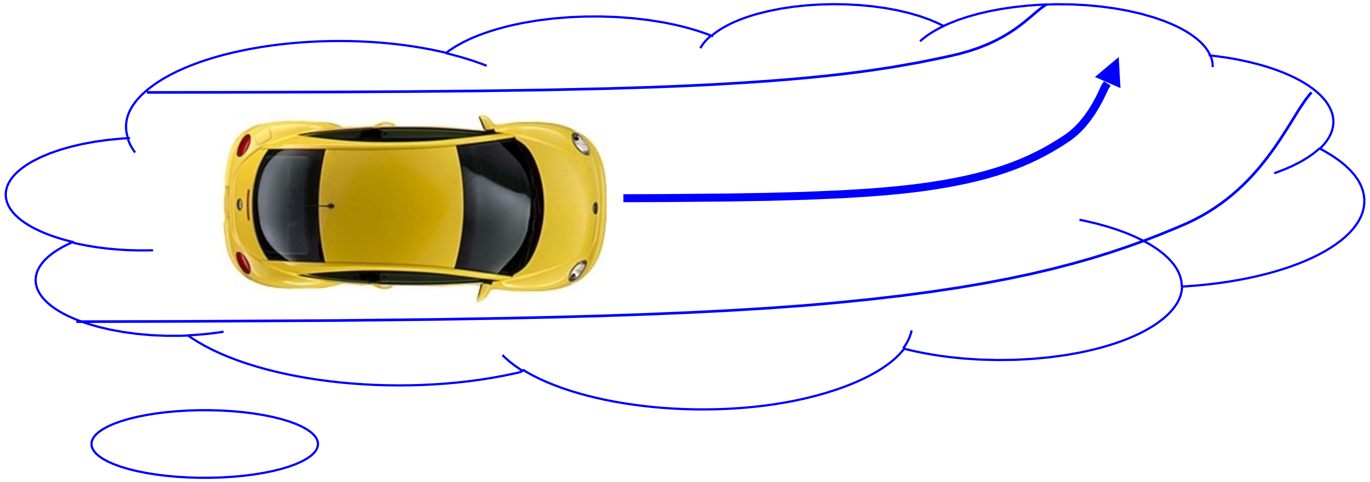
Real-Time Optimization for Nonlinear Model Predictive Control

Moritz Diehl

(First a Distillation NMPC Story for Warm-Up)

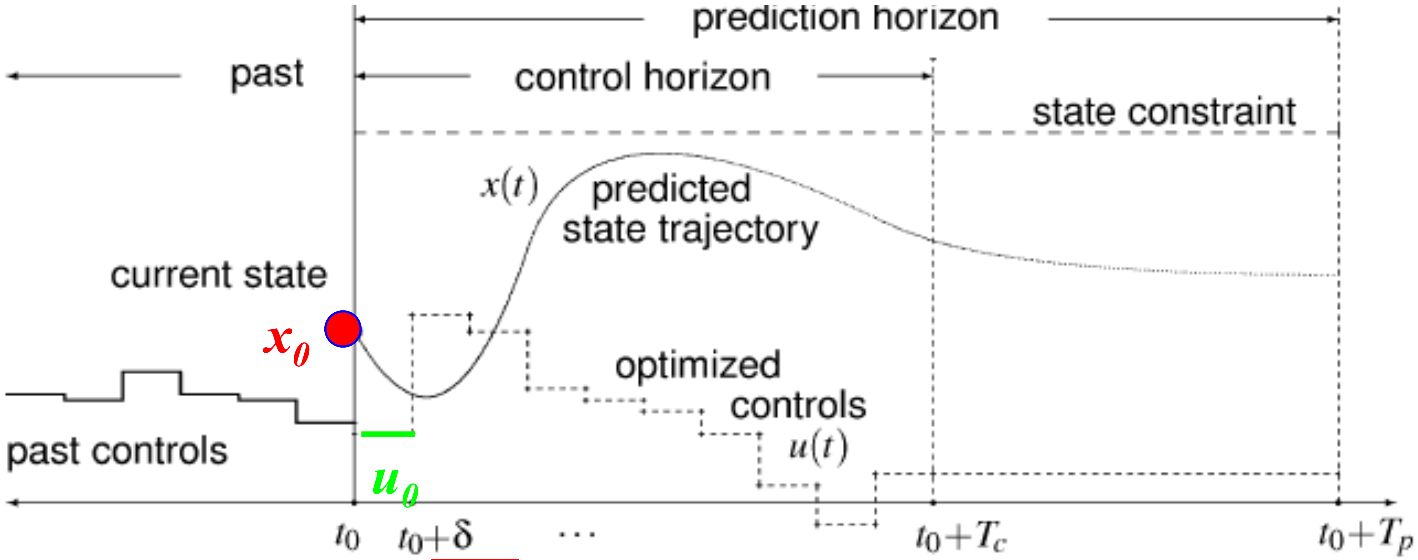
Model Predictive Control (MPC)

Always look a bit into the future.



Brain predicts and optimizes:
e.g. slow down **before** curve

Computations in Model Predictive Control (MPC)



1. Estimate current system state x_0 (and parameters) from measurements.

2. Solve *in real-time* an optimal control problem:

$$\min_{x,z,u} \int_{t_0}^{t_0+T_p} L(x,z,u)dt \; + E\left(x(t_0+T_p)\right) \; s.t. \; \begin{cases} x(t_0)-x_0 = 0, \\ \dot{x}-f(x,z,u) = 0, \; t \in [t_0, t_0+T_p] \\ g(x,z,u) = 0, \; t \in [t_0, t_0+T_p] \\ h(x,z,u) \geq 0, \; t \in [t_0, t_0+T_p] \\ r\left(x(t_0+T_p)\right) \geq 0. \end{cases}$$

3. Implement first control u_0 for time δ at real plant. Set $t_0 = t_0 + \delta$ and go to 1.

Main challenge for MPC: fast and reliable real-time optimization

Example: Distillation Column (ISR, Stuttgart)

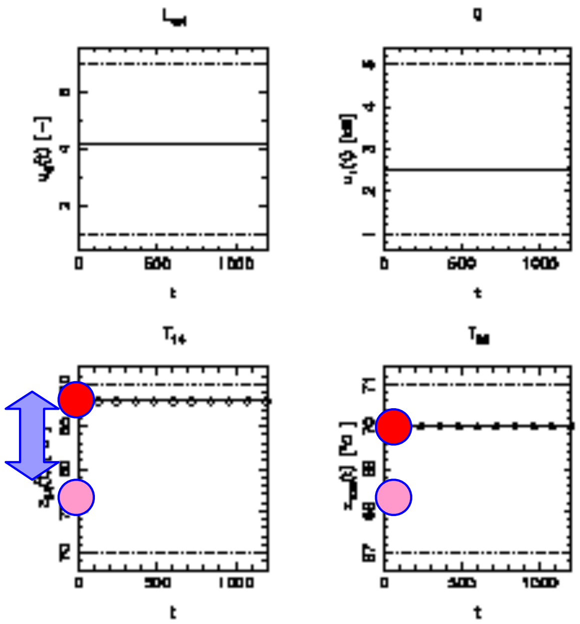
- Aim: to ensure product purity, keep two temperatures (T_{14}, T_{28}) constant despite disturbances
- least squares objective:

$$\min \int_{t_0}^{t_0+T_p} \left\| \begin{array}{c} T_{14}(t) - T_{14}^{\mathrm{ref}} \\ T_{28}(t) - T_{28}^{\mathrm{ref}} \end{array} \right\|_2^2 dt$$

- control horizon 10 min
- prediction horizon 10 h
- stiff DAE model with 82 differential and 122 algebraic state variables
- Desired sampling time: 30 seconds.

Distillation Online Scenario

- System is in steady state, optimizer predicts constant trajectory:

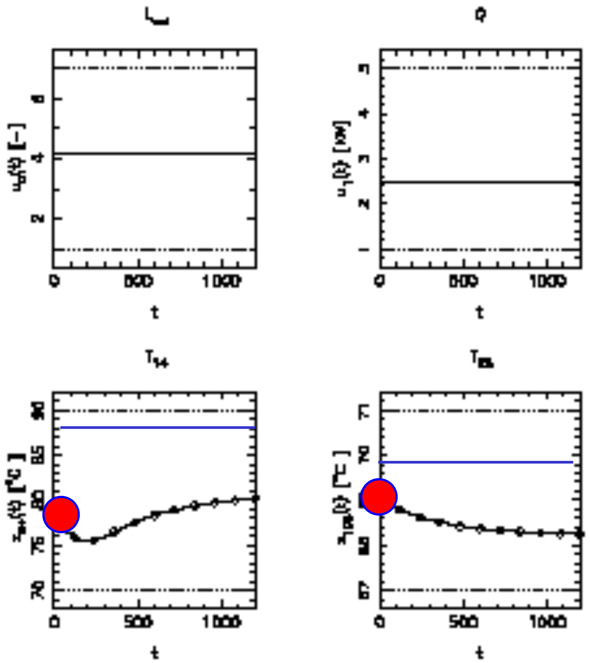


- **Suddenly**, system state x_0 is disturbed.
- What to do with optimizer?

Conventional Approach

- use offline method, e.g. MUSCOD-II with BFGS (Leineweber, 1999).
- initialize with **new** initial value x_0 and integrate system with **old** controls.
- iterate until convergence.

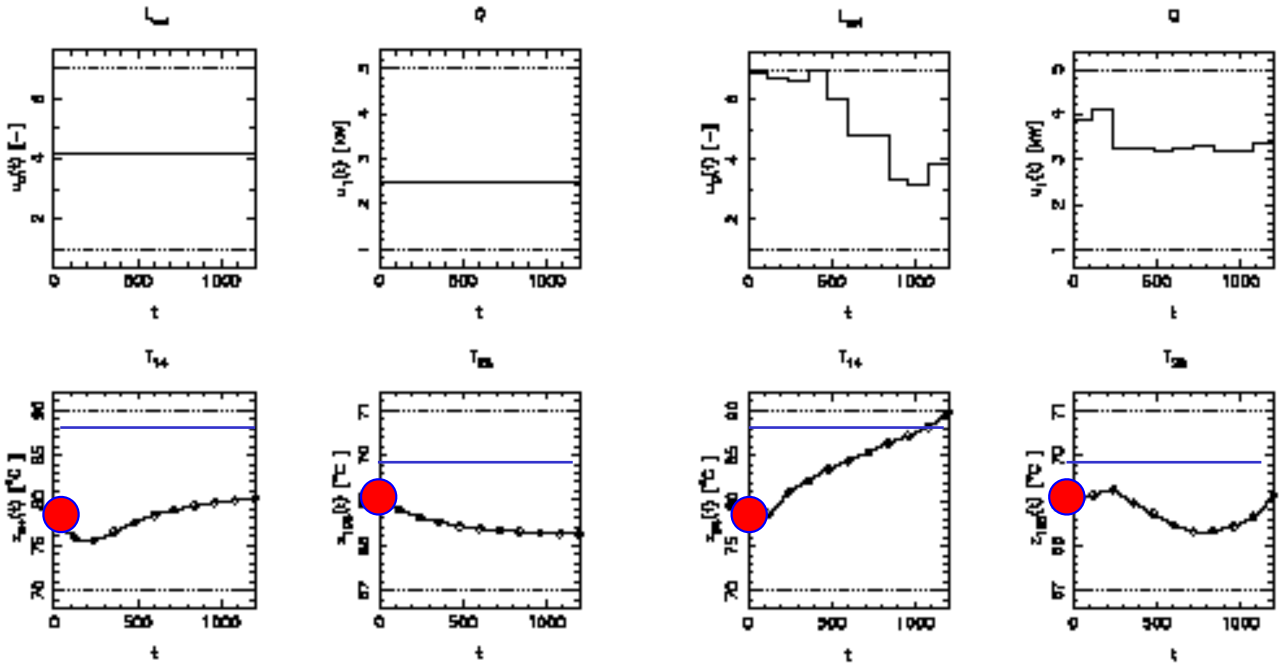
Initialization



Conventional Approach

- use offline method, e.g. MUSCOD-II with BFGS (Leineweber, 1999).
- initialize with **new** initial value x_0 and integrate system with **old** controls.
- iterate until convergence.

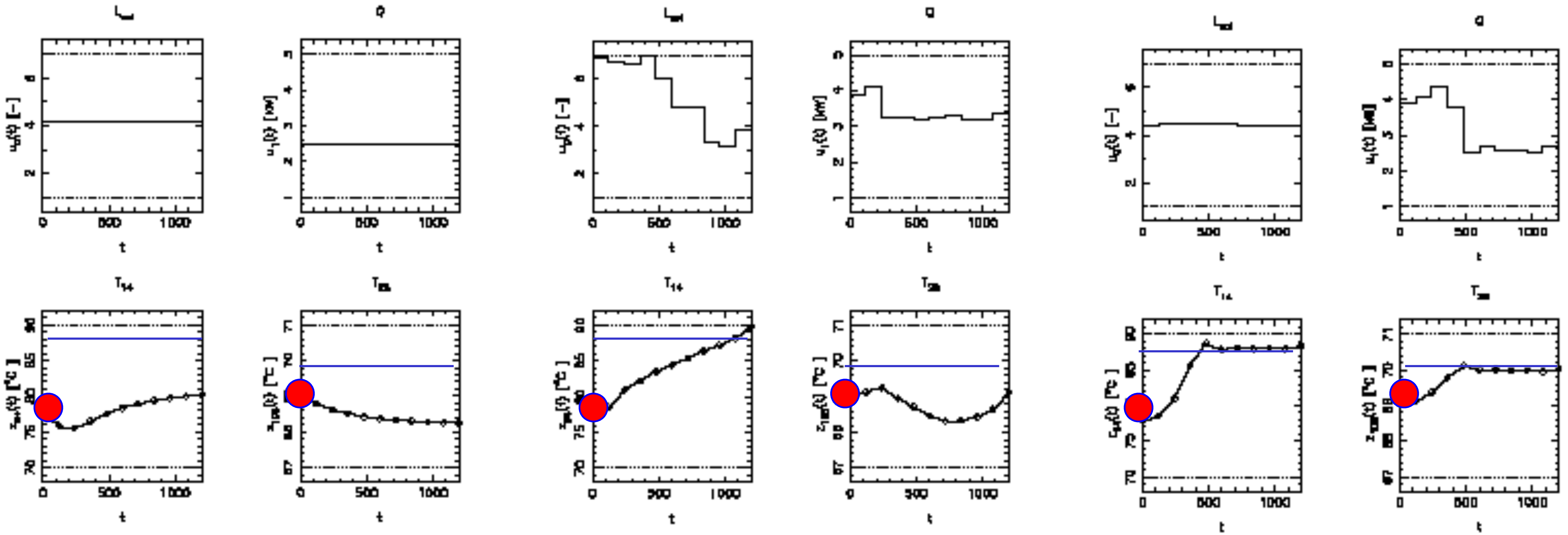
Initialization 16th Iteration



Conventional Approach

- use offline method, e.g. MUSCOD-II with BFGS (Leineweber, 1999).
- initialize with **new** initial value x_0 and integrate system with **old** controls.
- iterate until convergence.

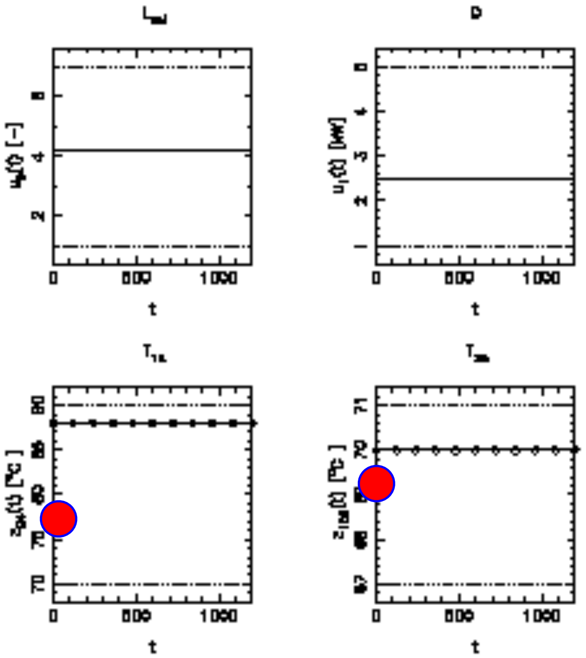
Initialization 16th Iteration Solution (32nd Iteration



New Approach: Initial Value Embedding

- Initialize with **old** trajectory, accept violation of $s_0^x - x_0 = 0$

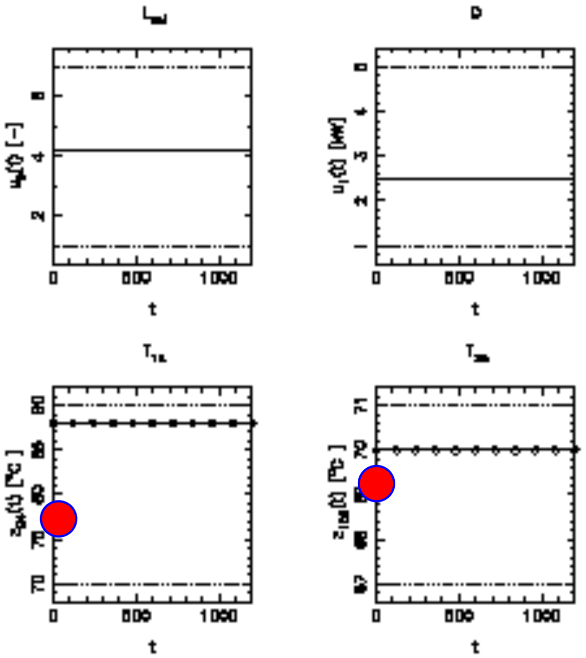
Initialization



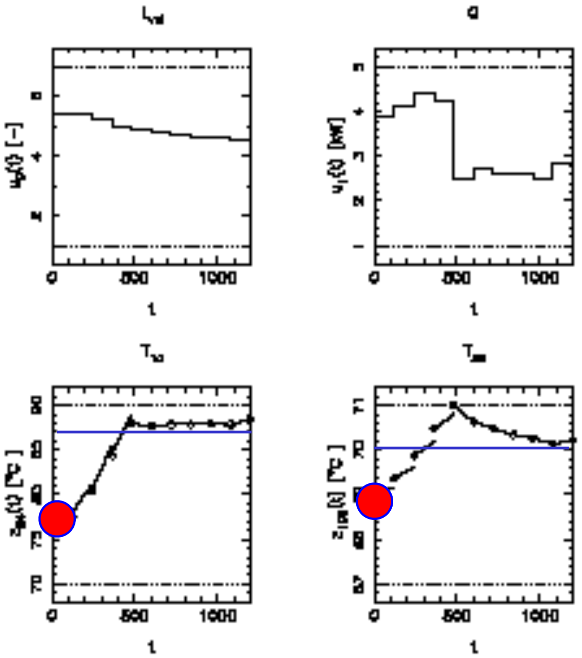
New Approach: Initial Value Embedding

- Initialize with **old** trajectory, accept violation of $s_0^x - \boxed{x_0} = 0$

Initialization



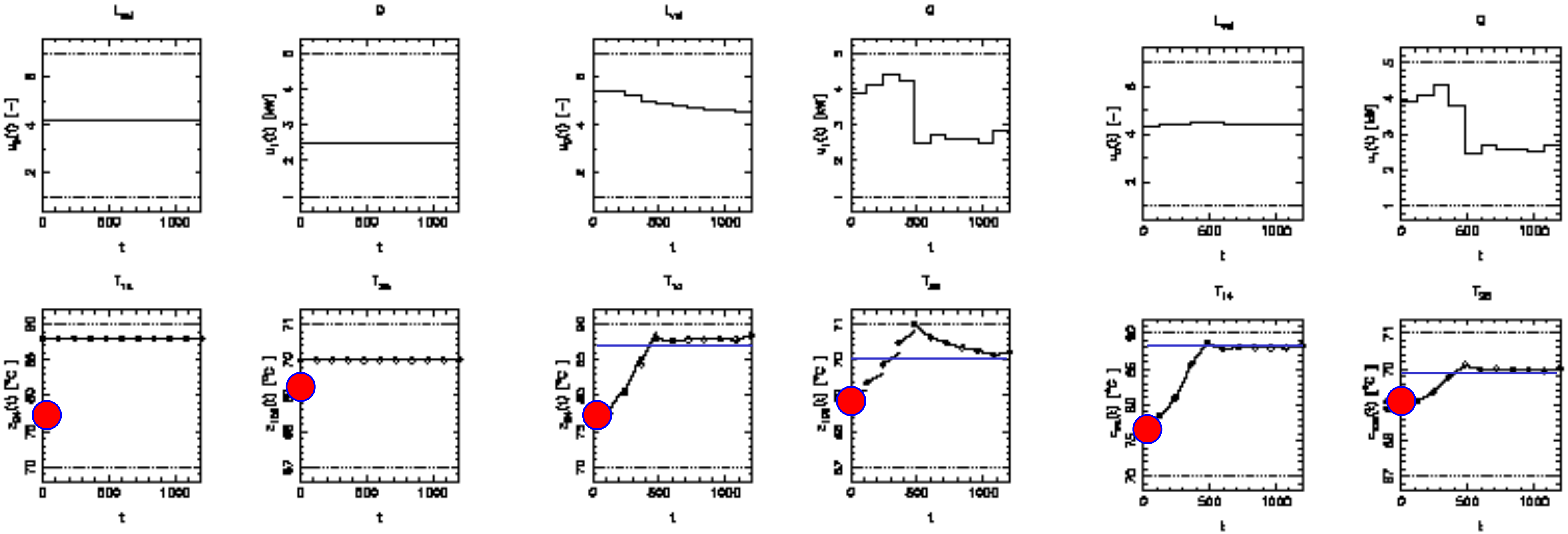
First Iteration



New Approach: Initial Value Embedding

- Initialize with **old** trajectory, accept violation of $s_0^x - \boxed{x_0} = 0$

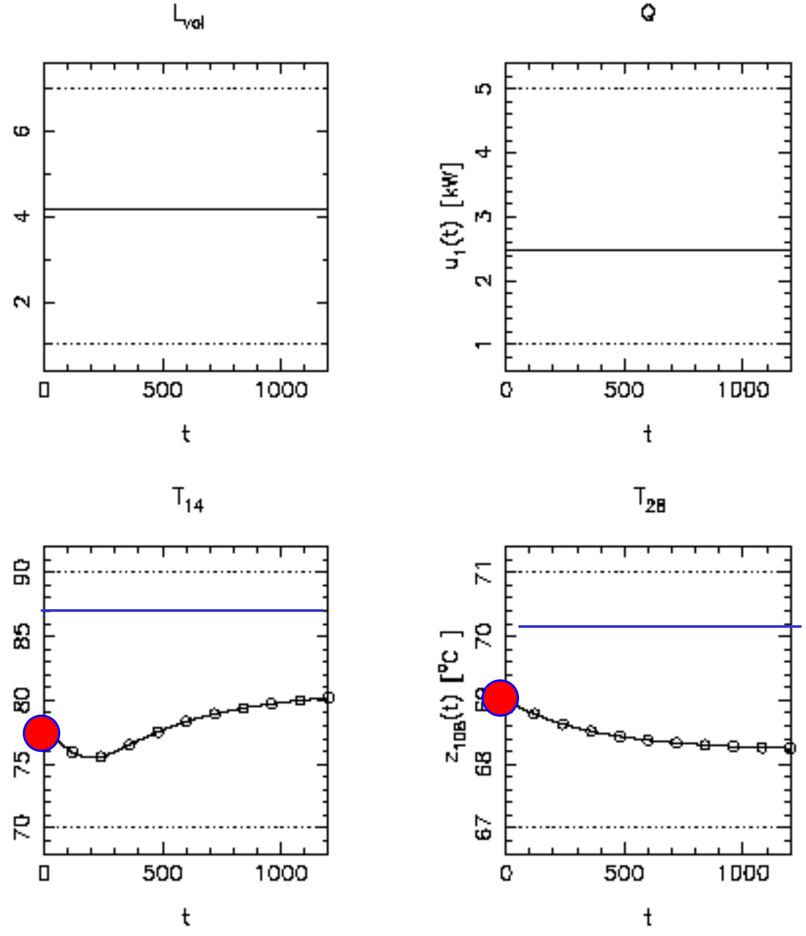
Initialization | First Iteration | Solution (3rd Iteration)



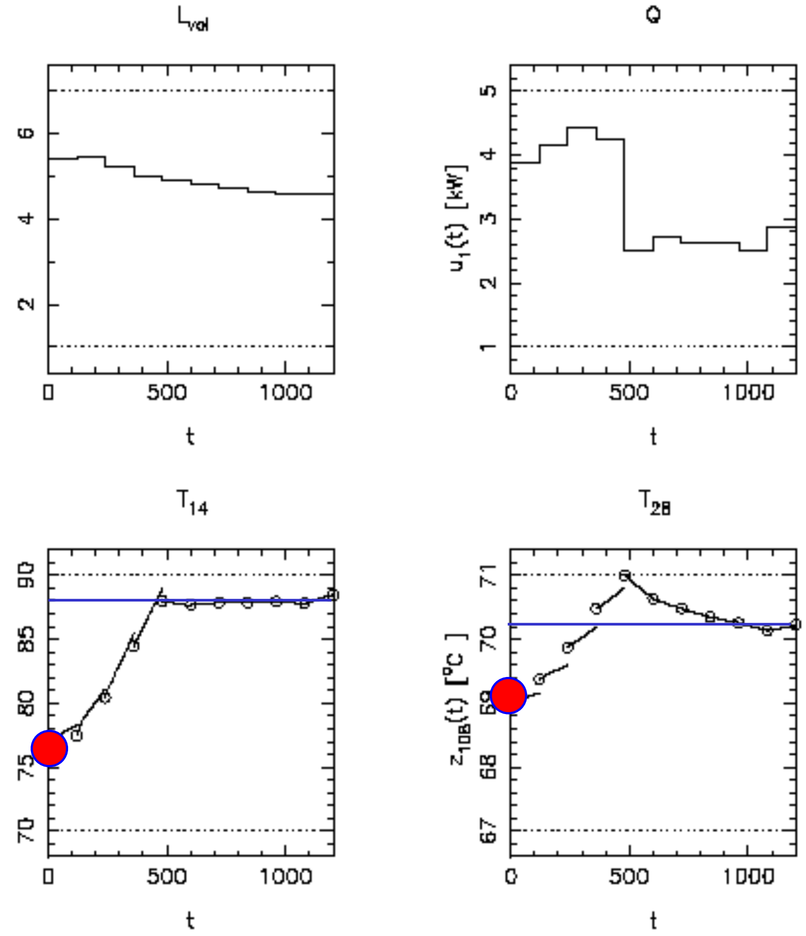
First iteration nearly solution!

Very different results after first iteration!

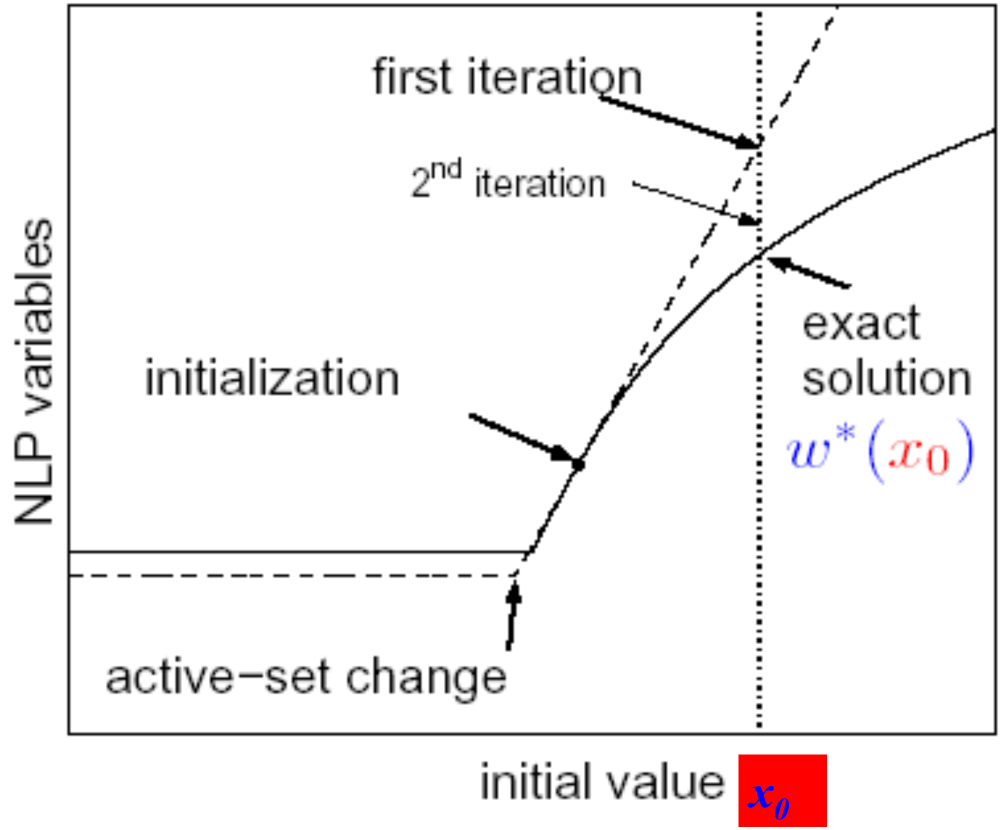
Conventional:



Initial Value Embedding:

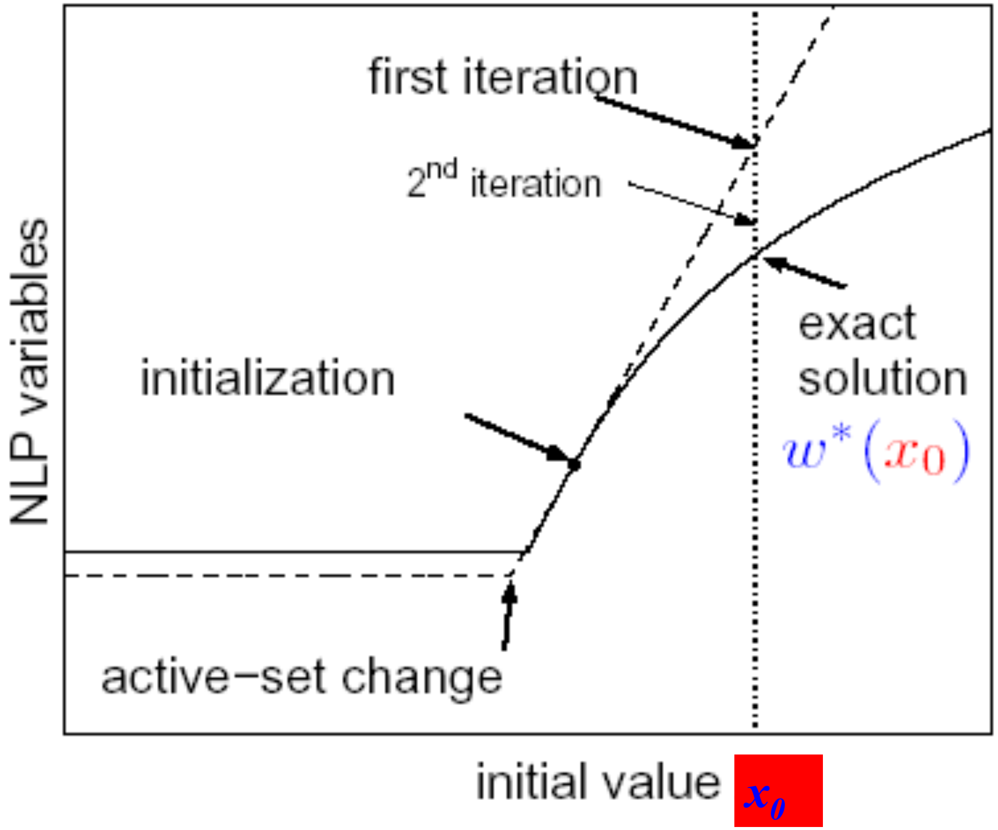


Initial Value Embedding



- first iteration is tangential predictor for exact solution (for exact hessian SQP)
- also valid for active set changes
- derivative can be computed *before* x_0 is known: first iteration nearly without delay

Initial Value Embedding

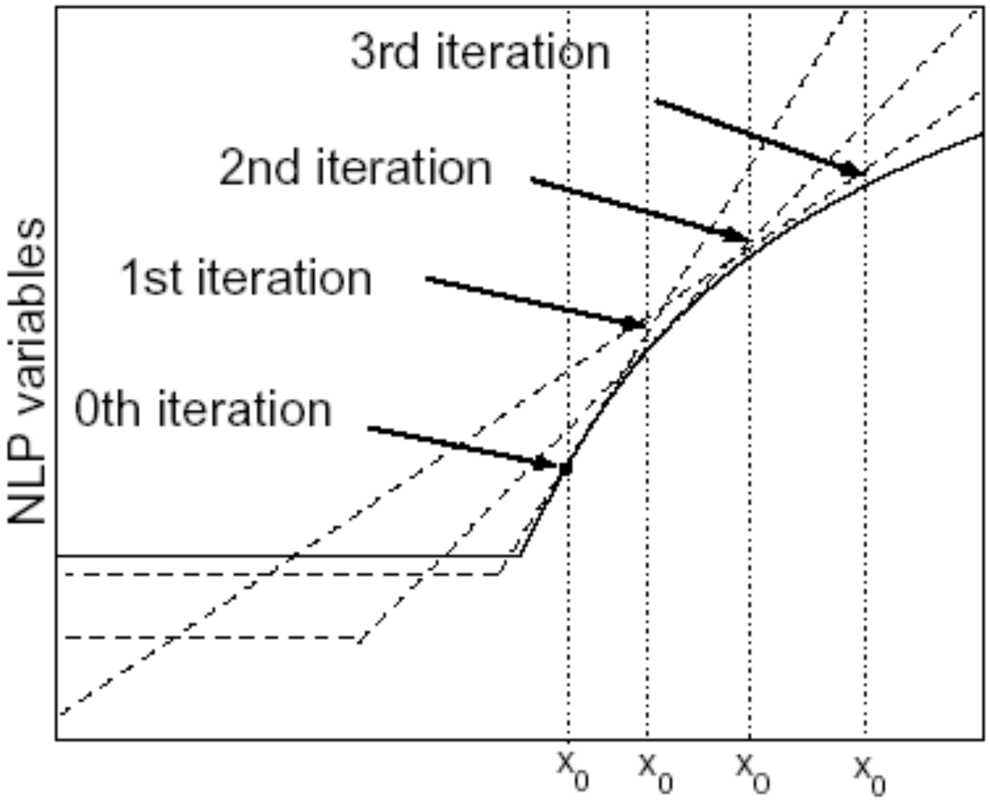


- first iteration is tangential predictor for exact solution (for exact hessian SQP)
- also valid for active set changes
- derivative can be computed *before* x_0 is known: first iteration nearly without delay

Why wait until convergence and do nothing in the meantime?

Real-Time Iterations [D. 2001]

Iterate, *while* problem is changing!



- tangential prediction after each change in x_0
- solution accuracy is increased with each iteration when x_0 changes little
- iterates stay close to solution manifold

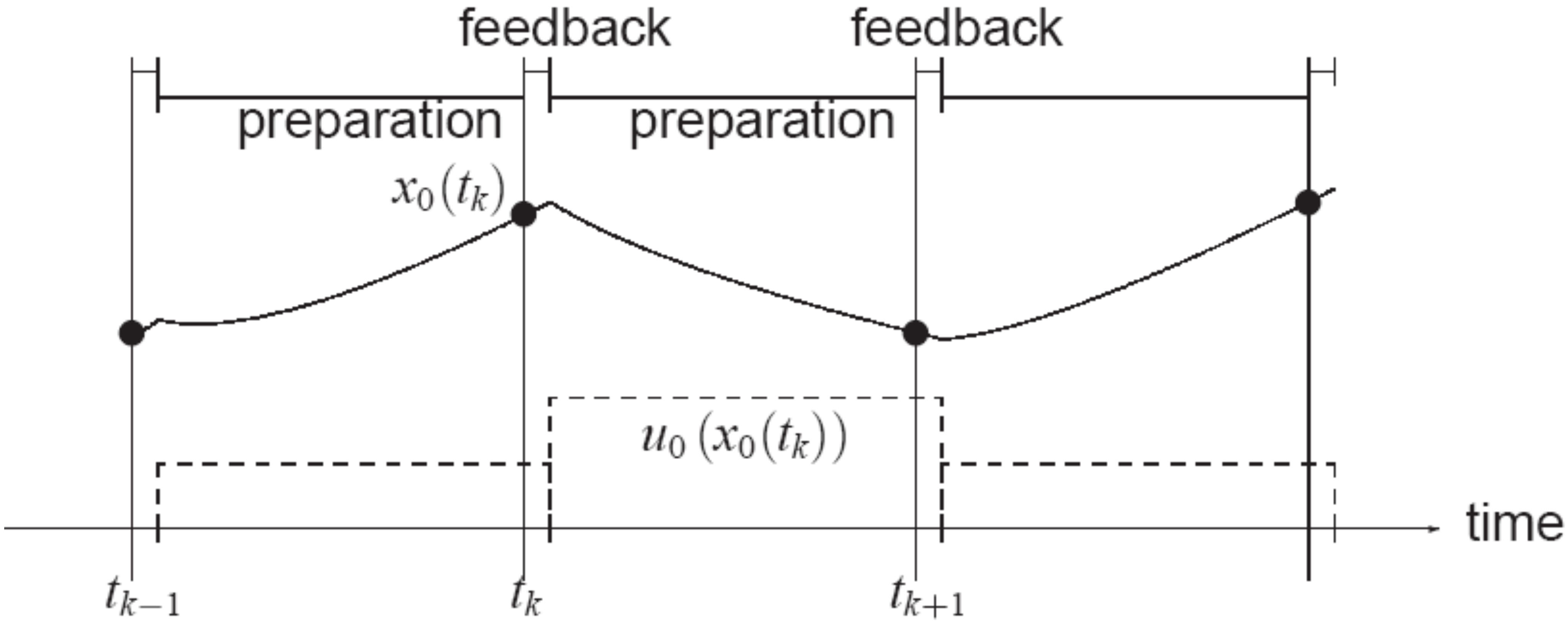
Real-Time Iteration Algorithm:

1. **Preparation Step (costly):**
   Linearize system at current iterate, perform partial reduction and condensing of quadratic program.

2. **Feedback Step (short):**
   When new x_0 is known, solve condensed QP and implement control u_0 immediately.
   Complete SQP iteration. Go to 1.

- short cycle-duration (as **one** SQP iteration)
- negligible feedback delay ($\approx$ 1 % of cycle)
- nevertheless fully nonlinear optimization

Real-Time Iterations minimize feedback delay

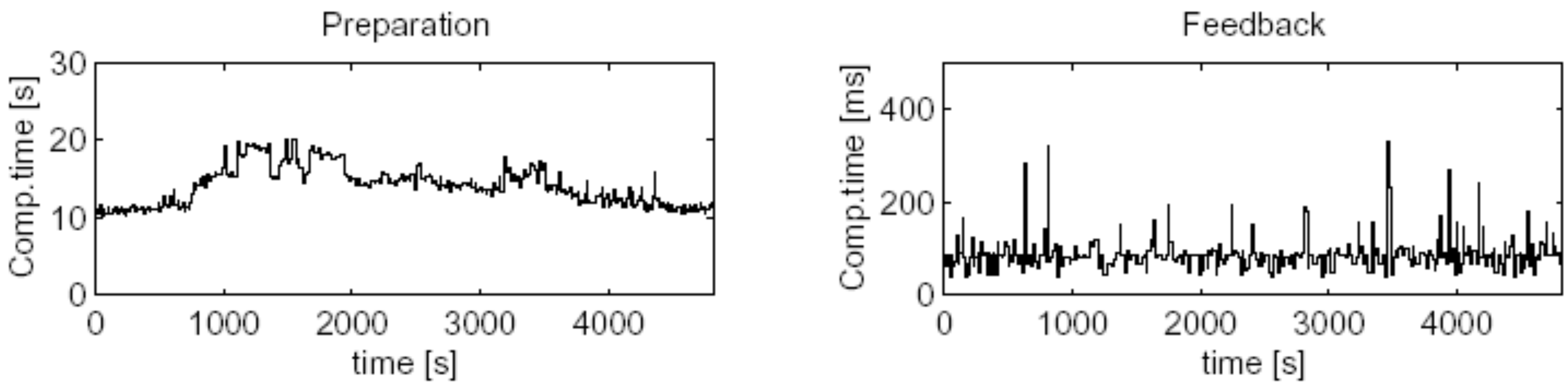


Realization at Distillation Column

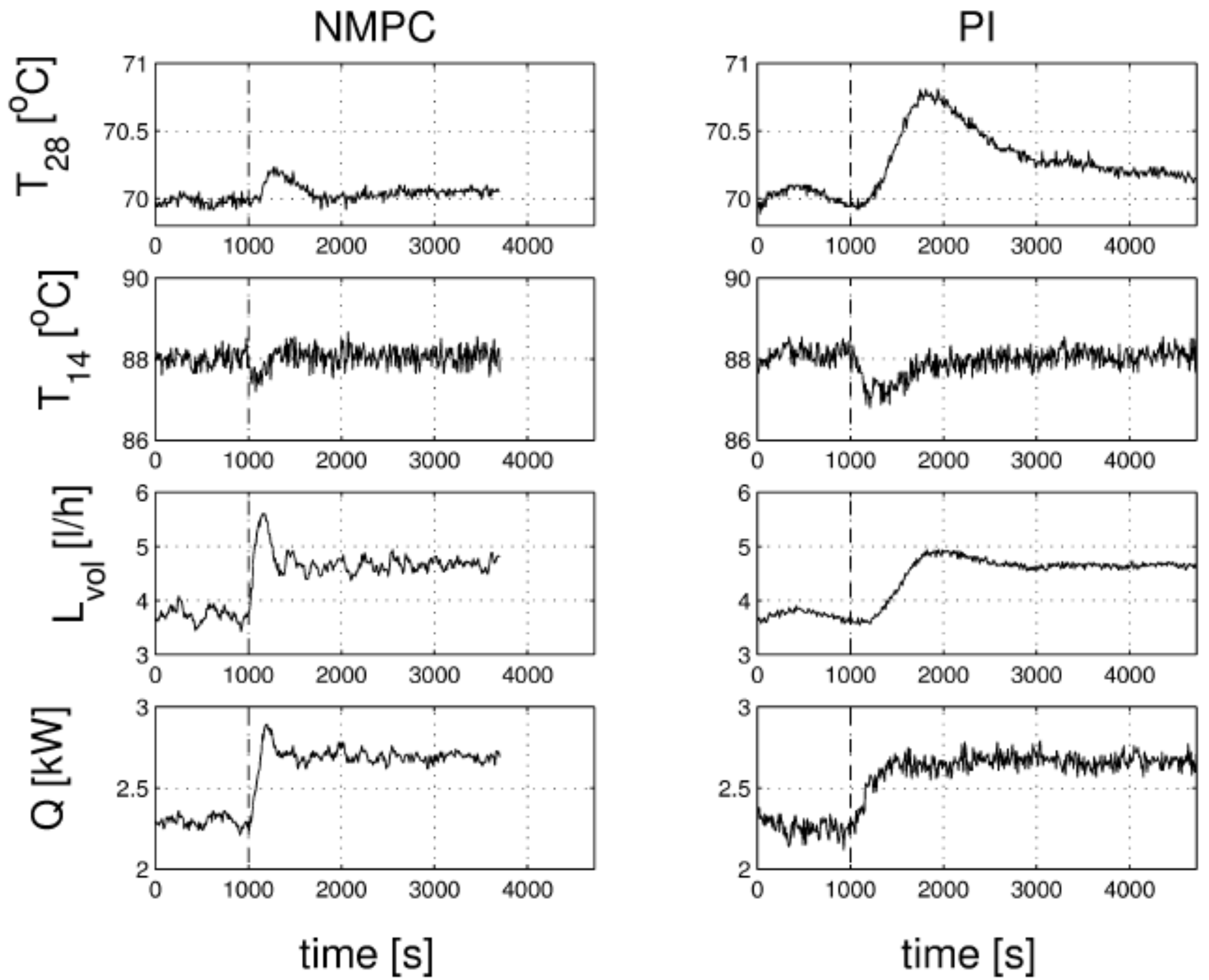
[D., Findeisen, Schwarzkopf, Uslu, Allgöwer, Bock, Schlöder,2002]

- Parameter estimation using dynamic experiments
- Online state estimation with Extended Kalman Filter variant, using only 3 temperature measurements to infer all 82 system states
- Implementation of estimator and optimizer on Linux Workstation.
- Communication with Process Control System via FTP all 10 seconds.
- Self-synchronizing processes.

Computation Times During Application

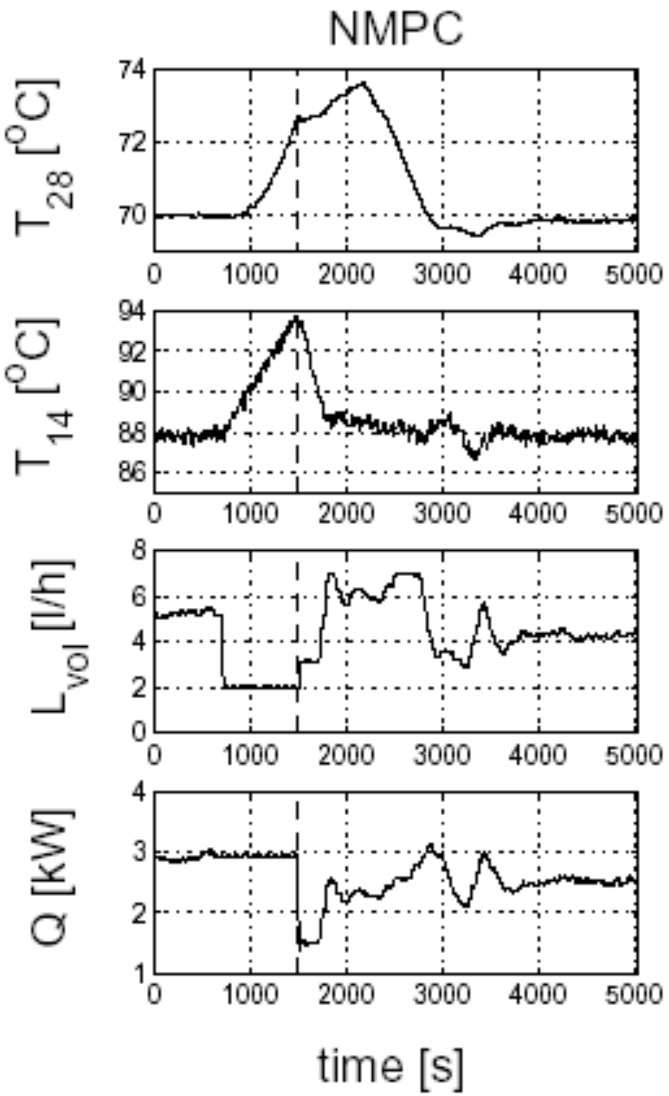


Feedflow Change by 20%: Transient Phase (Comparison with PI-Controller)



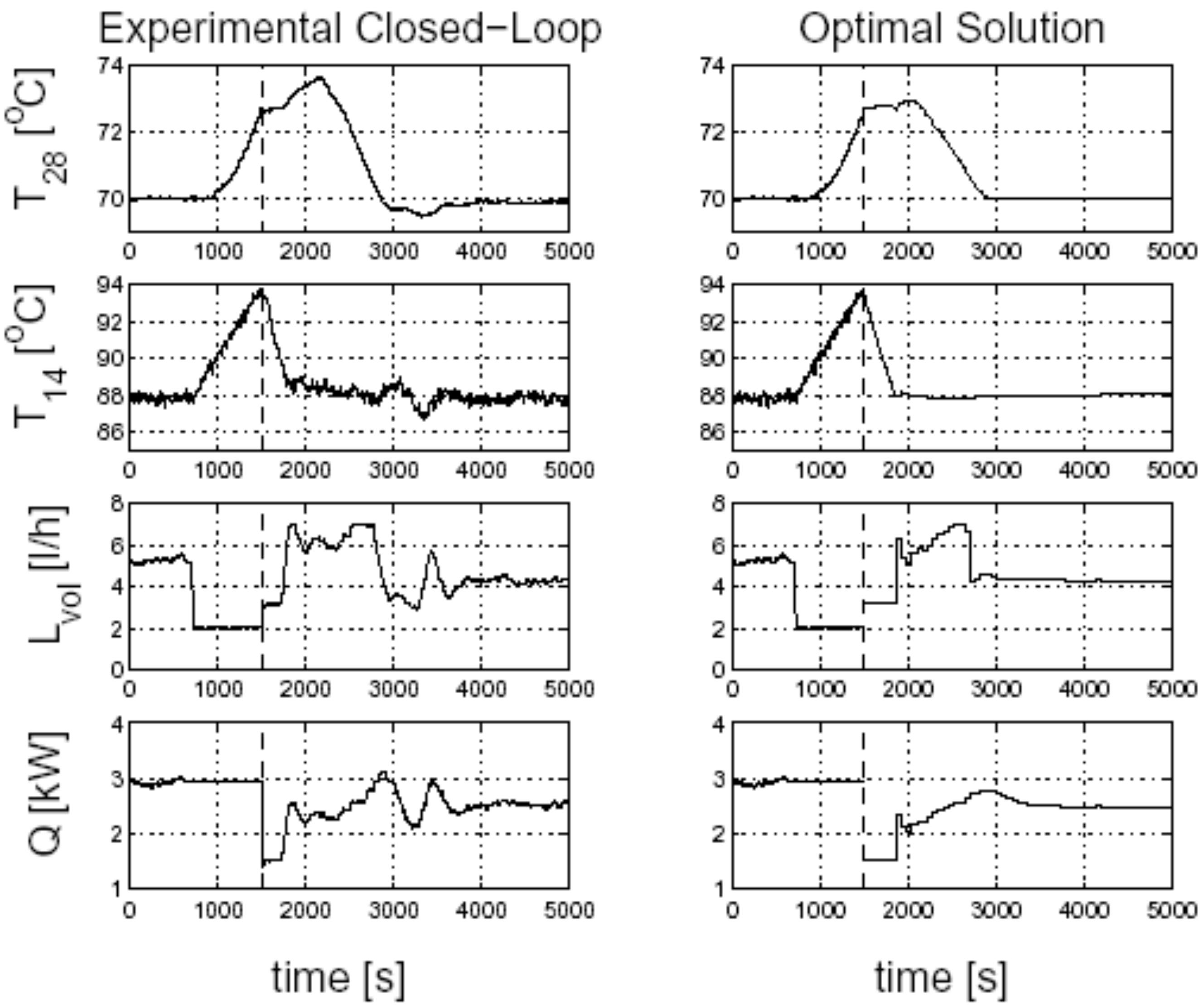
Transient in 15 minutes instead of 2 hours!

Large Disturbance (Heating), then NMPC



- Overheating by manual control
- NMPC only starts at $t = 1500$ s
- PI-controller not implementable, as disturbance too large (valve saturation)
- NMPC: at start control bound active $\Rightarrow T_{28}$ rises further
- Disturbance attenuated after half an hour

Real vs. Theoretical Optimal Solution



(Now back to the history of NMPC)

Outline of the Talk

PART I: Offline Optimal Control

- NMPC and MHE Problem Statement
- Simultaneous vs. Sequential Formulation
- Newton Type Optimization: IP vs. SQP Methods

PART II: Online Algorithms

- Parametric Sensitivities
- Review of Three Classical Algorithms

NMPC Optimal Control Problem in Continuous Time

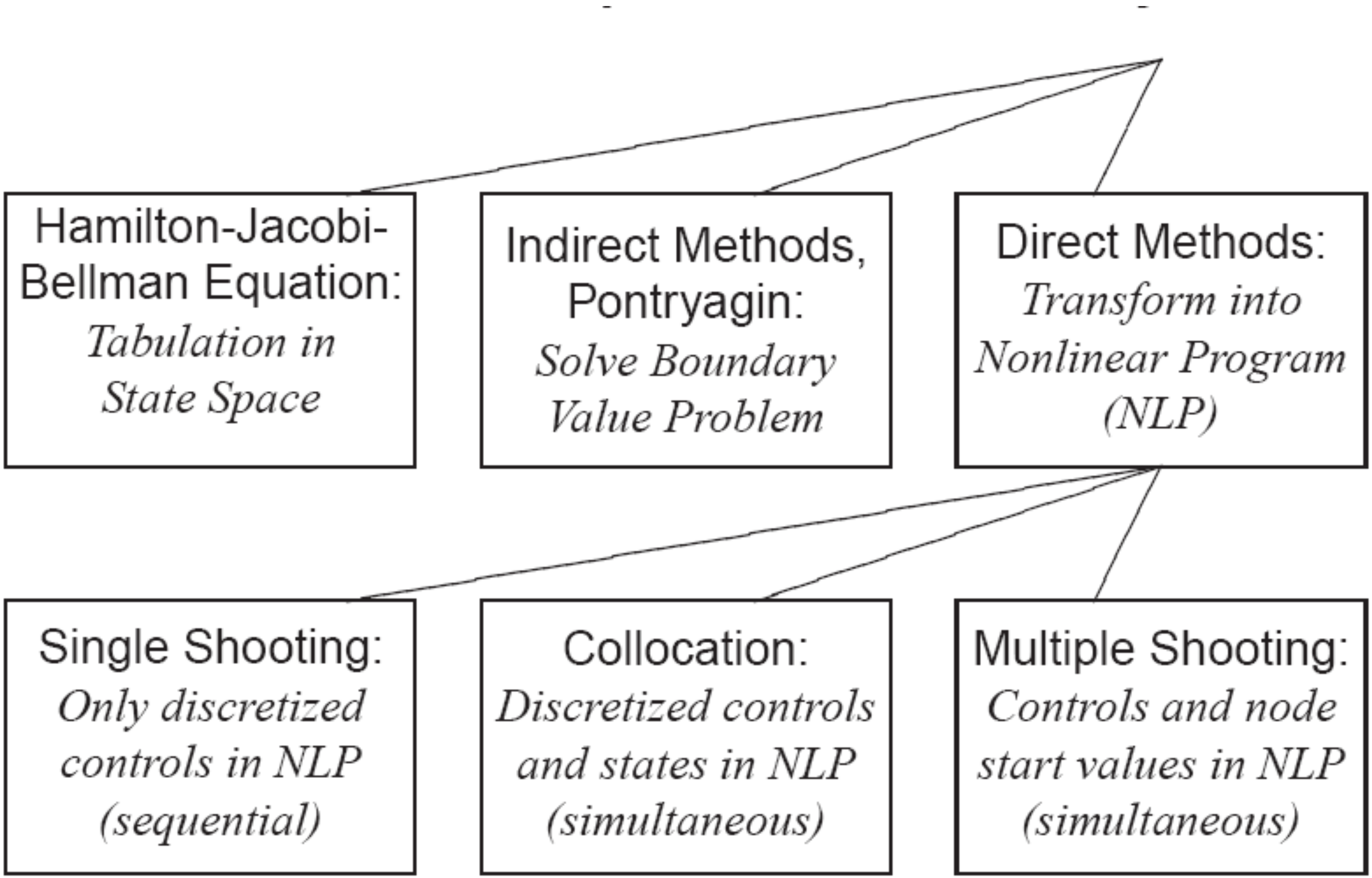
$$\underset{x(\cdot),\, u(\cdot)}{\text{minimize}} \quad \int_0^T L(x(t), u(t))\, dt \;+\; E\,(x(T))$$

subject to

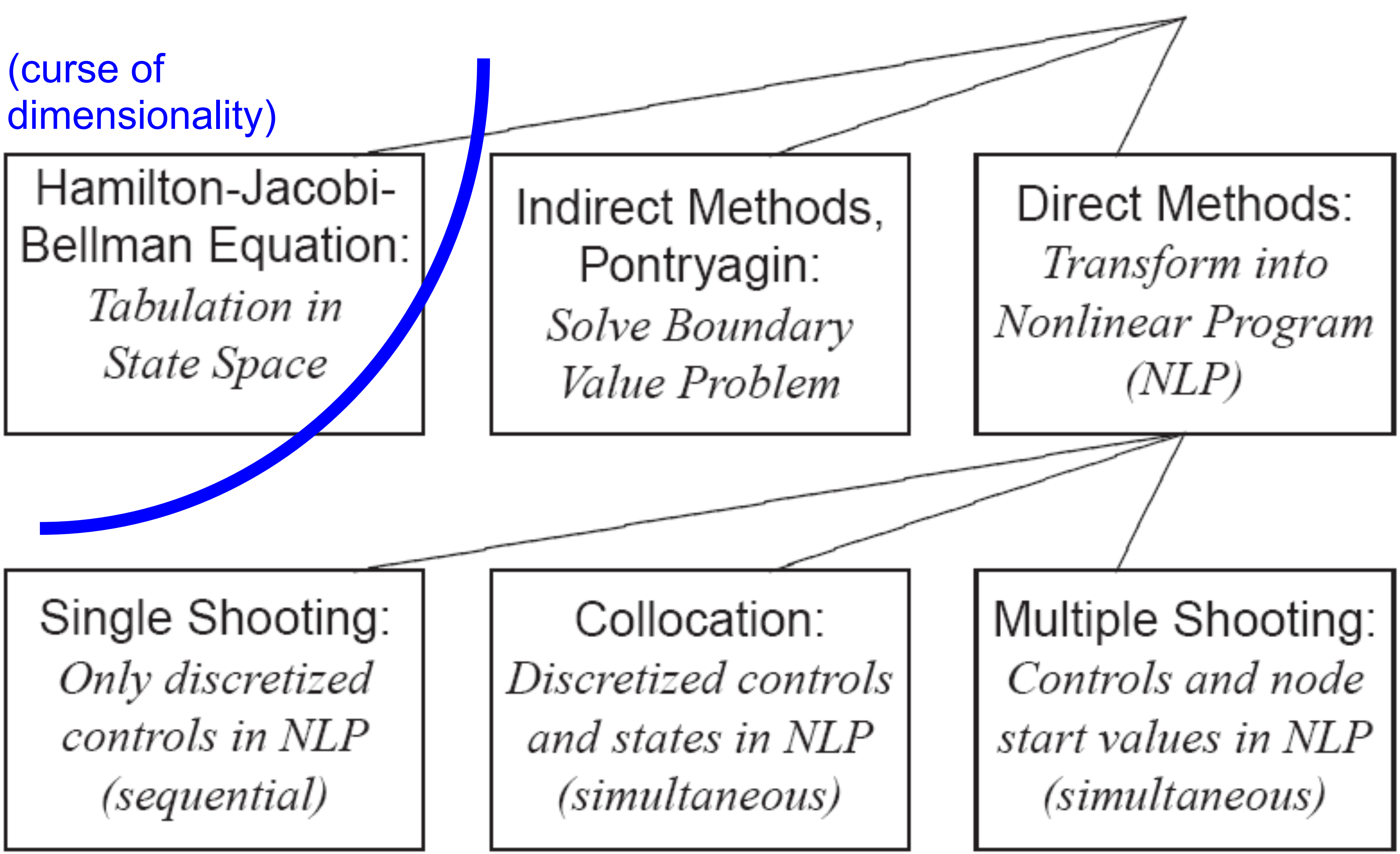
$$\begin{aligned}
x(0) - x_0 &= 0, && && \text{(fixed initial value)}\\
\dot{x}(t) - f(x(t), u(t)) &= 0, && t \in [0, T], && \text{(ODE model)}\\
h(x(t), u(t)) &\geq 0, && t \in [0, T], && \text{(path constraints)}\\
r\,(x(T)) &= 0 && && \text{(terminal constraints).}
\end{aligned}$$

How to solve these nonlinear problems reliably and fast?

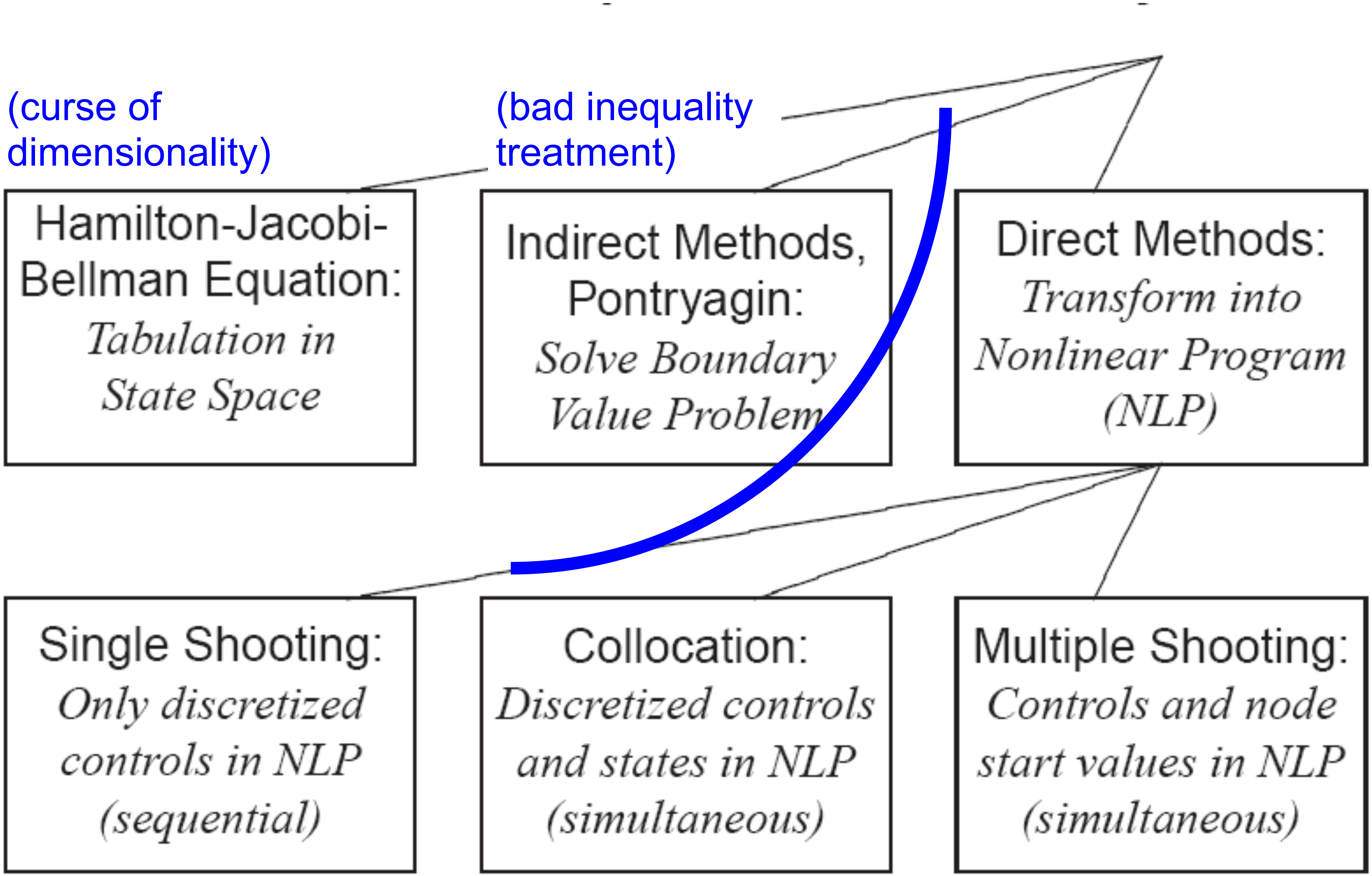
Optimal Control Family Tree



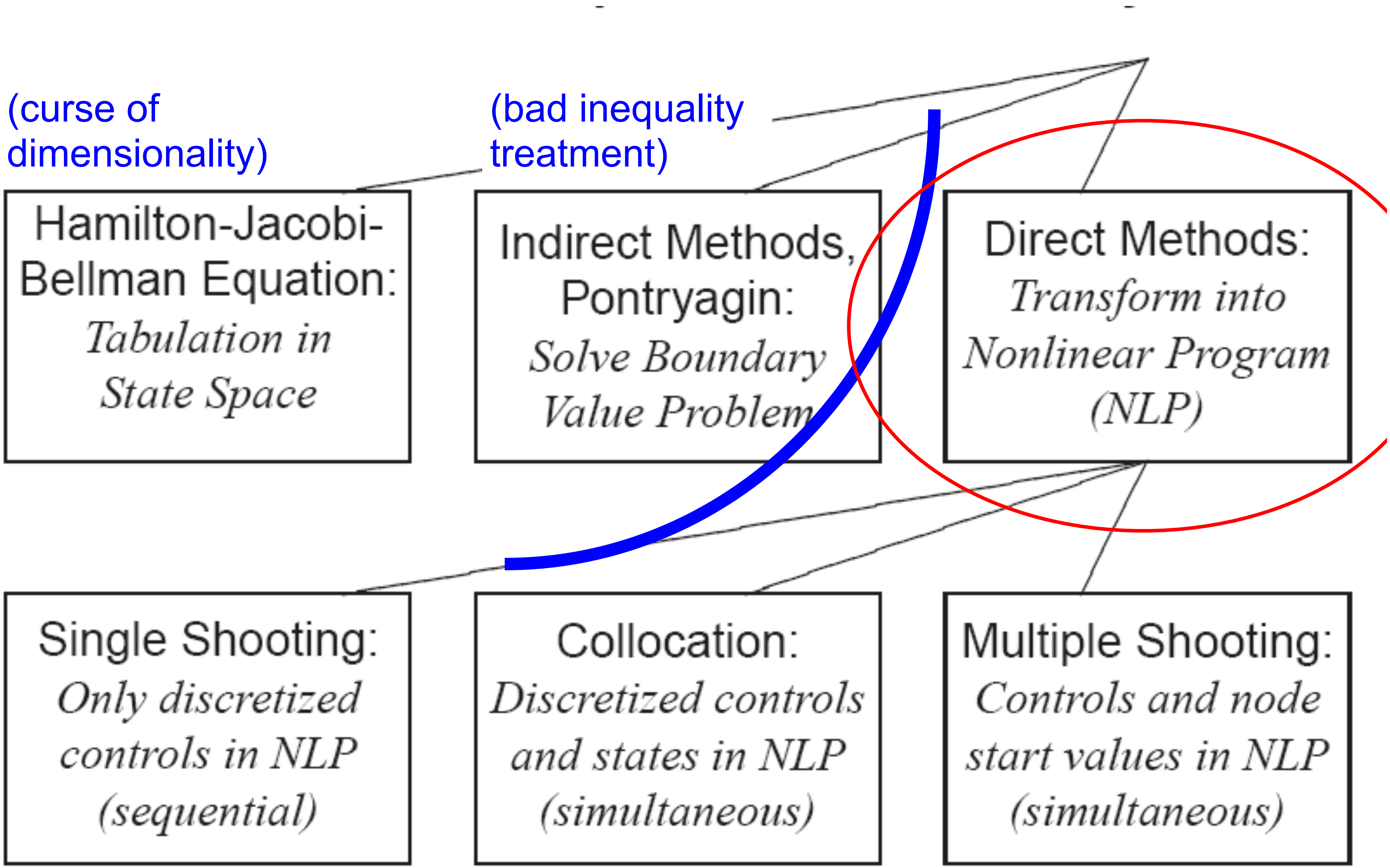
Optimal Control Family Tree



Optimal Control Family Tree



Optimal Control Family Tree



NMPC Problem in Discrete Time

$$
\begin{array}{lrcll}
\underset{x,\, z,\, u}{\text{minimize}} & \displaystyle\sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) & + & E\left(x_N\right) & \\
\text{subject to} & x_0 - \bar{x}_0 & = & 0, & \\
& x_{i+1} - f_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
& g_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
& h_i(x_i, z_i, u_i) & \leq & 0, & i = 0, \ldots, N-1, \\
& r\left(x_N\right) & \leq & 0. &
\end{array}
$$

Structured parametric Nonlinear Program, “mp-NLP”

- Initial Value $\bar{x}_0$ is not known beforehand (“online data”)
- Discrete time dynamics often come from ODE simulation (“shooting”)
- “Algebraic States” z implicitly defined via third condition, can come from DAEs or from collocation discretization

[Moving Horizon Estimation (MHE) Problem]

$$\underset{x,\,z,\,w}{\text{minimize}} \quad \|x_0 - \bar{x}_0\|_P^2 + \sum_{i=0}^{N-1} \|y_i - m_i(x_i, z_i, u_i, w_i)\|_Q^2 + \|w_i\|_R^2$$

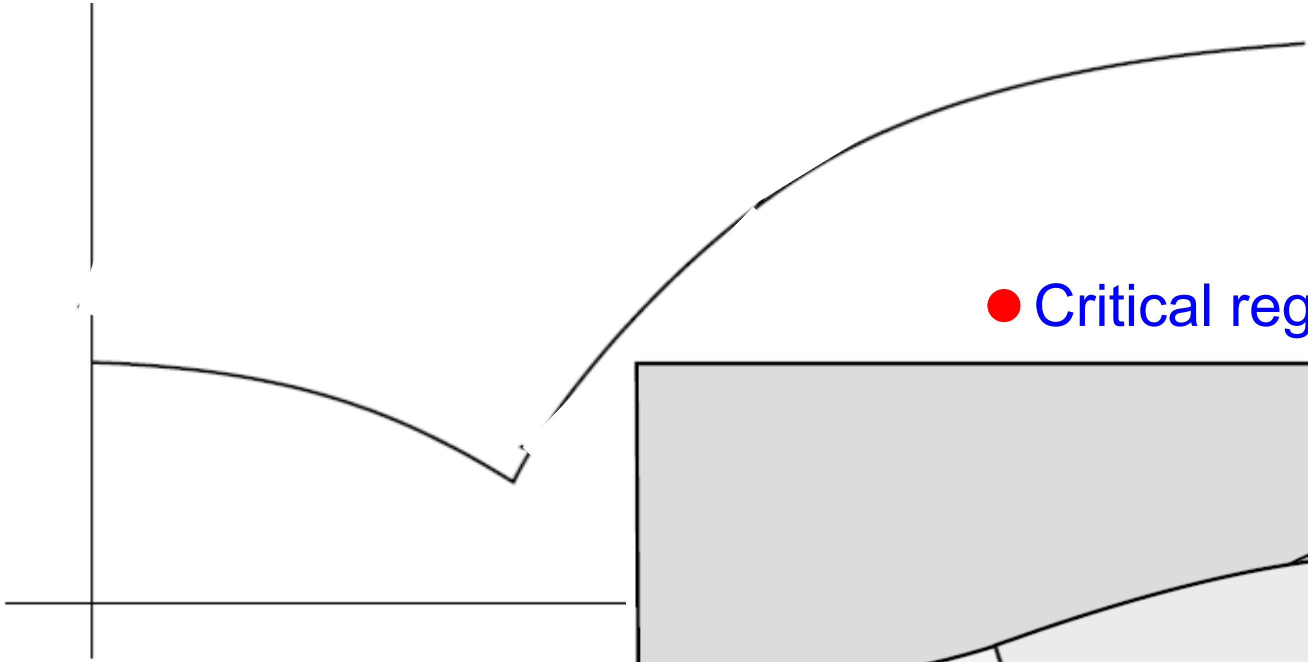
$$\text{subject to}$$

$$\begin{aligned}
x_{i+1} - f_i(x_i, z_i, u_i, w_i) &= 0, \quad i = 0, \ldots, N-1, \\
g_i(x_i, z_i, u_i, w_i) &= 0, \quad i = 0, \ldots, N-1, \\
h_i(x_i, z_i, u_i, w_i) &\leq 0, \quad i = 0, \ldots, N-1,
\end{aligned}$$

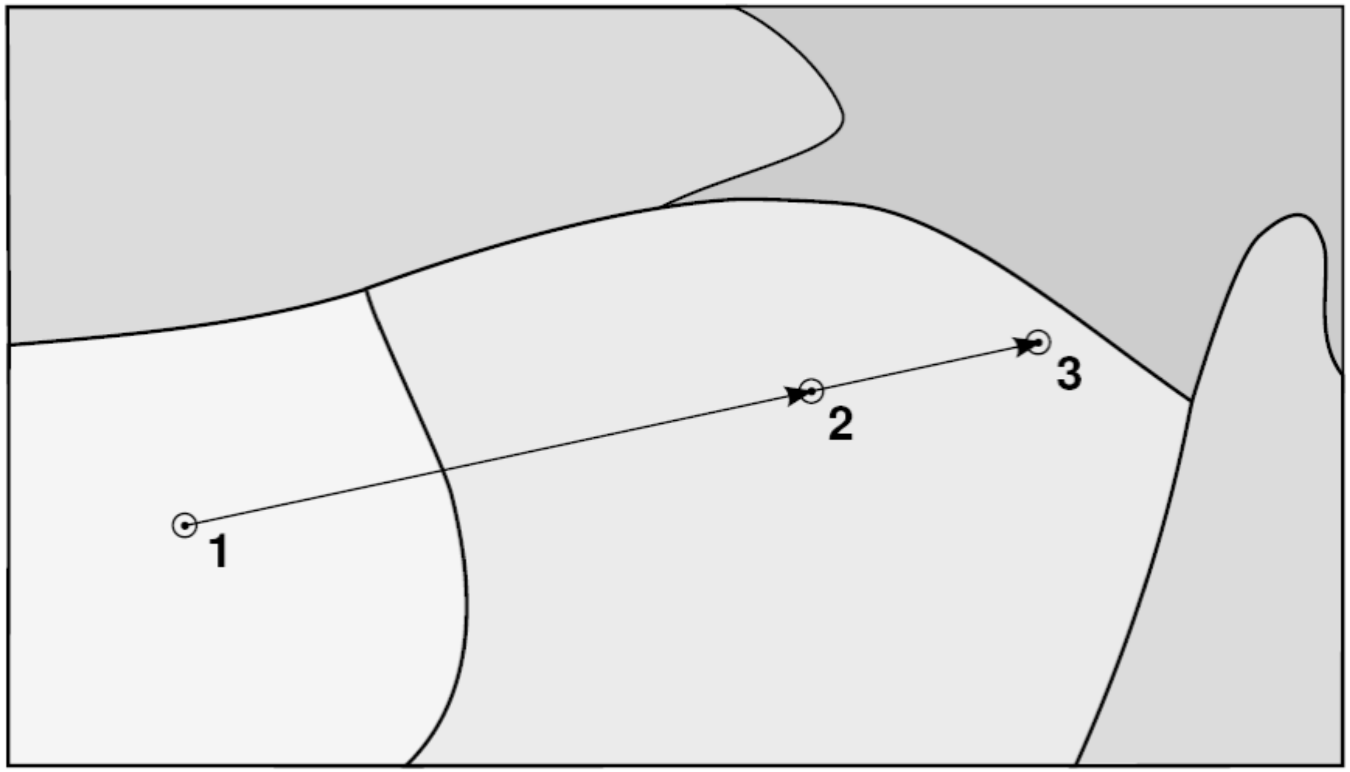
- Online problem data: y_i
- “Controls” w account for unknown disturbances. Often many w.
- Initial value is free

NMPC = mp-NLP

- Solution manifold is piecewise differentiable

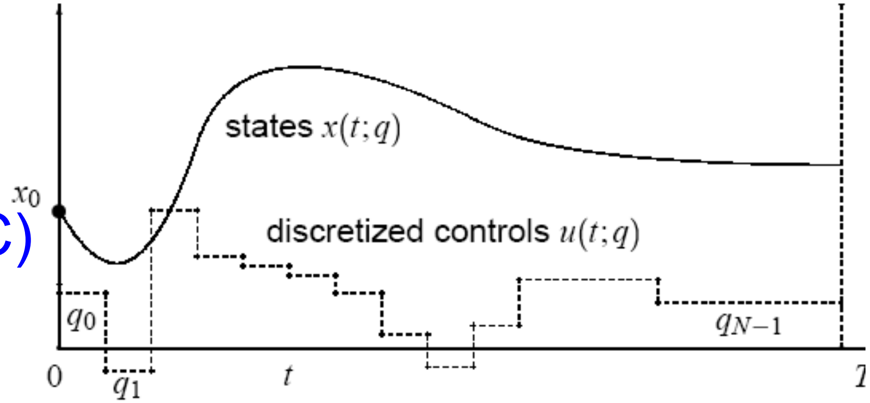


- Critical regions are non-polyhedral



Sequential Approach (Single Shooting): Eliminate States

$$
\begin{array}{lrcl}
\underset{u}{\text{minimize}} & \displaystyle\sum_{i=0}^{N-1} L_i(\tilde{x}_i(u), \tilde{z}_i(u), u_i) & + & E\left(\tilde{x}_N(u)\right) \\
\text{subject to} & h_i(\tilde{x}_i(u), \tilde{z}_i(u), u_i) & \leq & 0, \quad i = 0, \ldots, N-1, \\
 & r\left(\tilde{x}_N(u)\right) & \leq & 0.
\end{array}
$$



Pros:
- Only control degrees of freedom (for NMPC)
- Can couple with "Vanilla NLP" solver

Cons:
- Sparsity of problem lost
- Unstable systems cannot be treated

Historically first "direct" approach ("single shooting", Sargent&Sullivan 1978)

Simultaneous Approach: Keep States in NLP

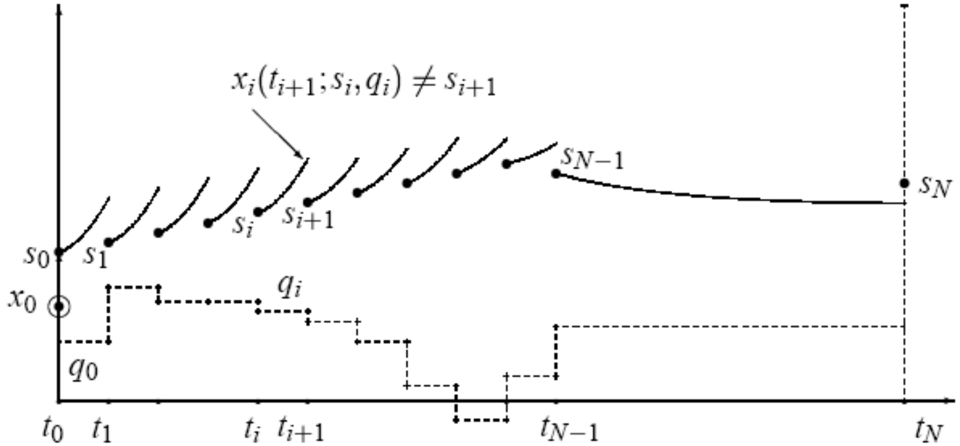
INTERNATIONAL FEDERATION
OF AUTOMATIC CONTROL
9TH WORLD CONGRESS

BUDAPEST, HUNGARY
JULY 2-6 1984

A MULTIPLE SHOOTING ALGORITHM FOR DIRECT SOLUTION OF OPTIMAL CONTROL PROBLEMS*

Hans Georg Bock and Karl J. Plitt

Institut für Angewandte Mathematik, SFB 72, Universität Bonn, 5300 Bonn, Federal Republic of Germany



Variants:
Multiple Shooting and Collocation

Pros:
- Sparsity of problem kept
- Unstable systems can be treated, nonlinearity reduced

Cons:
- Large scale problems
- Need to develop (or use) structure exploiting NLP solver

Outline of the Talk

PART I: Offline Optimal Control

- NMPC and MHE Problem Statement
- Simultaneous vs. Sequential Formulation
- **Newton Type Optimization: IP vs. SQP Methods**

PART II: Online Algorithms

- Parametric Sensitivities
- Review of Three Classical Algorithms

PART III: Software and Mechatronic Applications

How to solve Nonlinear Programs (NLPs) ?

$$\boxed{\underset{X}{\text{minimize}} \; F(X) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} G(X) & = & 0 \\ H(X) & \leq & 0 \end{array} \right.}$$

Lagrangian: $\mathcal{L}(X, \lambda, \mu) = F(X) + G(X)^T \lambda + H(X)^T \mu$

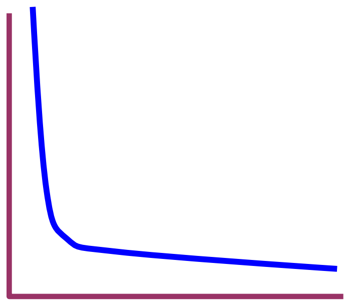
Karush Kuhn Tucker (KKT) conditions: for optimal X^* exist λ^*, μ^* such that:

$$\boxed{\begin{array}{rcl} \nabla_X \mathcal{L}(X^*, \lambda^*, \mu^*) & = & 0 \\ G(X^*) & = & 0 \\ 0 \geq H(X^*) \quad \perp \quad \mu^* & \geq & 0. \end{array}}$$

Newton type methods try to find points satisfying these conditions. But last condition non-smooth: cannot apply Newton's method directly. What to do?

Approach 1: Interior Point (IP) Methods

- Replace last condition by smoothed version:



$$\boxed{\begin{aligned} \nabla_X \mathcal{L}(X^*, \lambda^*, \mu^*) &= 0 \\ G(X^*) &= 0 \\ -\,H_i(X^*)\,\mu_i^* &= \tau, \quad i = 1, \ldots, n_H. \end{aligned}}$$

Summarize as $R(W) = 0$

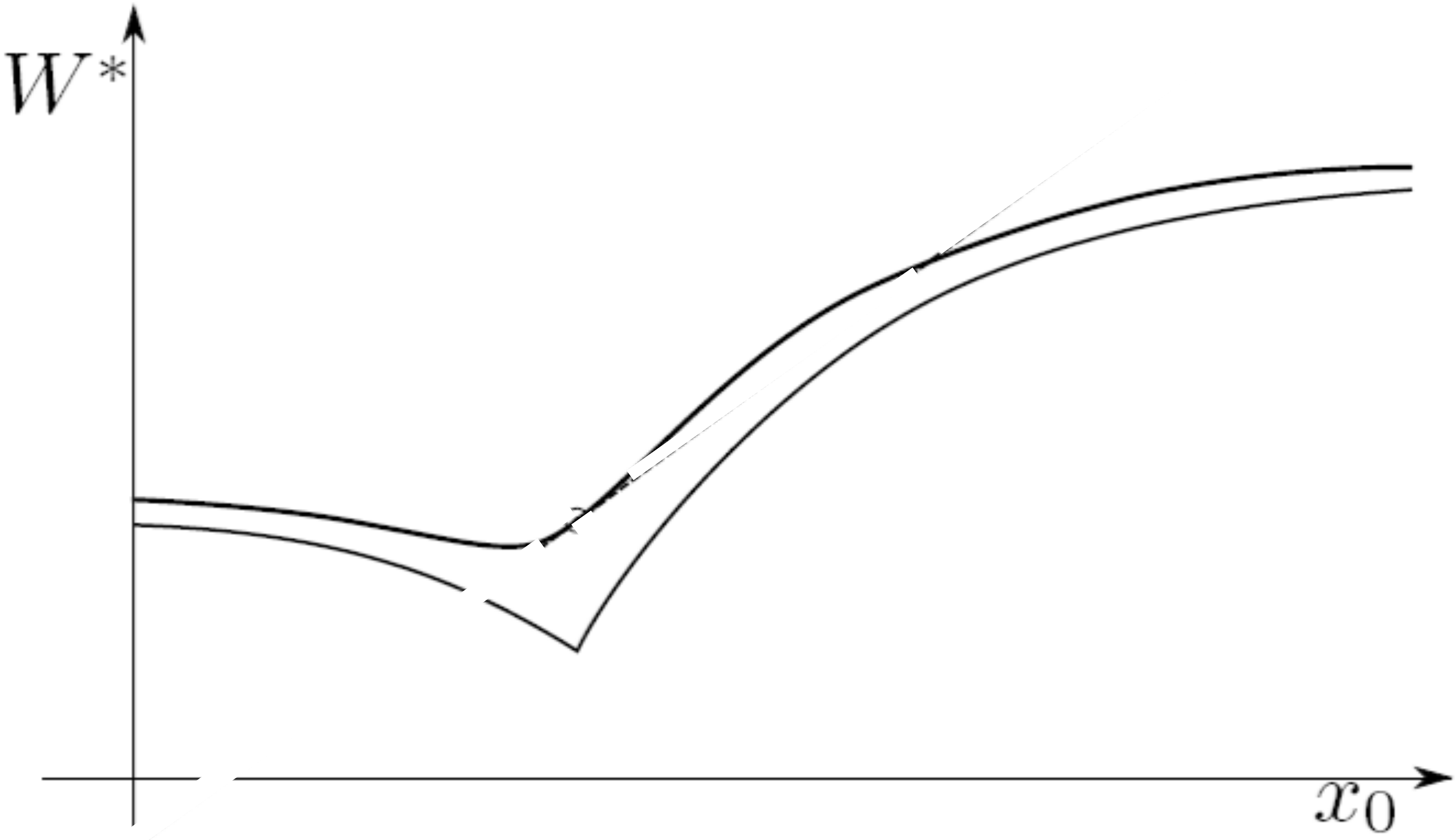
- Solve with Newton's method, i.e.,
  - Linearize at current guess $W^k = (X^k, \lambda^k, \mu^k)$

$$\boxed{R(W^k) + \nabla R(W^k)^T (W^{k+1} - W^k) = 0}$$

  - solve linearized system, get new trial point

- For τ small, IP problem gets close to original (path-following, self-concordance, polynomial time for convex problems, …)

(Note: IP with fixed τ makes mp-NLP smooth)



Approach 2: Sequential Quadratic Programming (SQP)

Mathematical Programming 14 (1978) 224–248.

ALGORITHMS FOR NONLINEAR CONSTRAINTS THAT USE LAGRANGIAN FUNCTIONS*

M.J.D. POWELL

University of Cambridge, Cambridge, United Kingdom

Received 10 October 1976

- Linearize all problem functions, solve Quadratic Program (QP):

$$\underset{X}{\text{minimize}} \; F_{\mathrm{QP}}^{k}(X) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} G(X^k) + \nabla G(X^k)^T (X - X^k) & = & 0 \\ H(X^k) + \nabla H(X^k)^T (X - X^k) & \leq & 0 \end{array} \right.$$

with convex quadratic objective using an approximation of Hessian.

Obtain new guesses for both X^* and λ^*, μ^* .

(Important Variant of SQP: Generalized Gauss-Newton)

If objective is sum of squares: $F(X) = \frac{1}{2}\|R(X)\|_2^2.$

then QP objective can well be approximated by:

$$F_{\mathrm{QP}}^k(X) = \frac{1}{2}\|R(X^k) + \nabla R(X^k)^T(X - X^k)\|_2^2$$

(often extremely good linear convergence)

Difference between IP and SQP ?

- Both generate sequence of iterates X^k, λ^k, μ^k
- Both need to linearize problem functions in each iteration.

- IP iterations cheaper:
  - IP solves only linear system
  - SQP solves a QP in each iteration (maybe even with an IP method!)

- IP needs more iterations:
  - IP multipliers change slowly, iterates always in interior
  - SQP multipliers jump, active set can quickly be identified

SQP good if problem function evaluations are expensive (shooting methods)

(Adjoint Based SQP: can use old Jacobians)

- SQP even works if all QP matrices are old. Only constraints and Lagrange gradient (cheap by adjoint differentiation) need to be exact.

- Trick: use "modified gradient" $a_k = \nabla_X \mathcal{L}(X^k, \lambda^k, \mu^k) - B_k \lambda^k - C_k \mu^k$

  in QP objective $F^k_{\mathrm{adjQP}}(X) = a_k^T X + \frac{1}{2}(X - X^k)^T A_k (X - X^k).$

Solve QP with inexact constraints

$$\boxed{\underset{X}{\text{minimize}} \quad F^k_{\mathrm{adjQP}}(X) \quad \text{s.t.} \quad \left\{ \begin{array}{lcl} G(X^k) + B_k^T (X - X^k) & = & 0 \\ H(X^k) + C_k^T (X - X^k) & \leq & 0. \end{array} \right.}$$

Can prove stability of active set and linear convergence [Bock, D., Kostina, Schloeder 2007]. Adjoint SQP iterations often orders of magnitude cheaper than full SQP iterations.

Linear Algebra Issues in Optimal Control

- In each SQP iteration, solve structured QP:

$$
\begin{array}{rcll}
\underset{x,\,z,\,u}{\text{minimize}} \quad \displaystyle\sum_{i=0}^{N-1} L_{\mathrm{QP},i}(x_i, z_i, u_i) & + & E_{\mathrm{QP}}(x_N) & \\
\text{subject to} \qquad x_0 - \bar{x}_0 & = & 0, & \\
x_{i+1} - f_i' - F_i^x x_i - F_i^z z_i - F_i^u u_i & = & 0, & i = 0, \ldots, N-1, \\
g_i' + G_i^x x_i + G_i^z z_i + G_i^u u_i & = & 0, & i = 0, \ldots, N-1, \\
h_i' + H_i^x x_i + H_i^z z_i + H_i^u u_i & \leq & 0, & i = 0, \ldots, N-1, \\
r' + R x_N & \leq & 0. &
\end{array}
$$

- Algebraic Reduction/compression: first eliminate z

QP after Algebraic Reduction

$$
\begin{aligned}
\underset{x,\,u}{\text{minimize}} \quad & \sum_{i=0}^{N-1} L_{\text{redQP},i}(x_i, u_i) \quad + \quad E_{\text{QP}}(x_N) \\
\text{subject to} \quad x_0 - \bar{x}_0 &= 0, \\
x_{i+1} - c_i - A_i x_i - B_i u_i &= 0, \quad i = 0, \ldots, N-1, \\
\bar{h}_i + \bar{H}_i^x x_i + \bar{H}_i^u u_i &\leq 0, \quad i = 0, \ldots, N-1, \\
r' + R x_N &\leq 0.
\end{aligned}
$$

How to solve this structured QP?

Approach 1: Banded Factorization

- Factorize large banded KKT Matrix e.g. by Riccati based recursion

$$M = \begin{bmatrix} & \mathbb{I} & & & & \\ \mathbb{I} & Q_0 & S_0 & -A_0^T & & \\ & S_0^T & R_0 & -B_0^T & & \\ & -A_0 & -B_0 & \ddots & \mathbb{I} & \\ & & & \mathbb{I} & Q_N & \end{bmatrix}$$

- Advantageous for long horizons and many controls

Approach 2: Condensing - Eliminate all States

- Eliminate states by linear system simulation, keep only controls in QP, solve QP with dense solver

$$\begin{array}{lll} & \underset{u}{\text{minimize}} & f_{\text{condQP},i}(\bar{x}_0, u) \\ \text{subject to} & \bar{r} + \bar{R}^{x0}\bar{x}_0 + \bar{R}^u u & \leq & 0. \end{array}$$

- Note: mp-QP in same form as needed by qpOASES and explicit MPC
- Can use this QP as fast feedback law for several $\bar{x}_0$
- But QP matrices change after each SQP re-linearization

Outline of the Talk

PART I: Offline Optimal Control

- NMPC and MHE Problem Statement
- Simultaneous vs. Sequential Formulation
- Newton Type Optimization: IP vs. SQP Methods

PART II: Online Algorithms

- Parametric Sensitivities
- Review of Three Classical Algorithms

The start: “Newton-Type Controller” by Li and Biegler

Chem Eng Res Des, Vol. 67, November 1989

MULTISTEP, NEWTON-TYPE CONTROL STRATEGIES FOR CONSTRAINED, NONLINEAR PROCESSES

W. C. LI and L. T. BIEGLER
Carnegie-Mellon University, Department of Chemical Engineering, Pittsburgh, USA

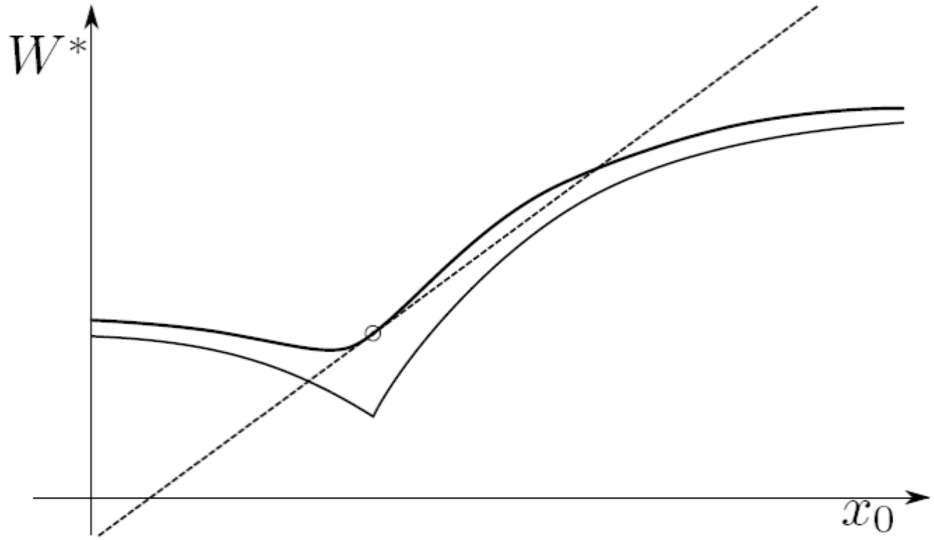
- SQP type method
- single shooting
- perform only one SQP iteration per problem (“real-time iteration”)
- Method also implemented in “NEPSAC Algorithm” by [De Keyser 1998] and many others [IPCOS, GE, …]. Many applications.
- was missing one important feature: *parametric sensitivities*

Parametric Sensitivities

- In IP case, smoothed KKT conditions are equivalent to parametric root finding problem: $$R(\bar{x}_0, W) = 0$$

  with solution $W^*(\bar{x}_0)$ depending on initial condition

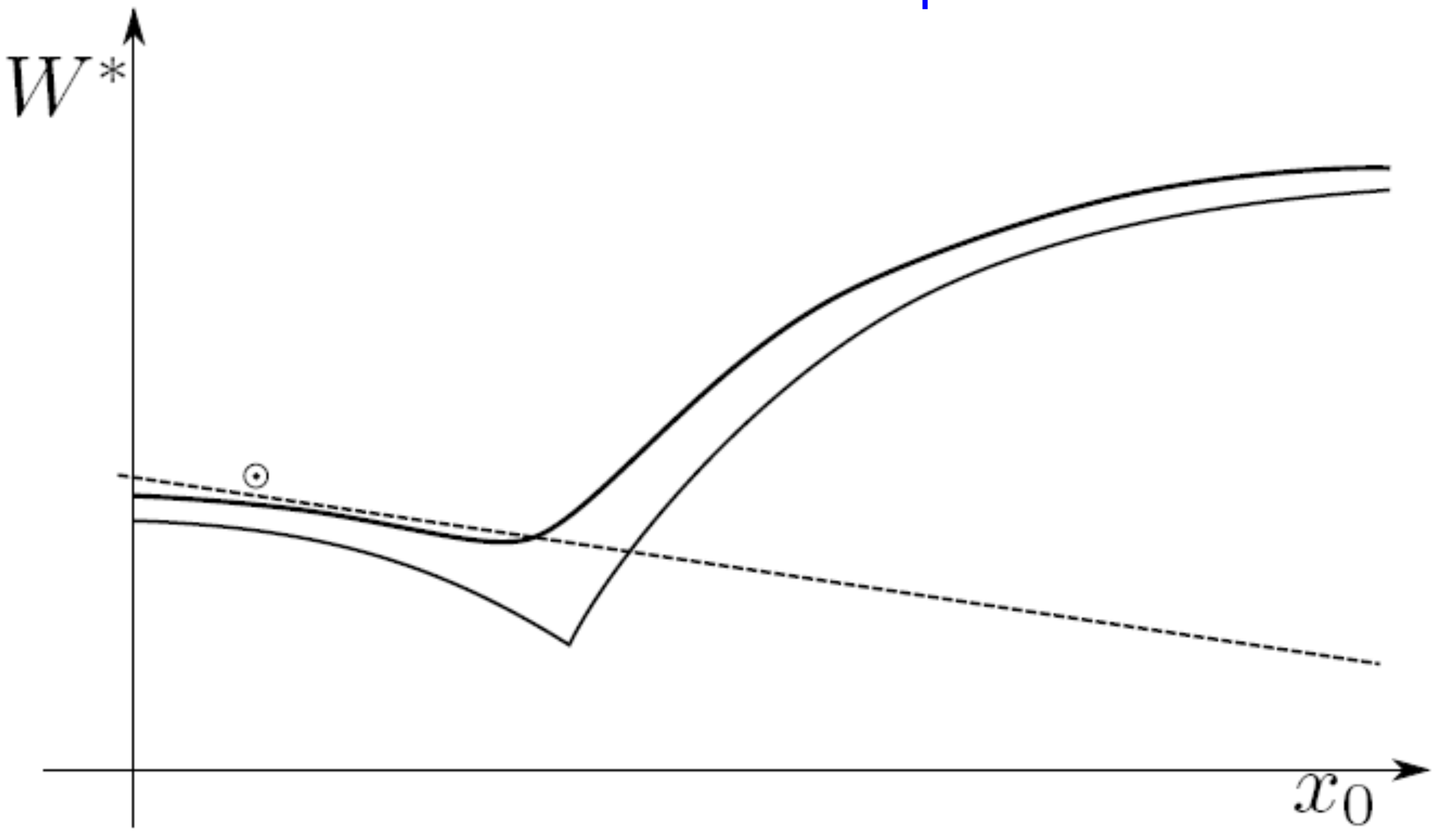
Based on old solution, can get "tangential predictor" for new one:



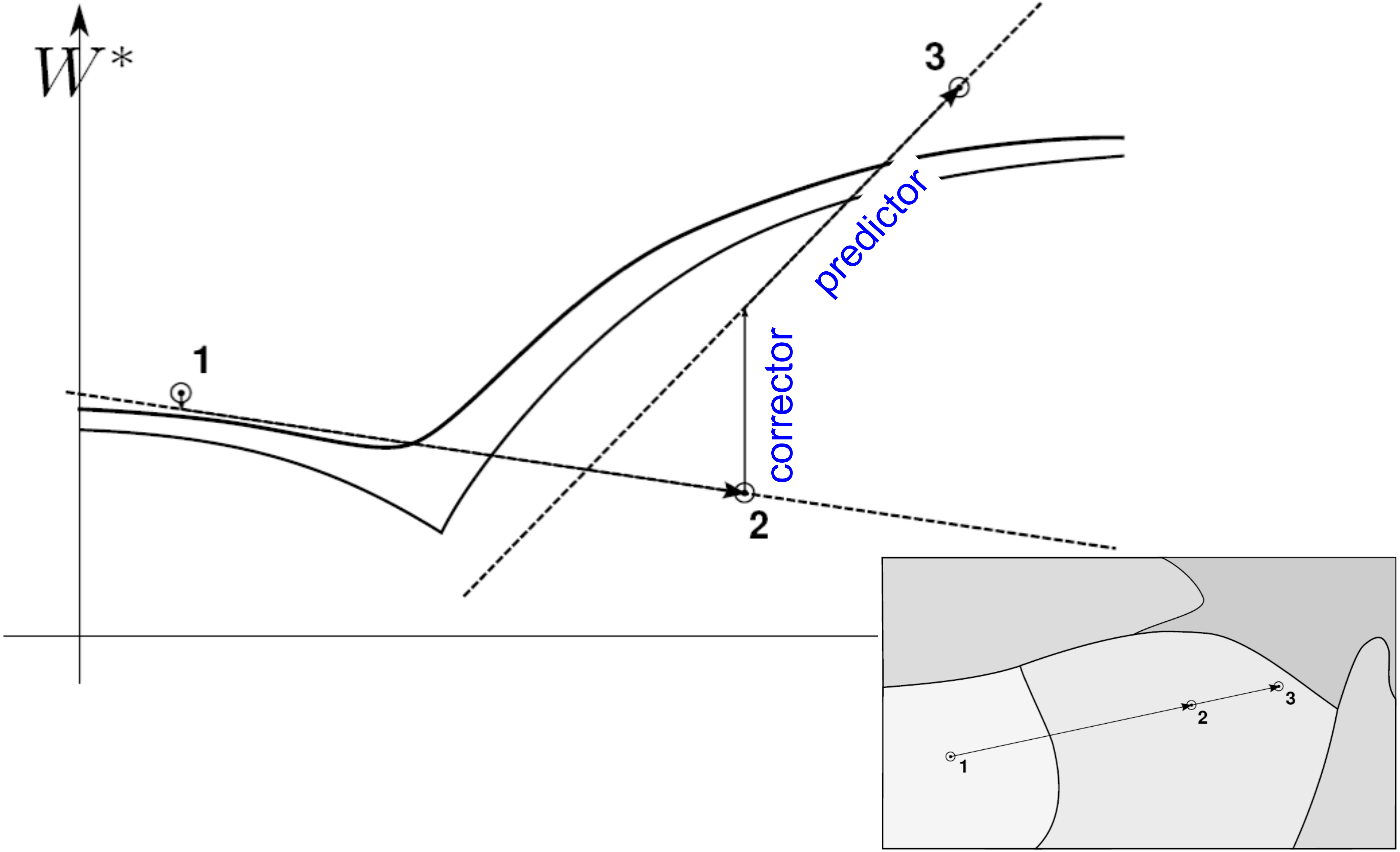
Initial Value Embedding Idea

- Can obtain sensitivity nearly for free in Newton type methods:

$$W' = W - \left(\frac{\partial R}{\partial W}(\bar{x}_0, W)\right)^{-1} \left[\underbrace{\frac{\partial R}{\partial \bar{x}_0}(\bar{x}_0, W)\left(\bar{x}_0' - \bar{x}_0\right)}_{\text{predictor}} + \underbrace{R(\bar{x}_0, W)}_{\text{corrector}}\right]$$



“IP real-time iteration” for sequence of NLPs



IP Real-Time Iteration ≈ Ohtsuka's Continuation Method

ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Automatica 40 (2004) 563–574

automatica

www.elsevier.com/locate/automatica

A continuation/GMRES method for fast computation of nonlinear receding horizon control☆

Toshiyuki Ohtsuka*

Department of Computer-Controlled Mechanical Systems, Graduate School of Engineering, Osaka University, 2-1 Yamadaoka, Suita, Osaka 565-0871, Japan

Additional features of Continuation/GMRES:

- matrix free and iterative linear algebra via GMRES
- Interior Point formulation via quadratic slacks
- Single shooting with adjoint gradient computation
- (recently extended to multiple shooting with condensing, faster contraction)

But: Ohtsuka's method "overshoots" at active set changes

(Variant of IP Methods: Quadratic Slacks)

- Ohtsuka (2004) uses for NMPC a variant of IP methods:

$$\underset{X,Y}{\text{minimize}} \;\; F(X) - \tau \sum_{i=1}^{n_H} Y_i \quad \text{s.t.} \begin{cases} G(X) &= 0 \\ H_i(X) + Y_i^2 &= 0, \; i = 1, \ldots, n_H. \end{cases}$$

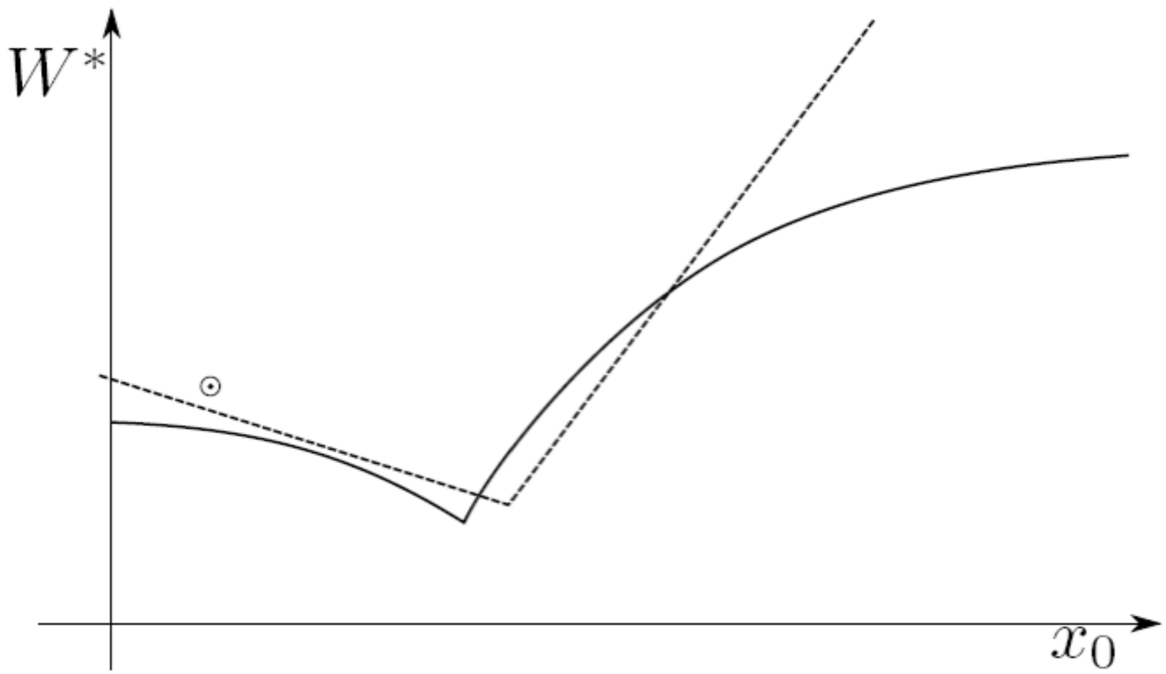
Seems to work well for fixed penalty parameter. But no self-concordance properties as in usual IP methods.

Generalized Tangential Predictor via SQP

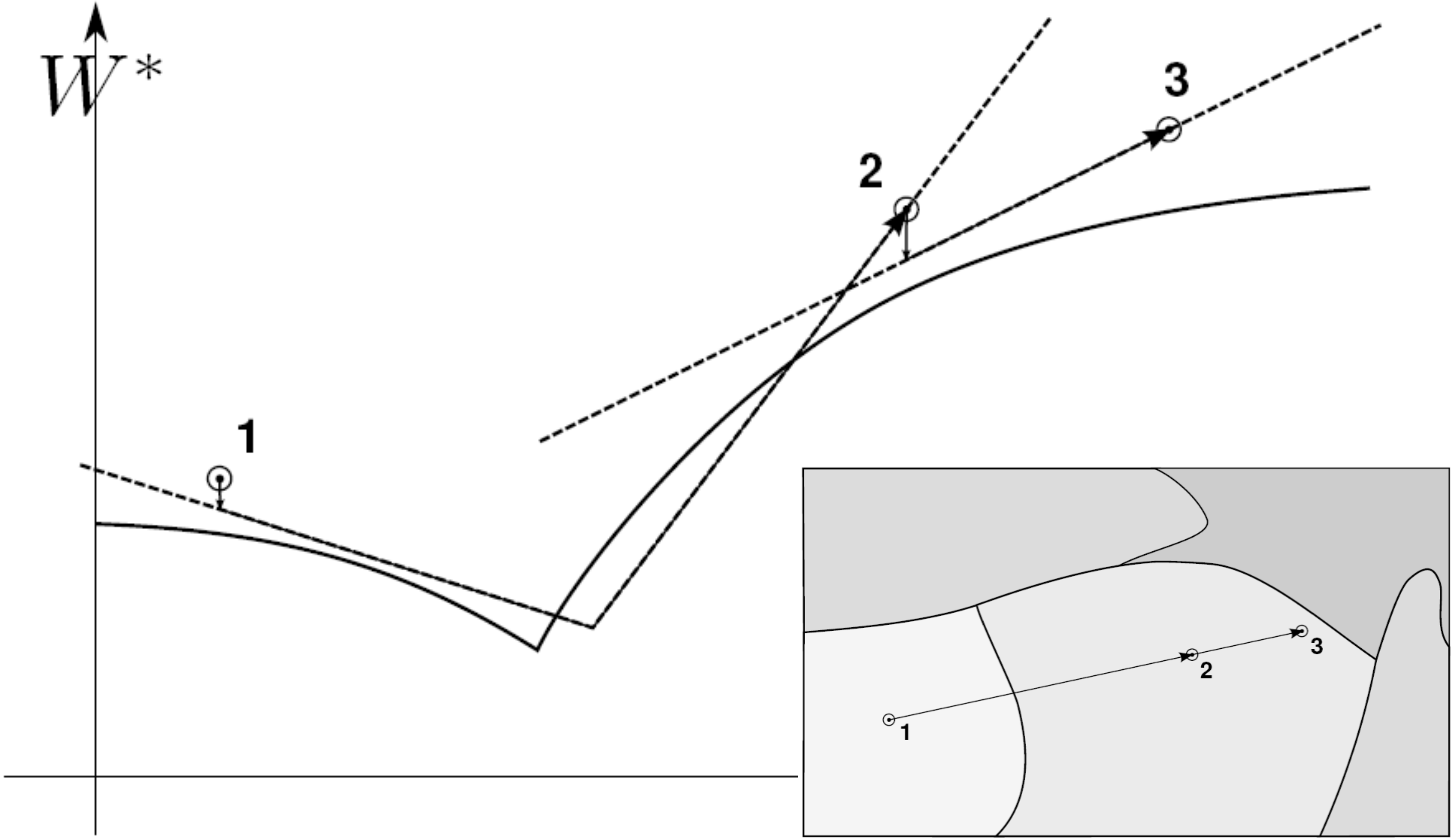
- Solve a full QP with "initial value embedding" [D. et al. 2002].

$$\begin{array}{lll} & \underset{u}{\text{minimize}} & f_{\text{condQP},i}(\bar{x}_0, u) \\ \text{subject to} & & \bar{r} + \bar{R}^{x_0}\bar{x}_0 + \bar{R}^u u \quad \leq \quad 0. \end{array}$$

- At smooth parts, delivers same predictor-corrector step as Newton. But is "Generalized Tangential Predictor" valid also across active set changes:

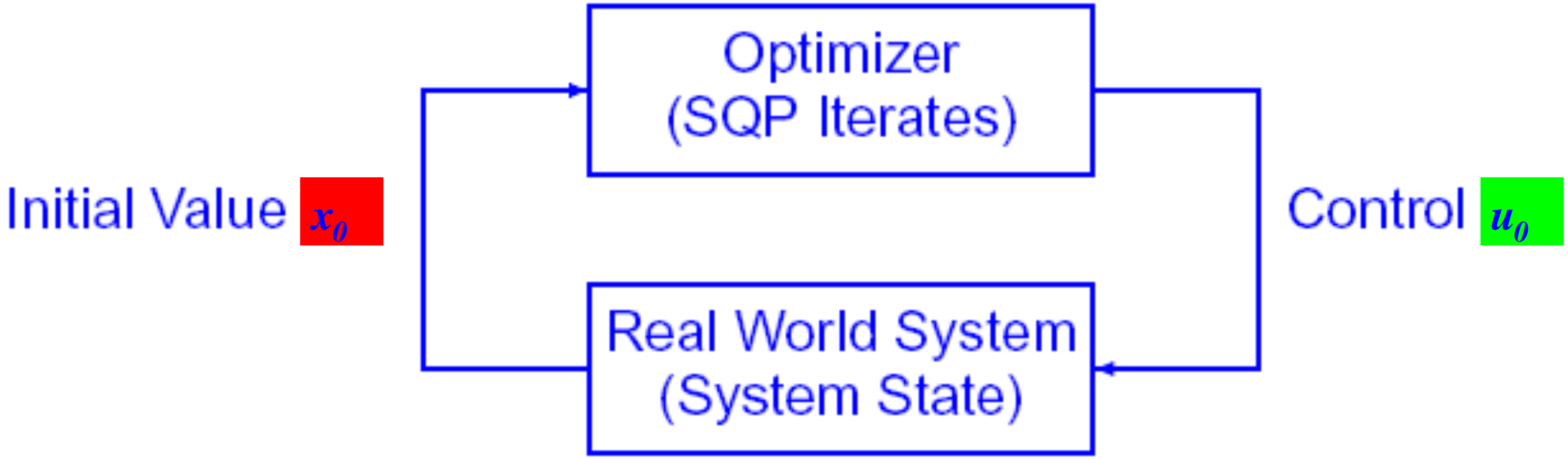


SQP Real-Time Iteration [D. et al 2002]



- long "preparation phase" for linearization, reduction, and condensing
- fast "feedback phase" (QP solution once $\bar{x}_0$ is known). Fast, but…

Stability of System-Optimizer Dynamics?

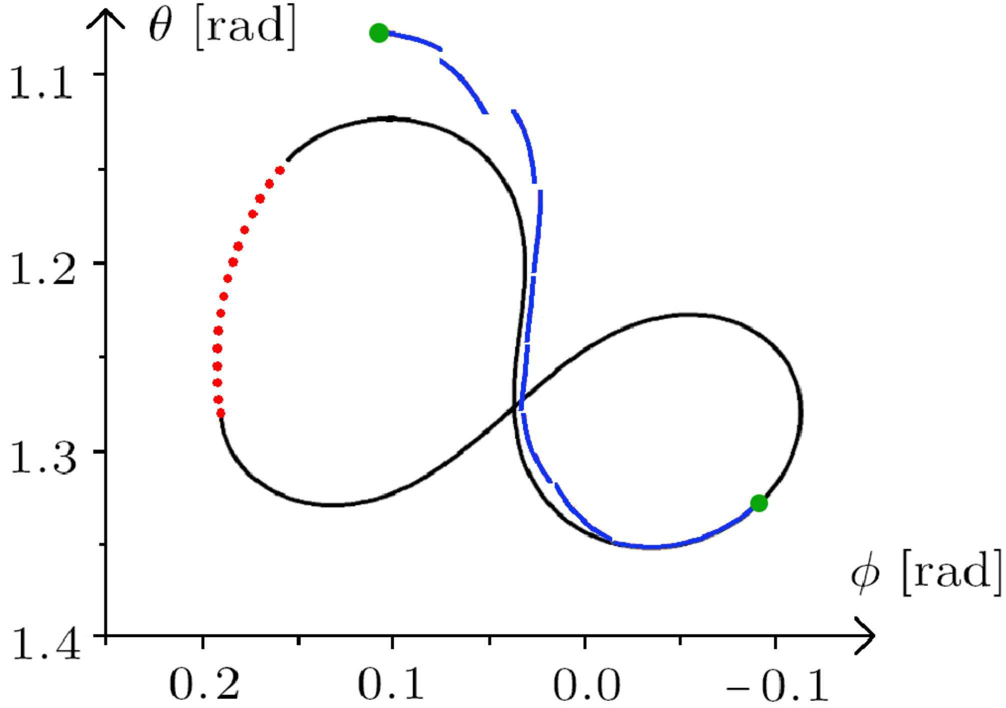


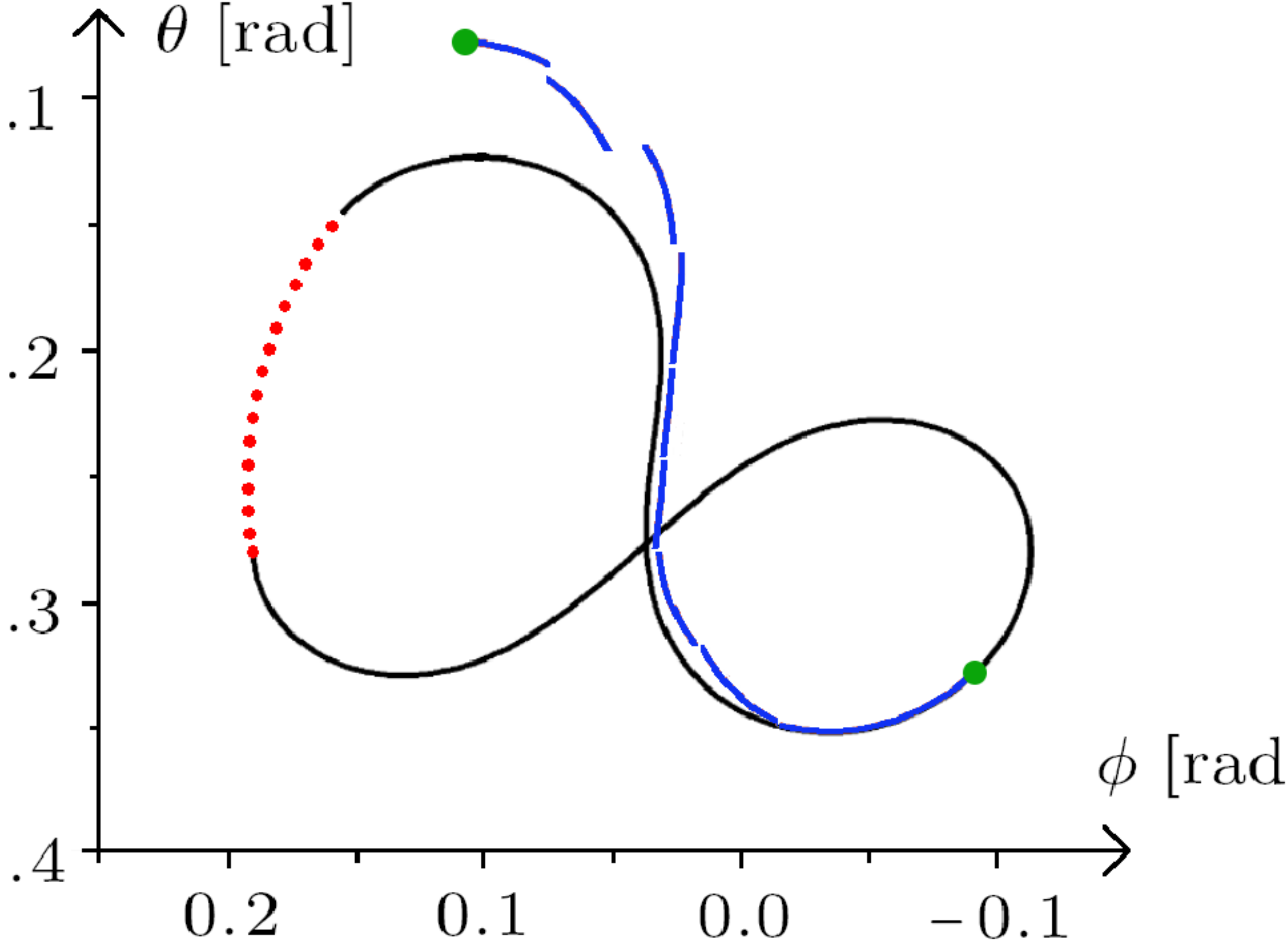
- System and optimizer are coupled: can numerical errors grow and destabilize closed loop?
- Stability analysis combines concepts from both, **NMPC stability theory** and **convergence theory of Newton-type optimization**.
- Stability shown under mild assumptions (short sampling times, stable NMPC scheme) [Diehl, Findeisen, Allgöwer, 2005]
- Losses w.r.t. optimal feedback control are $O(\kappa^2\epsilon^2)$ after ϵ disturbance [Diehl, Bock, Schlöder, 2005]

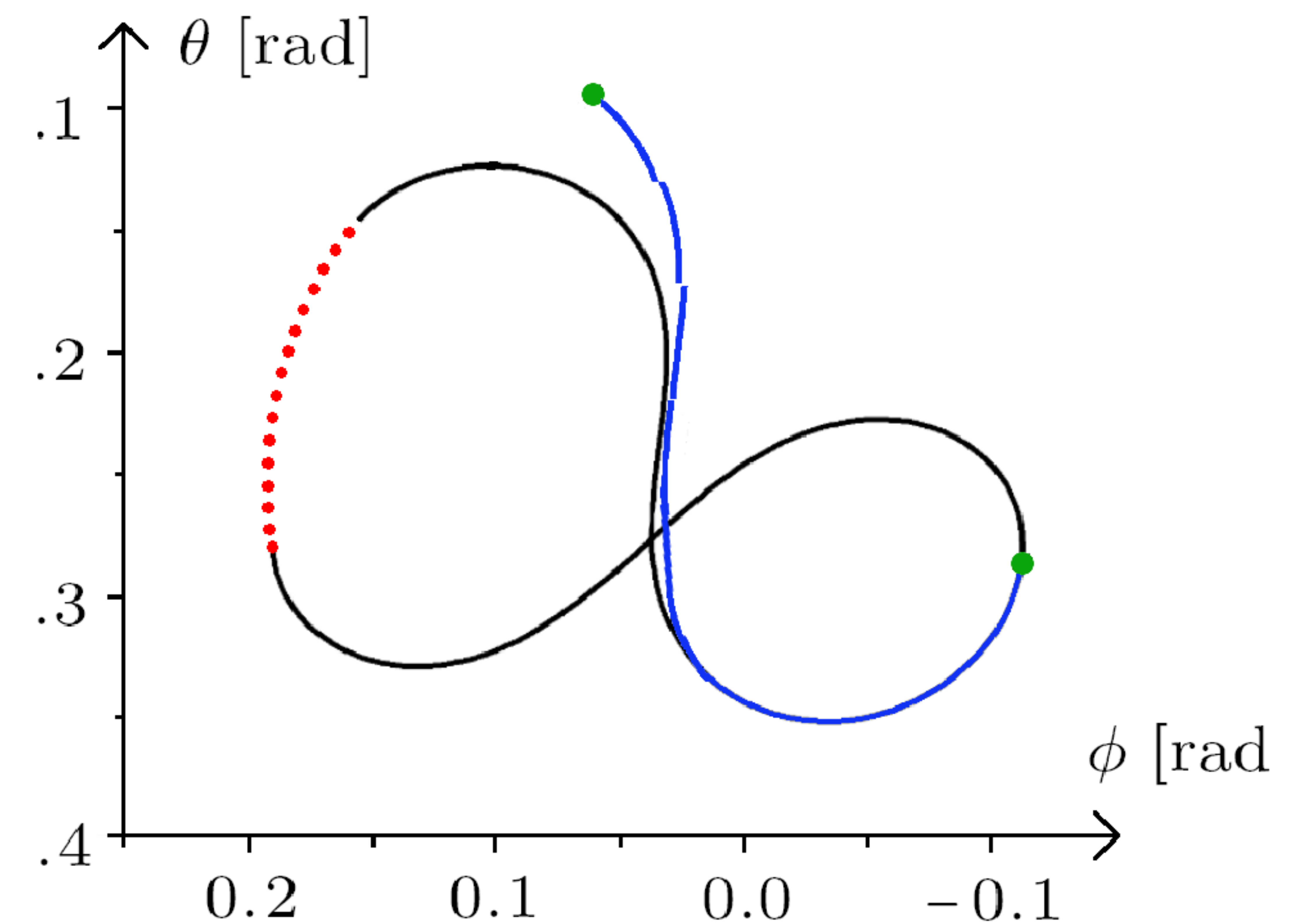
Kite NMPC Problem solved with ACADO (B. Houska)

- 9 states, 3 controls
- Penalize deviation from "lying eight"
- Predict half period
- zero terminal constraint
- 10 multiple shooting intervals

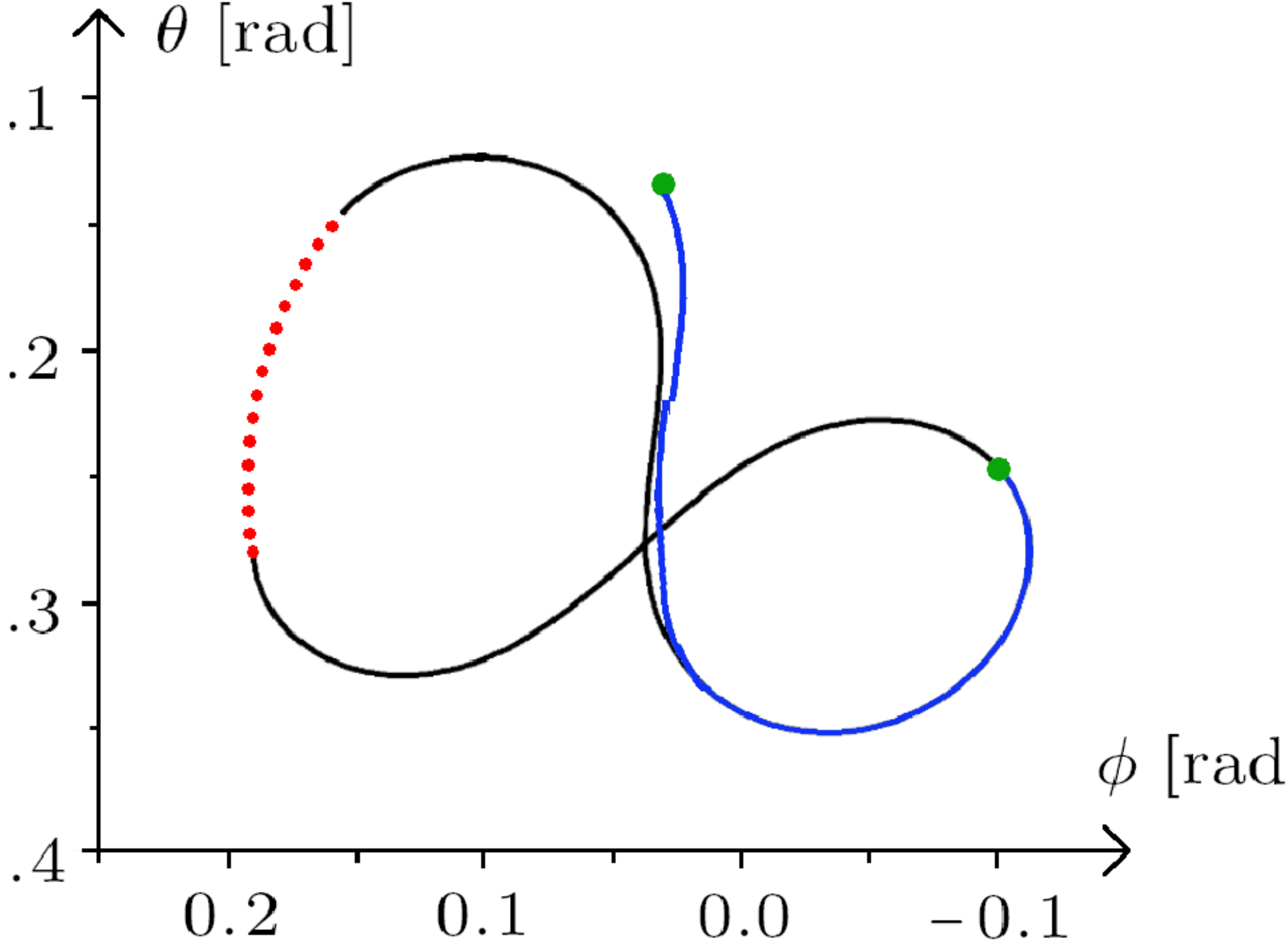
Solve with **SQP real-time iterations with shift** (implemented in ACADO)



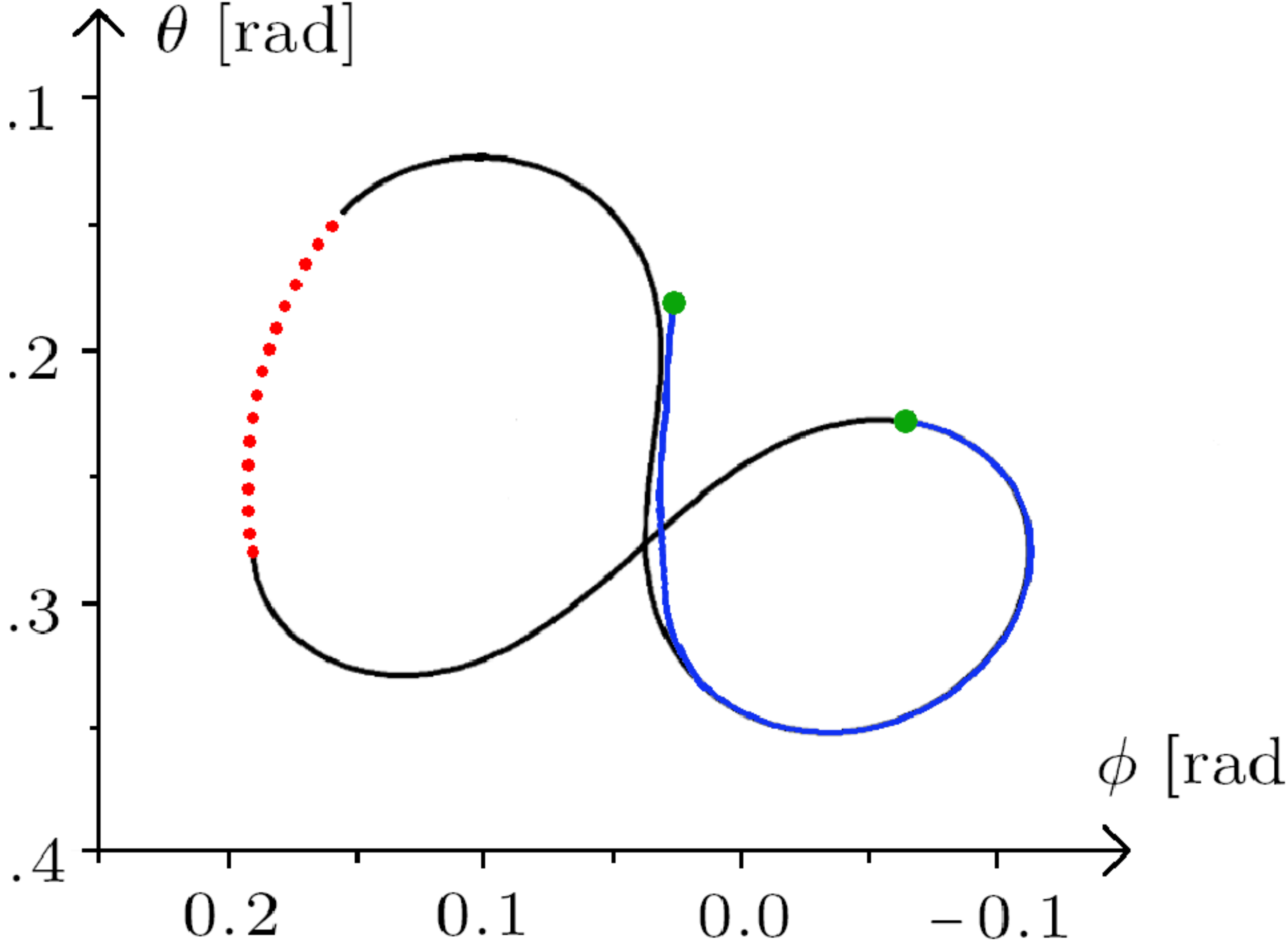

θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1



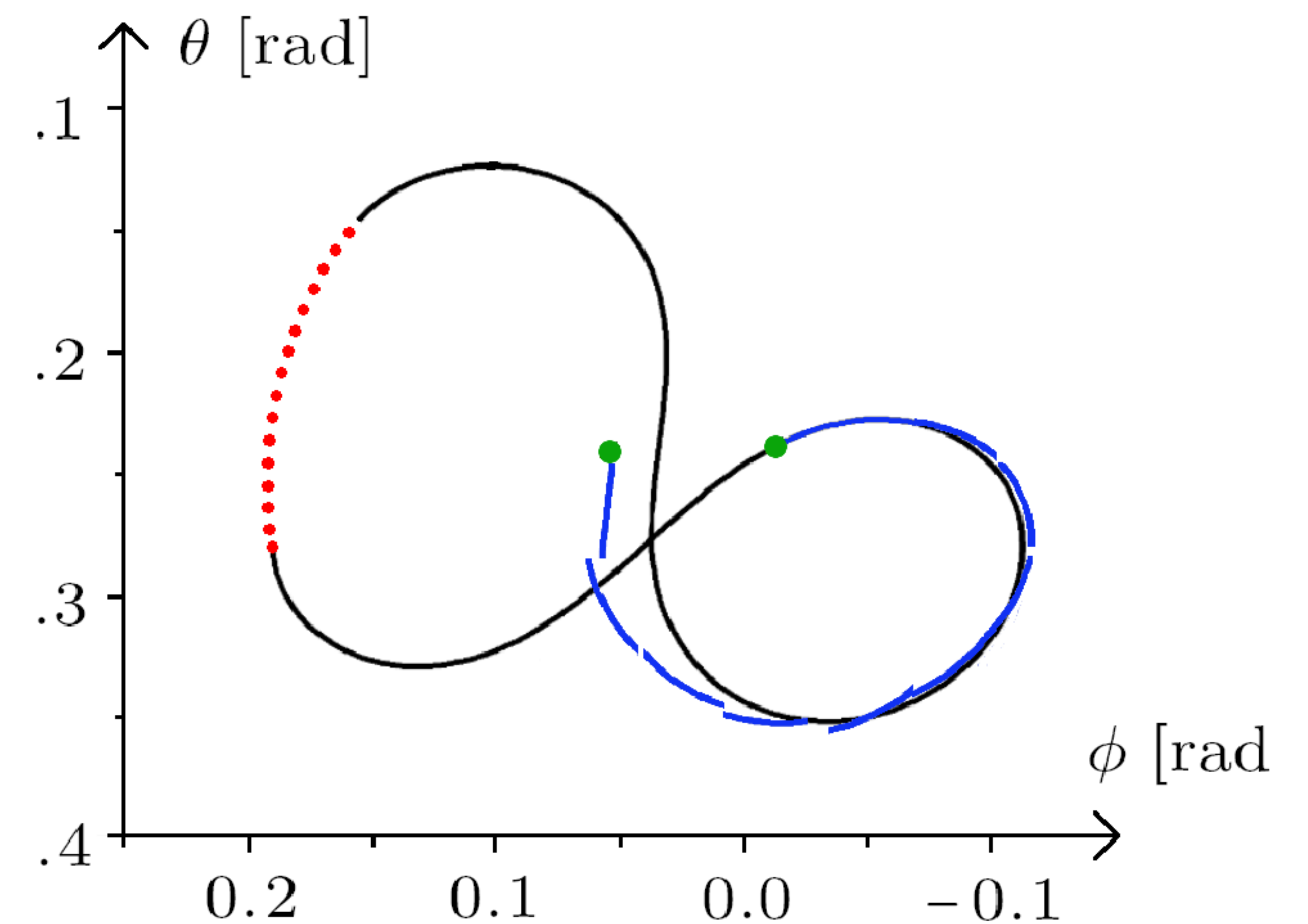
θ [rad]
.1
.2
.3
.4
φ [rad
0.2
0.1
0.0
−0.1



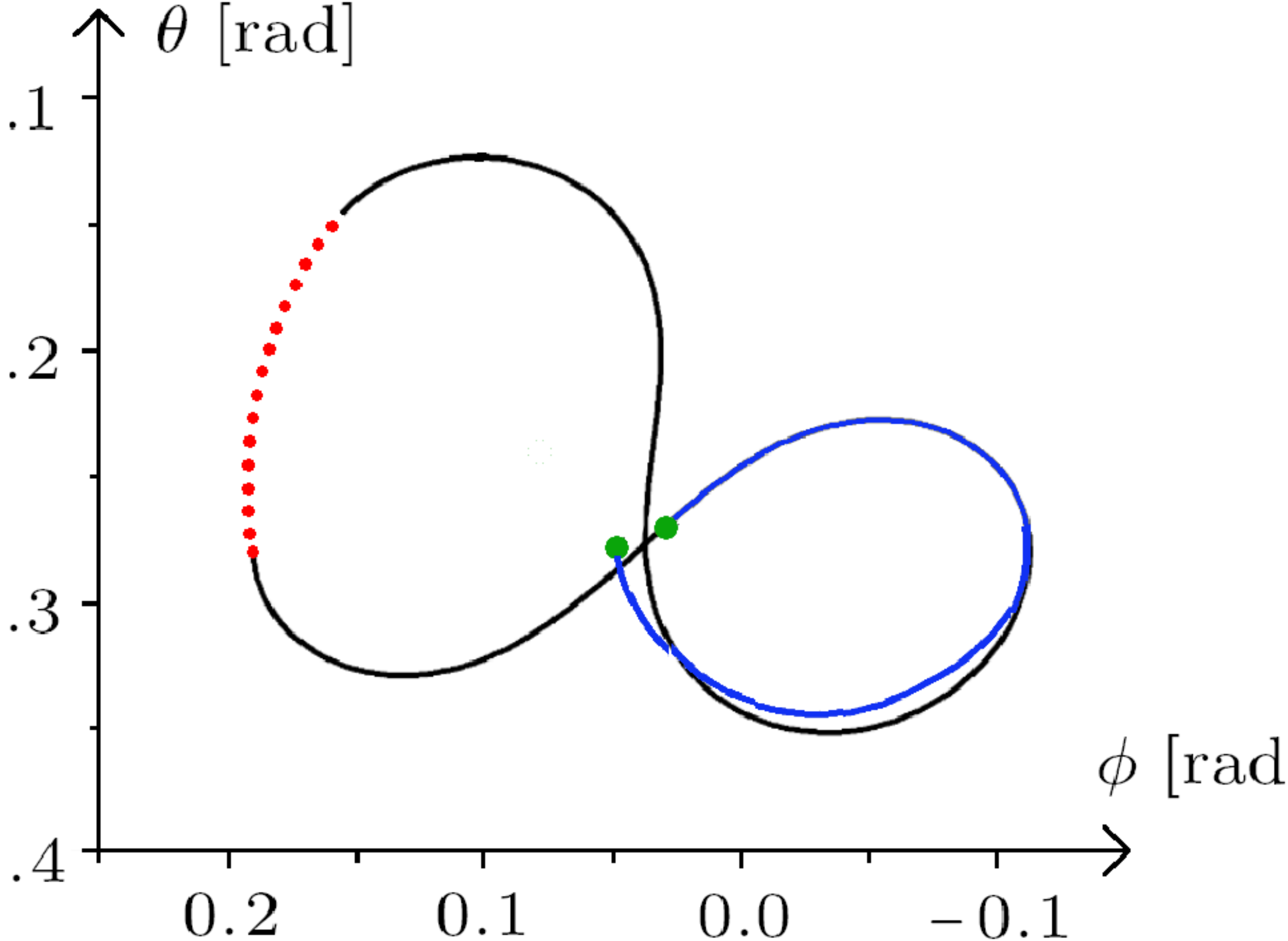
θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1

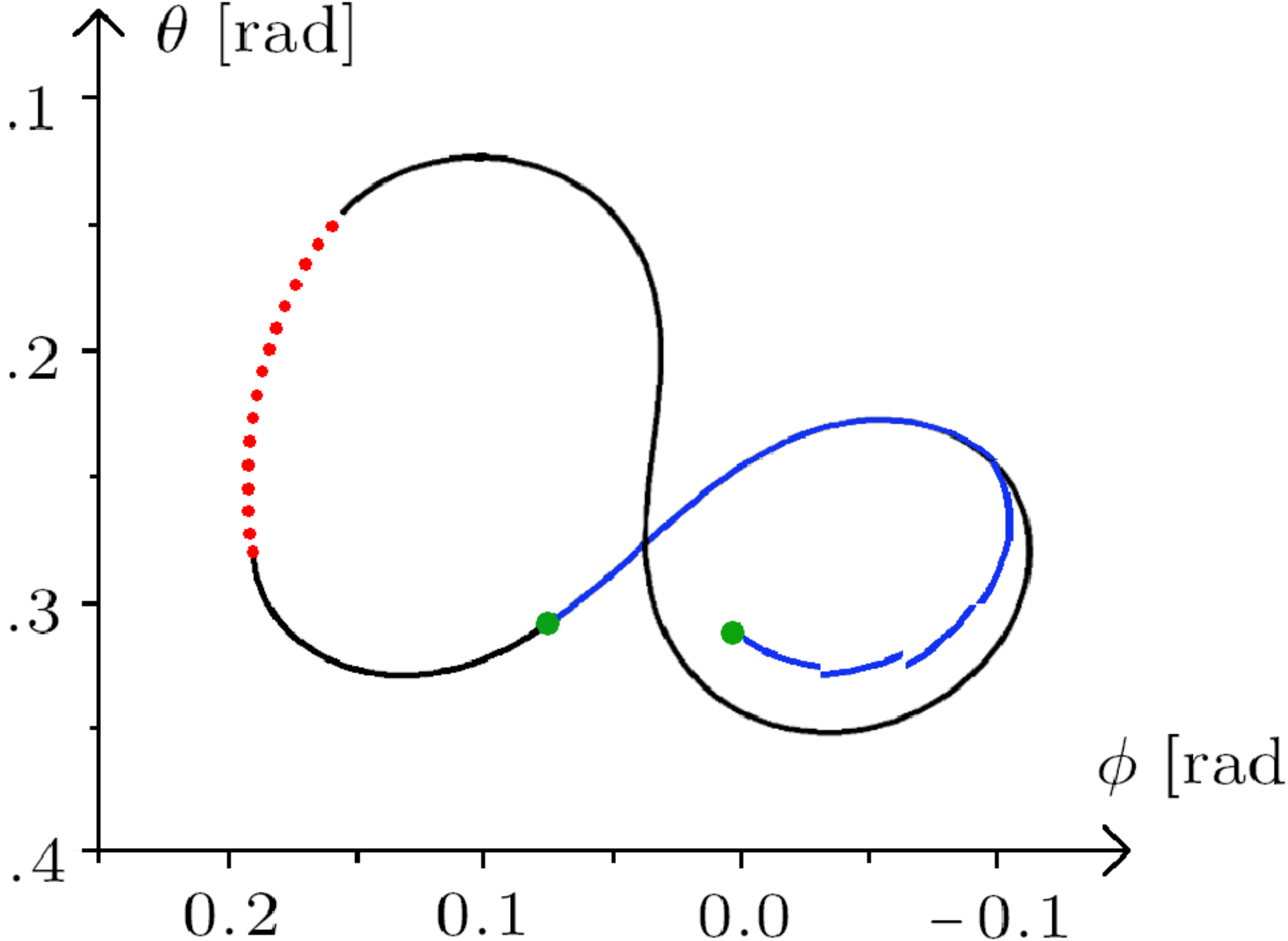


θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1

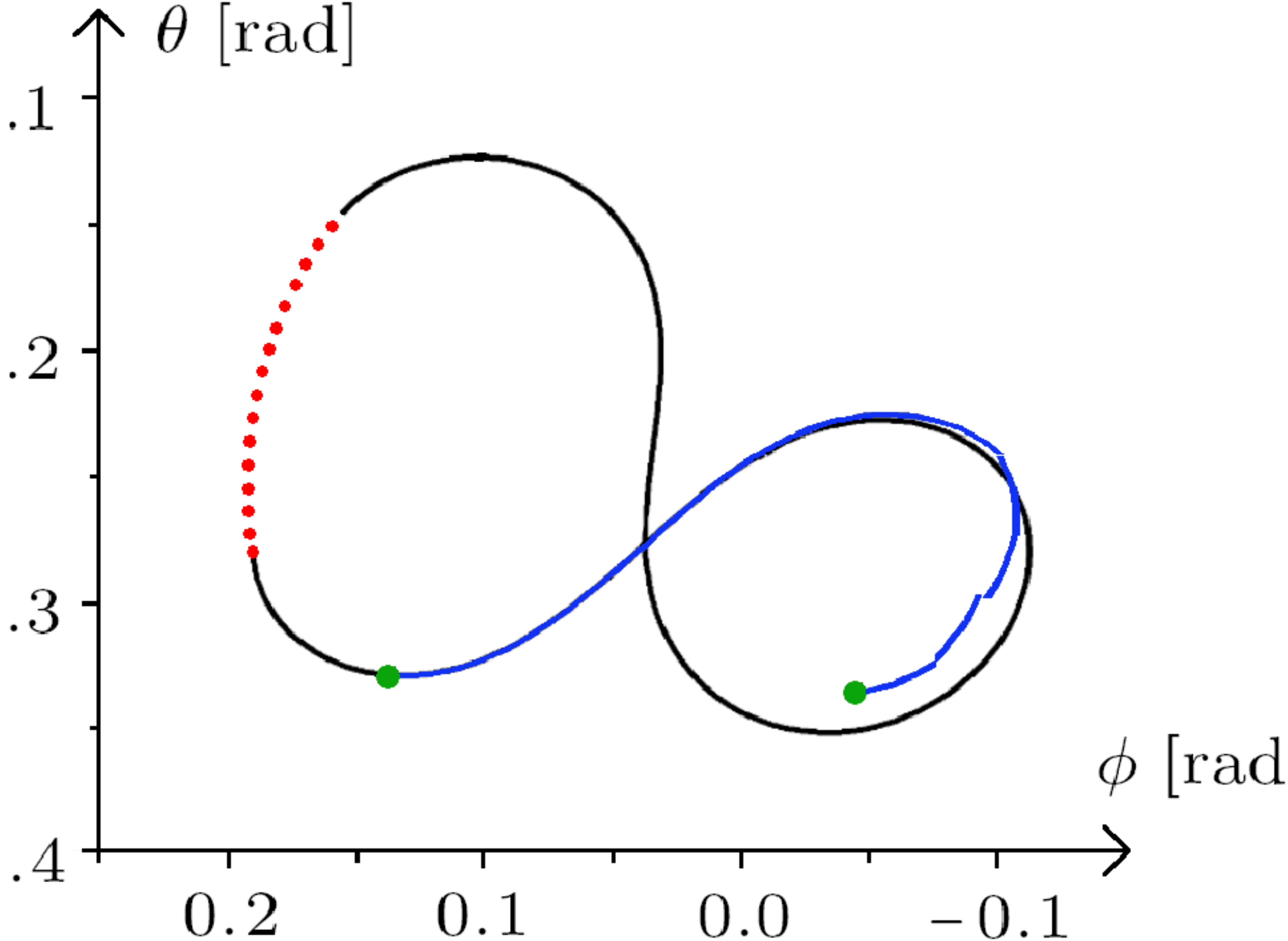


θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1


θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1



θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1



θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1

Kite NMPC: CPU Time per RTI below 50 ms

- Initial-Value Embedding : 0.03 ms
- QP solution (qpOASES) : 2.23 ms

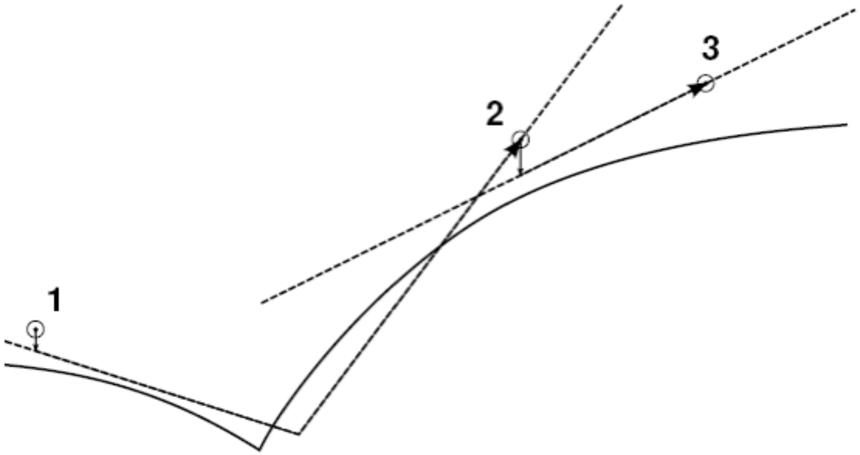
Feedback Phase: 3 ms

(QP after condensing: 30 vars. / 240 constr.)

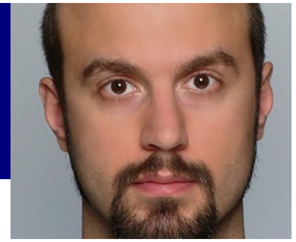
- Expansion of the QP : 0.10 ms
- Simulation and Sensitivities : 44.17 ms
- Condensing (Phase I) : 2.83 ms

Preparation Phase: 47 ms

(on Intel Core 2 Duo CPU T7250, 2 GHz…
without code generation yet)



Nonlinear MPC and MHE on Flight Carousel



Milan Vukov

(sampling time 50 Hz, using ACADO Code Generation)

Closed loop experiments with NMPC & NMHE

KU LEUVEN

UNI FREIBURG

ERC HIGHWIND

Further algorithmic developments in opposite directions

Multi-Level Real-Time Iterations

[Bock, D. et al. NMPC 05, Wirsching 2007]

Make real-time iterations cheaper.

Four Levels:

A) mp-QP at innermost level

B) Feasibility improvement

C) Optimality Improvement

D) Full re-linearization, only rarely in outer loop

- Allows extremely fast sampling rates at innermost level A (feedback phase).
- Level C allows to converge to NLP solution WITHOUT NEW JACOBIAN EVALUATIONS.

Advanced Step NMPC

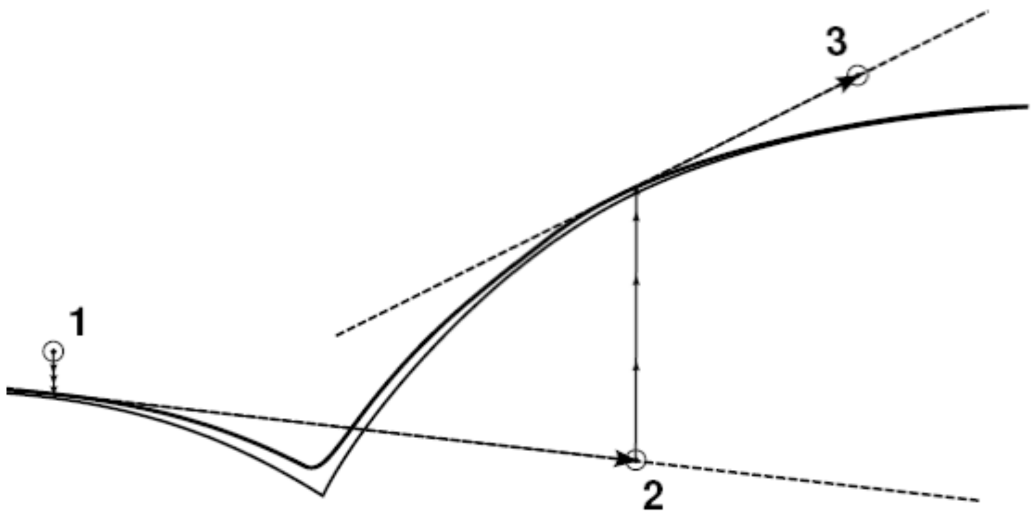
[Zavala and Biegler 2007]

Combine two well-tested ideas [D. 2001]

- Preparation vs. Feedback Phase
- Tangential Predictor in Feedback

with two new building blocks

- For preparation, iterate next problem to convergence via IP method
- use IP predictor in feedback phase



Summary: six ideas for fast nonlinear MPC

- **simultaneous** optimisation: keep states in problem
- **real-time iteration**: use linearisation in non-converged points
- **fast feedback phase** to avoid delays, and longer preparation phase
- **tangential predictor** by initial value embedding
- **solve full QP** to make predictions across active set changes
- **code generation** to minimise overhead (cf afternoon talk R. Quirynen)

Literature

[16] Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., Allgöwer, F.: Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. J. Proc. Contr. 12(4), 577–585 (2002)

[17] Diehl, M., Findeisen, R., Allgöwer, F.: A stabilizing real-time implementation of nonlinear model predictive control. In: Biegler, L., Ghattas, O., Heinkenschloss, M., Keyes, D., van Bloemen Waanders, S.B. (eds.) Real-Time and Online PDE-Constrained Optimization, pp. 23–52. SIAM, Philadelphia (2007)

[18] Diehl, M., Findeisen, R., Allgöwer, F., Bock, H.G., Schlöder, J.P.: Nominal stability of the real-time iteration scheme for nonlinear model predictive control. IEE Proc.-Control Theory Appl. 152(3), 296–308 (2005)

[40] Li, W.C., Biegler, L.T.: Multistep, Newton-type control strategies for constrained nonlinear processes. Chem. Eng. Res. Des. 67, 562–577 (1989)

[41] Li, W.C., Biegler, L.T.: Newton-type controllers for constrained nonlinear processes with uncertainty. Industrial and Engineering Chemistry Research 29, 1647–1657 (1990)

[45] Ohtsuka, T.: A continuation/gmres method for fast computation of nonlinear receding horizon control. Automatica 40(4), 563–574 (2004)

Overview Paper

Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation

Moritz Diehl, Hans Joachim Ferreau, and Niels Haverbeke

L. Magni et al. (Eds.): Nonlinear Model Predictive Control, LNCIS 384, pp. 391–417.
springerlink.com