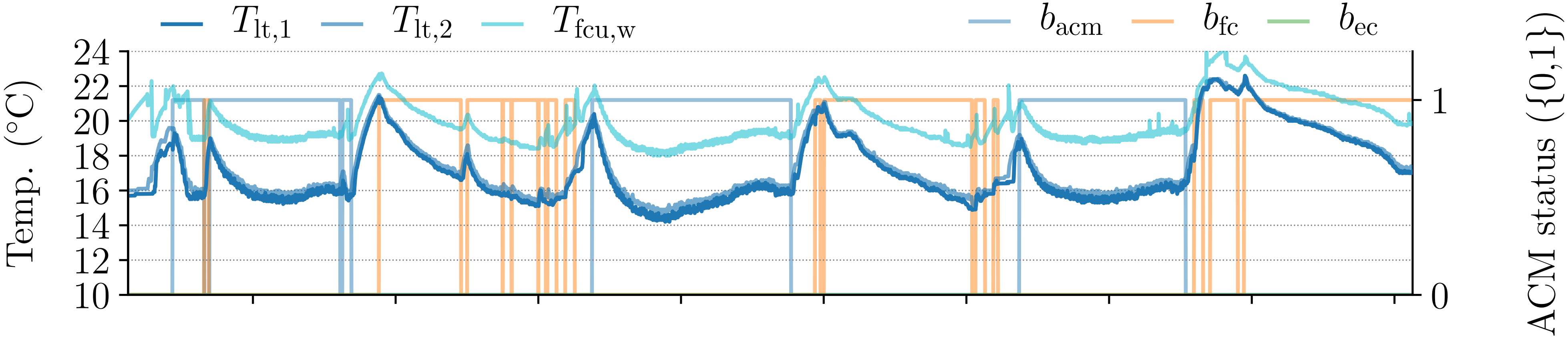
# Numerical Results: Three Step CIA Algorithm




(25 CPU sec) (0.02 CPU sec) (18 CPU sec)

# Experimental Results from Sept 14-17, 2019



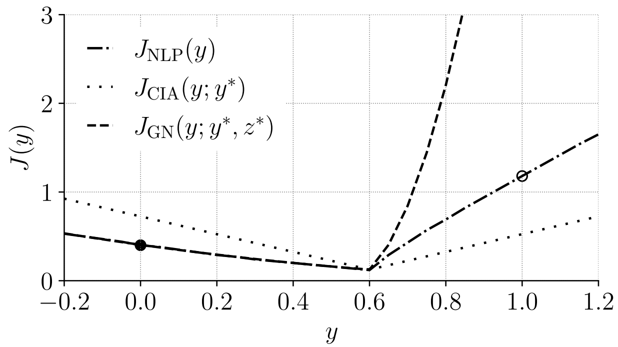




Every 2 minutes, a new nonlinear mixed integer optimal control problem is solved, using a real-time algorithm based on CasADi, IPOPT [Wächter and Biegler 2006], and Pycombina [Bürger et al, 2019], an implementation of the combinatorial integral approximation (CIA) method [Sager et al 2011].

# Alternative to CIA Decomposition: Gauss-Newton based MIQP

## [Bürger et al., in preparation]

- Derive convex Gauss-Newton-type approximation of original MINLP from linearization at relaxed MINLP solution.
- Solution of resulting MIQP can yield improved integer solution in terms of objective and feasibility of the original MINLP.
- MIQP is equivalent to minimization of a distance function that is a first order accurate approximation of the true objective.



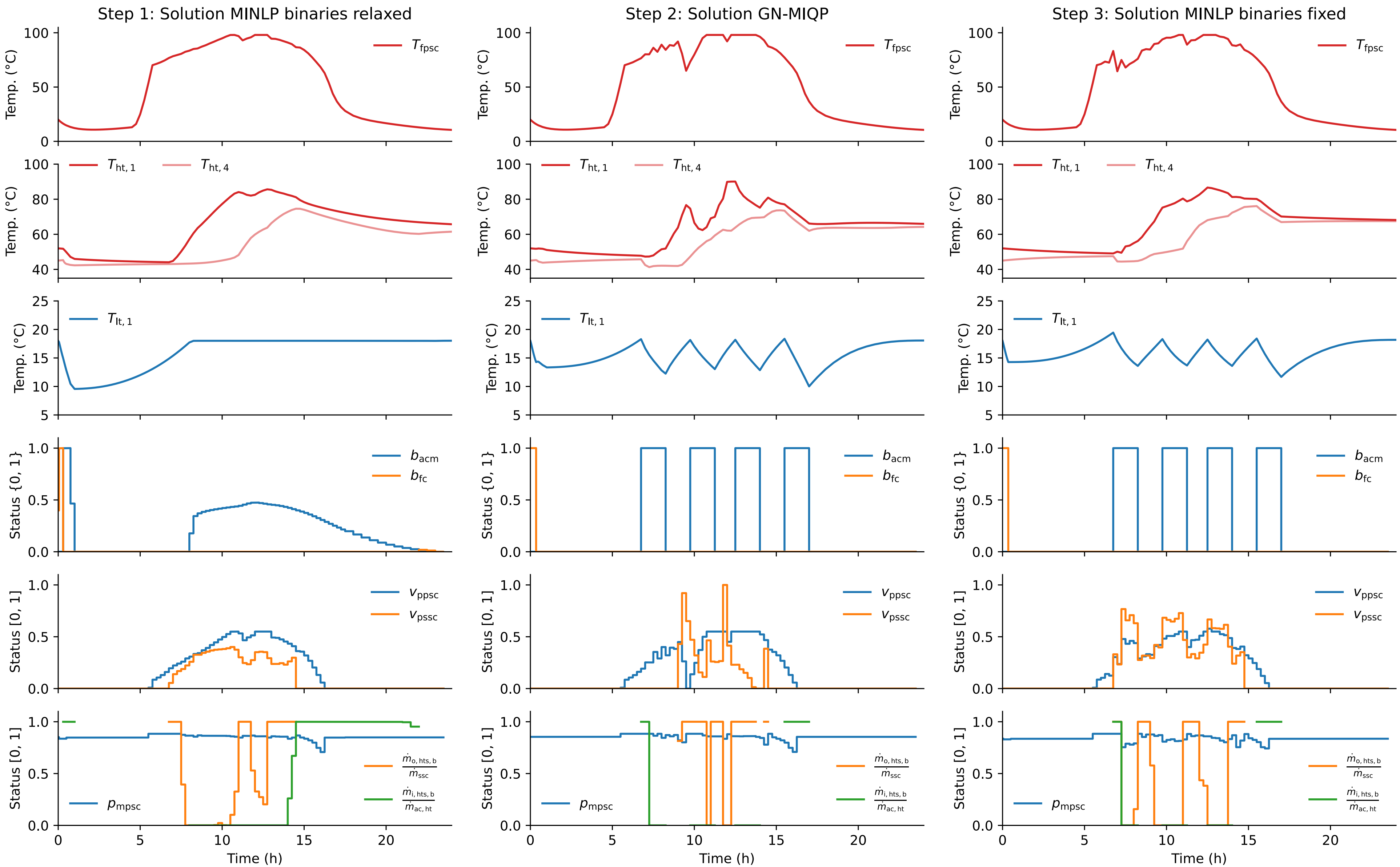

**Original MINLP**

$$\begin{aligned} \min_{y,z} \quad & \frac{1}{2}\|F_1(y,z)\|_2^2 + F_2(y,z) \\ \text{s.t.} \quad & G(y,z) = 0 \\ & H(y,z) \leq 0 \\ & y \in \mathbb{Z}^{n_y} \end{aligned}$$

**GN-MIQP from linearization at $(y^*, z^*)$**

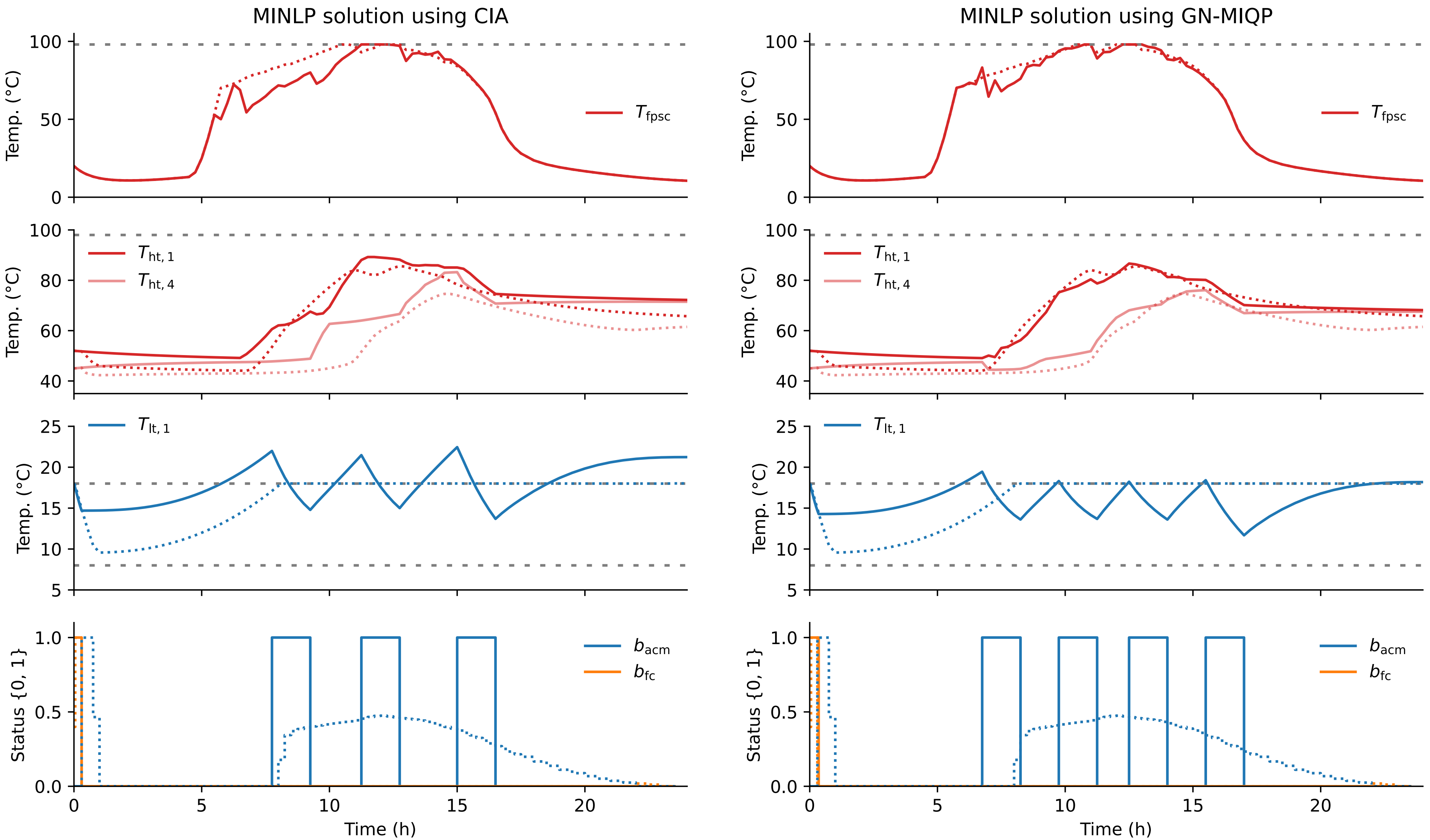
$$\begin{aligned} \min_{y,z} \quad & \frac{1}{2}\|F_{1,\mathrm{L}}(y,z;\bar{y},z^*)\|_2^2 + F_{2,\mathrm{L}}(y,z;y^*,z^*) \\ \text{s.t.} \quad & G_{\mathrm{L}}(y,z;y^*,z^*) = 0 \\ & H_{\mathrm{L}}(y,z;y^*,z^*) \leq 0 \\ & y \in \mathbb{Z}^{n_y} \end{aligned}$$

# Numerical results: Three Step GN-MIQP Decomposition




(25 CPU sec) (130 CPU sec) (15 CPU sec)

# Comparison of CIA and GN-MIQP Solution




(43 CPU sec) (170 CPU sec)

GN-MIQP delivers significant feasibility improvements, at the expense of increased computational cost.

# Summary and Recent Software Developments

- Exploiting convex structures in nonlinear problems is key for reliable and fast nonlinear MPC algorithms.
- Sequential Convex Programming (SCP) and its variants converge linearly. They avoid “bad” minimizers (where the nonlinearity dominates the convex substructure).
- Zero-Order Optimization allows us to design theoretically solid Iterative Learning Control algorithms. They can recover an optimal solution in special cases.
- Mixed Integer Optimal Control can be addressed by Three-Step-Decomposition method with classical CIA or novel Gauss-Newton MIQP variant
- Latest open-source (BSD 2) software developments from the team are:
  - BLASFEO: Basic Linear Algebra Subroutines For Embedded Optimization (Frison et al.), targeting dense matrices from 10x10 to 400x400
  - HPIPM: interior point QP and QCQP solver for block-sparse problems with optimal control and tree structure, based on BLASFEO (Frison et al., IFAC 2020, ECC 2022)
  - acados: Nonlinear MPC and MHE library implementing SCP type algorithms, using HPIPM and CasADi, with user interfaces from MATLAB and Python (Verschueren, Kouzoupis, Frison, Frey et al., successor of ACADO)
  - pycombina: fast solution of a special class of mixed integer linear programs arising in the combinatorial integral approximation (CIA) method for nonlinear mixed integer optimal control (Bürger et al., IFAC 2020)
  - NOS-NOC: Non-Smooth Systems Numerical Optimal Control package based on MATLAB, CasADi, IPOPT (Nurkanovic et al., CDC 2022, submitted)

Thank you