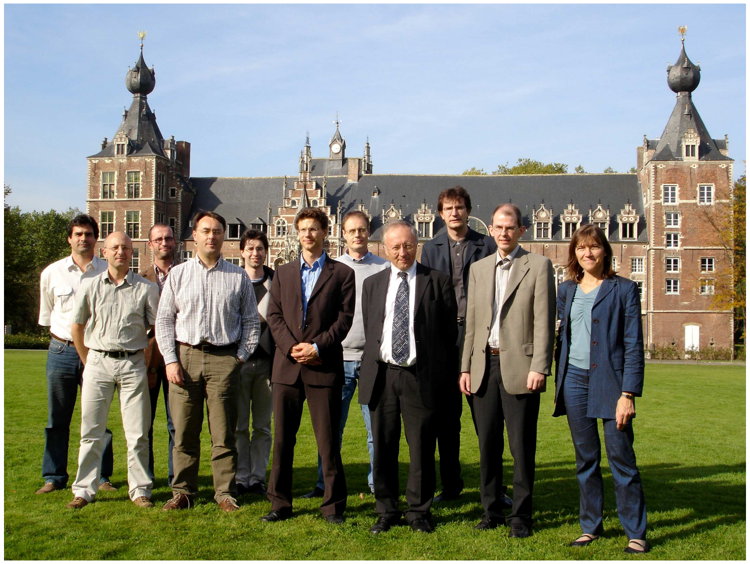
Real-Time Optimization for Fast Nonlinear MPC: Algorithms, Theory, and Applications

Moritz Diehl

Optimization in Engineering Center OPTEC & ESAT, K.U. Leuven



Joint work with H. J. Ferreau, B. Houska*, D. Verscheure*, L. Van den Broeck*, N. Haverbeke*, J. Swevers*, J. De Schutter*, S. Boyd**, H. G. Bock***,*

**K.U. Leuven / **Stanford University / ***University of Heidelberg*

DYSCO Study Day, Mons, May 28, 2009

Topics of this talk

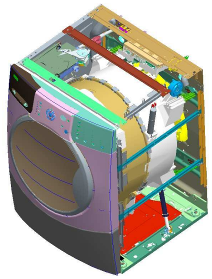
Nonlinear **DY**namic **S**ystems

Model Predictive **C**ontrol

Real-Time **O**ptimization

Dynamic Optimization Applications at OPTEC

Distillation column (Stuttgart; Leuven)

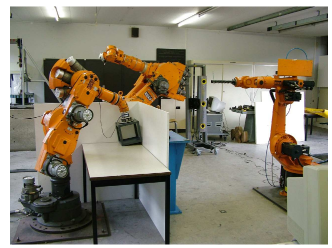


Washing machine design and control (with Bauknecht)

Chemical Process Control (with IPCOS)

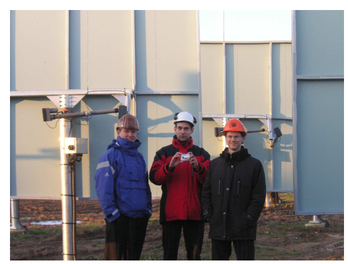
Combine Harvester Control (with Mebios/CNH)

Walking Robots (with Grenoble/Tsukuba)

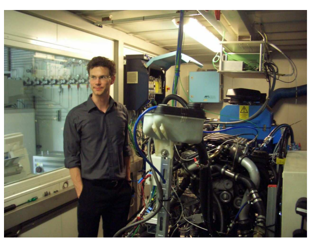


Robot arm time optimal motion (Leuven)

Power generating kites (Leuven)



Solar Thermal Power Plant (Jülich)

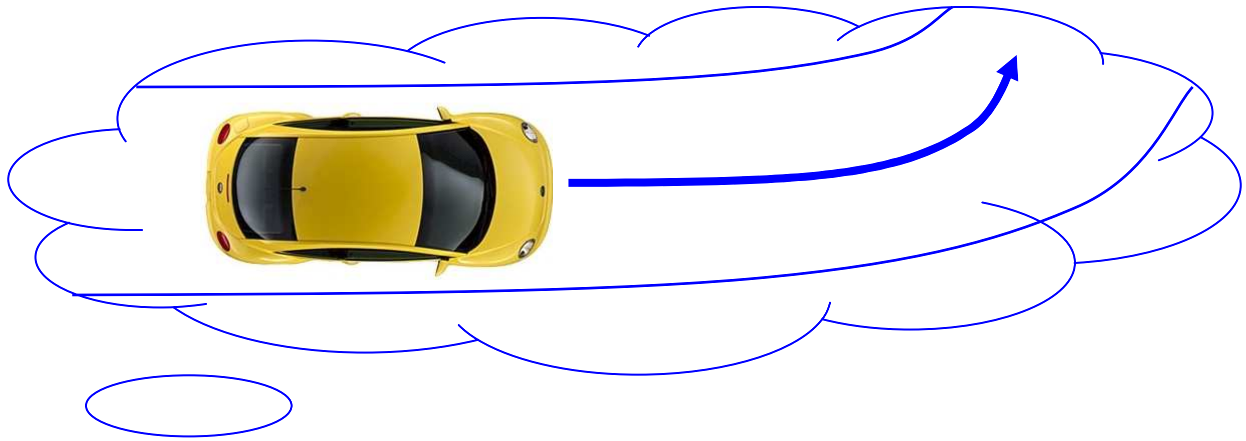


Combustion Engines (Leuven; Linz; Hoerbiger, US)

Nonlinear, constrained...how to control optimally?

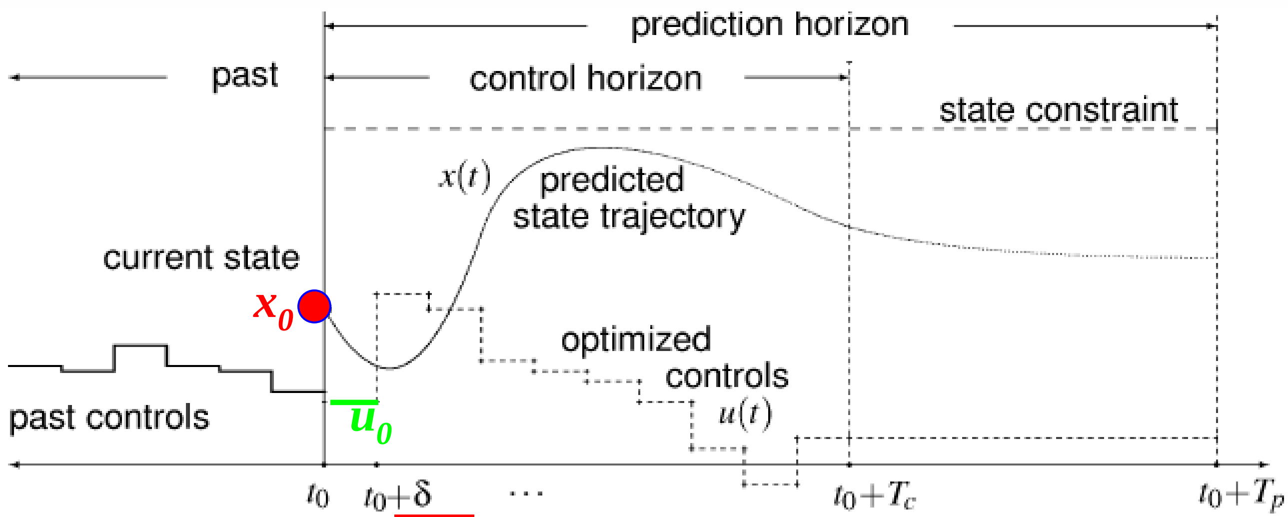
Idea of Model Predictive Control (MPC)

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:

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

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

Outline of the Talk

- Model Predictive Control: A Computational Challenge
- Real-Time Optimization Algorithms:
  - Newton Type Optimization
  - Parametric Sensitivities
- Software and Applications:
  - qpOASES: Predictive Prefilter, Engine Control
  - ACADO: Wind Power Generating Kites
  - TimeOpt: Time Optimal Robot Arm Control

Nonlinear MPC Problem in Discrete Time

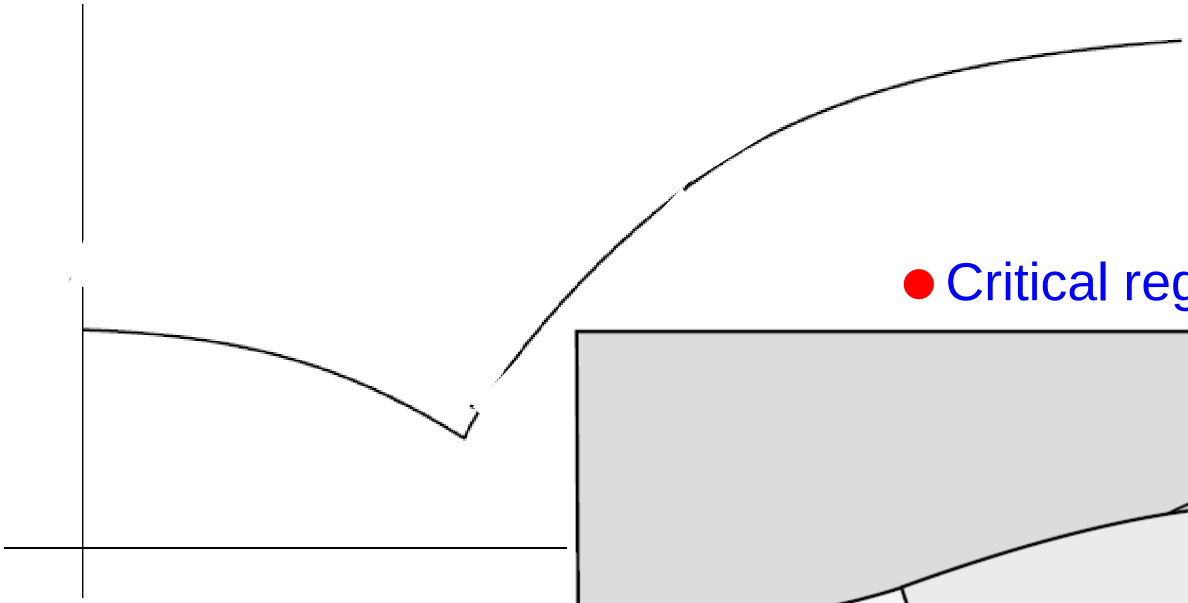
$$
\begin{array}{lrcll}
\underset{x,\,z,\,u}{\text{minimize}} & \displaystyle\sum_{i=0}^{N-1} L_i(x_i, z_i, u_i) & + & E\,(x_N) & \\
\text{subject to} & x_0 - \bar{x}_0 & = & 0, & \\
& x_{i+1} - f_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\
& g_i(x_i, z_i, u_i) & = & 0, & i = 0, \ldots, N-1, \\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

Structured “parametric Nonlinear Program (p-NLP)”

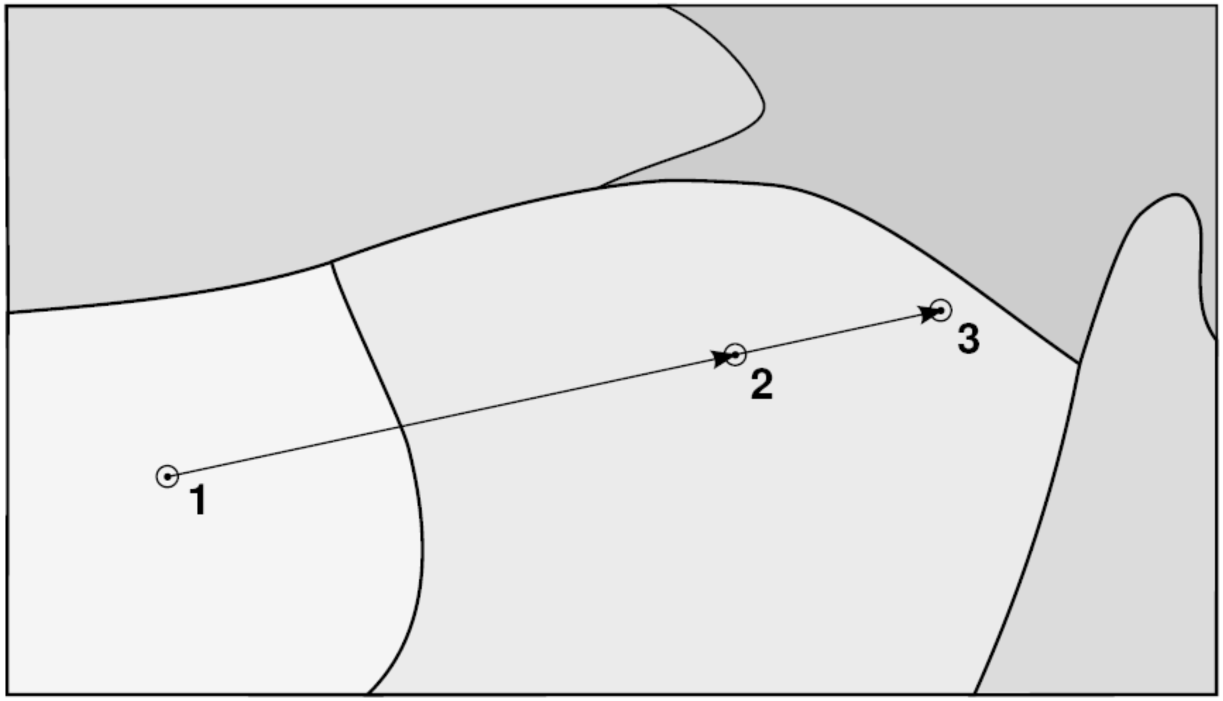
- Initial Value $\bar{x}_0$ is not known beforehand (“online data”)
- Discrete time dynamics often come from ODE simulation cf. “multiple shooting” [Bock & Plitt 1984]
- “Algebraic States” z implicitly defined via third condition, can come from DAEs or from collocation discretization

NMPC = parametric NLP

- Solution manifold is piecewise differentiable (kinks at active set changes)



- Critical regions are non-polyhedral



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

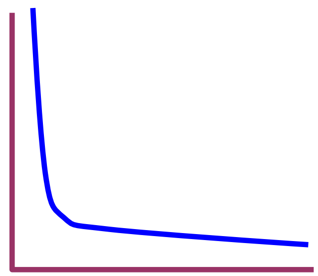
Karush Kuhn Tucker (KKT) conditions: for optimal X^* exist λ^*, μ^* such that:

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

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:



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

Summarize as $R(W) = 0$

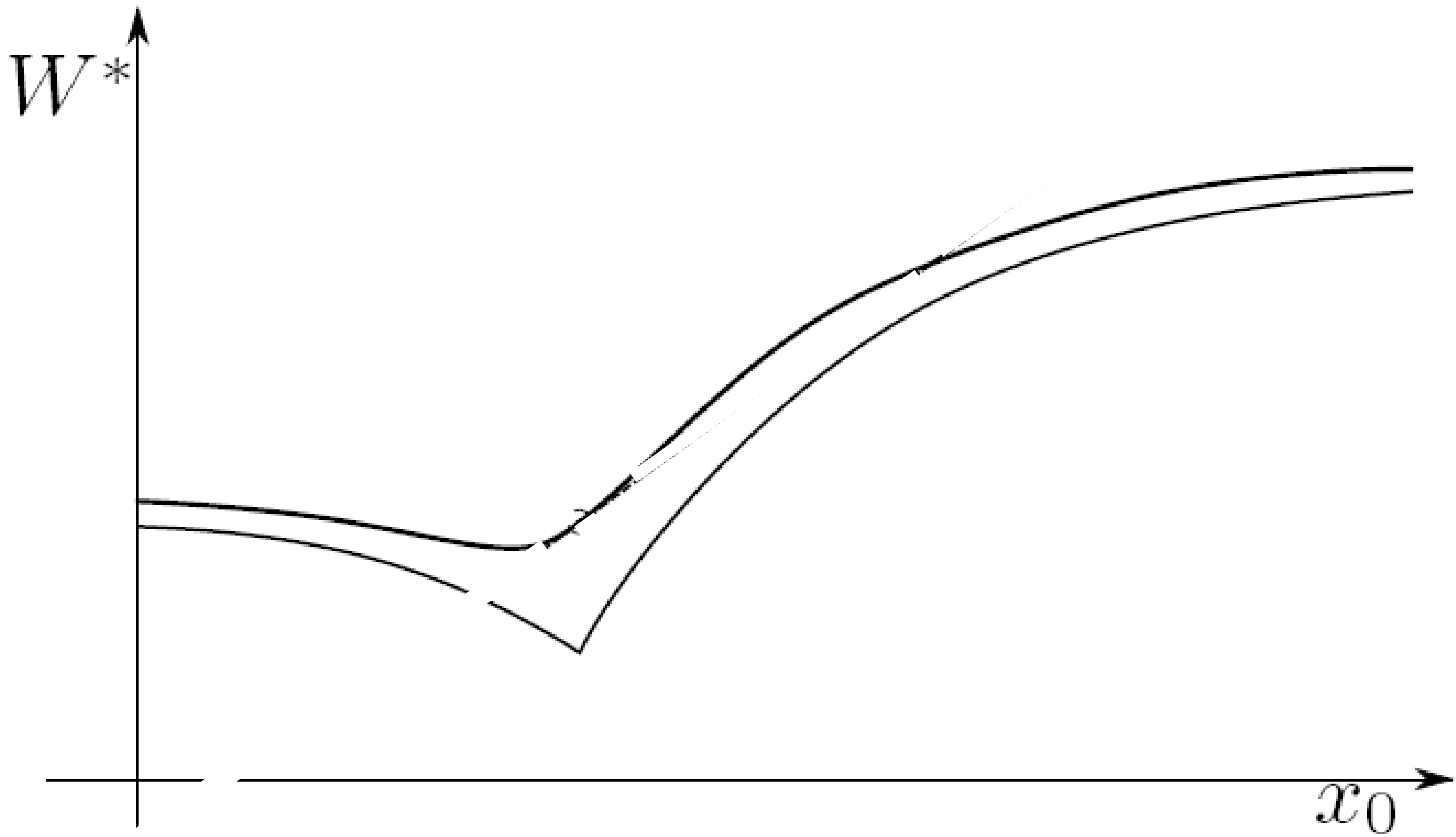
- Solve with Newton's method, i.e.,
  - Linearize at current guess $W^k = (X^k, \lambda^k, \mu^k)$:

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

  - solve linearized system, get new trial point

- For τ small, duality gap becomes provably small. In convex case: self-concordance, polynomial time algorithms, …

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



Approach 2: Sequential Quadratic Programming (SQP)

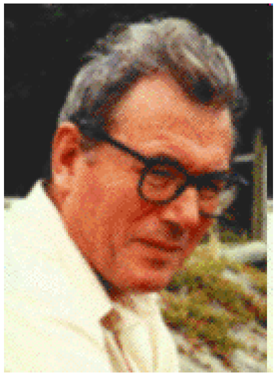
Mathematical Programming 14 (1978) 224–248.

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):

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

with convex quadratic objective using an approximation of (Lagrange-)Hessian.
Obtain new guesses for both X^* and λ^*, μ^* . [cf. Wilson 1963,Robinson 1974]

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.

- But:
  - SQP solves a QP in each iteration: more expensive
  - SQP quickly identifies active set: fewer iterates

SQP good if problem function evaluations are expensive (shooting methods)
→ can generalize to "Sequential Convex Programming - SCP"

Linear Algebra Issues in Optimal Control

- In each SQP iteration, solve structured QP (after algebraic reduction):

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

How to solve this structured QP?

- A - Condensing: eliminate all states (for short horizons, many states)
- B - Banded Factorization…

B - Block Banded Factorizations

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.
- Niels Haverbeke develops fast Riccati QP solvers: e.g. NMPC with 200 steps, 10 states, 10 controls (6000 x 6000 matrix): 20 ms

Outline of the Talk

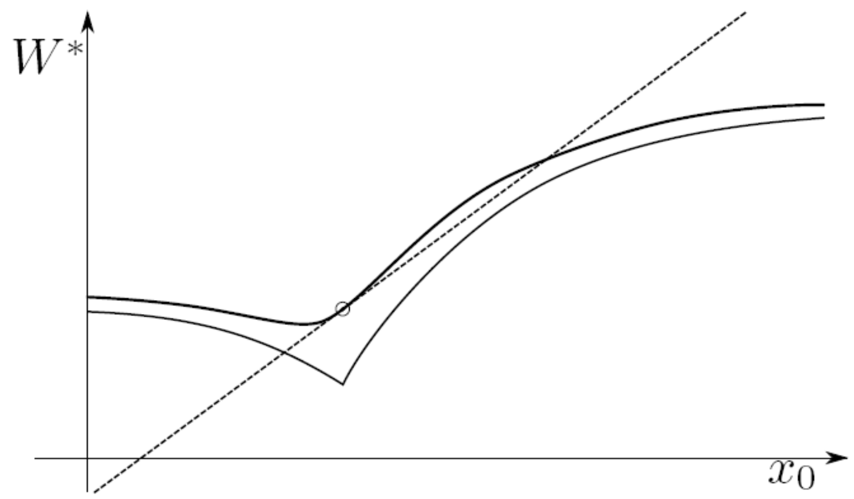
- Model Predictive Control: A Computational Challenge
- Real-Time Optimization Algorithms:
  - Newton Type Optimization
  - Parametric Sensitivities
- Software and Applications:
  - qpOASES: Predictive Prefilter, Engine Control
  - ACADO: Wind Power Generating Kites
  - TimeOpt: Time Optimal Robot Arm Control

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

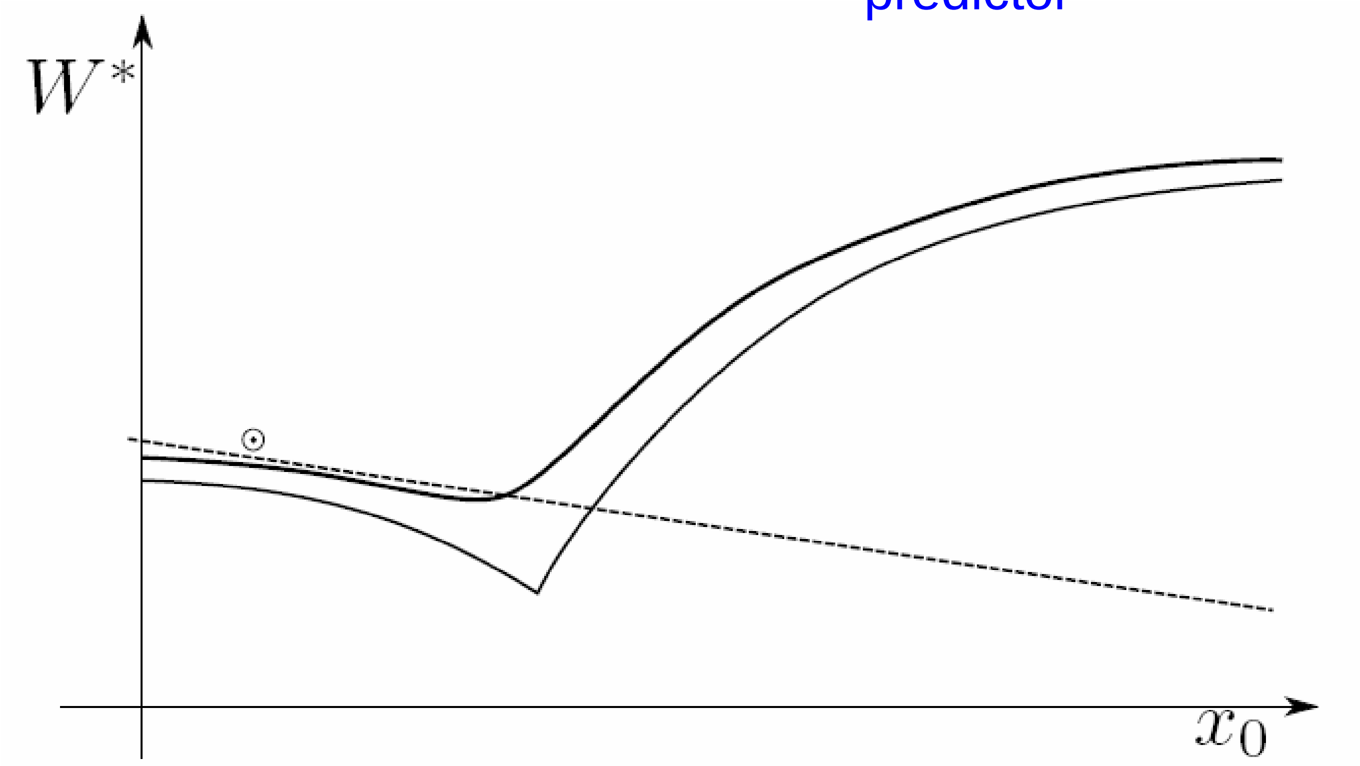
Based on old solution, can get "tangential predictor" for new one:



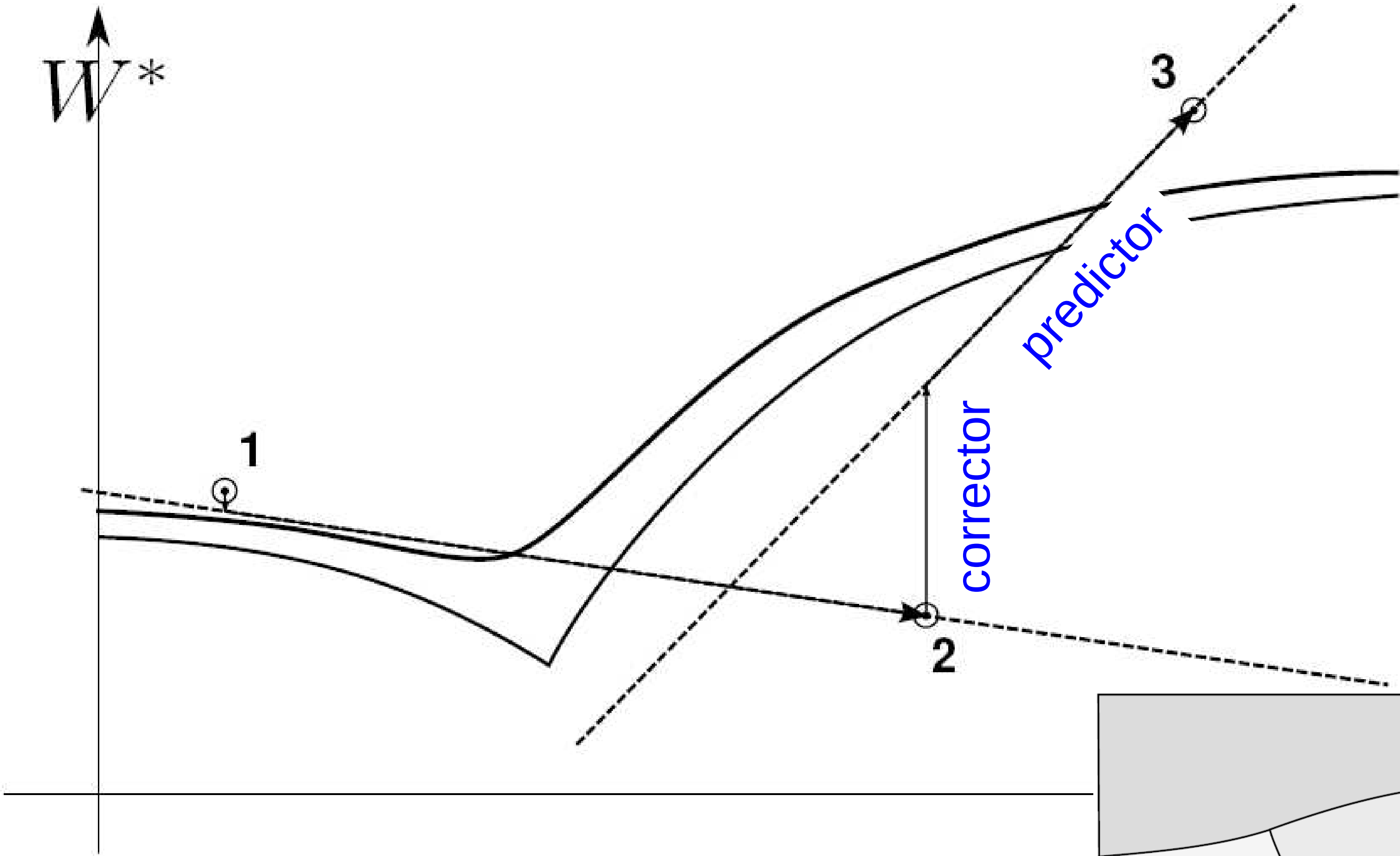
Sensitivity by Newton Predictor-Corrector

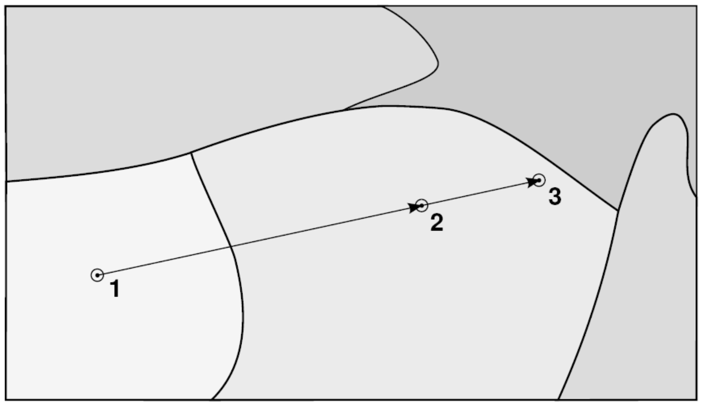
- Can obtain parametric sensitivity for free in Newton type methods:

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



“IP real-time iteration” for sequence of NLPs





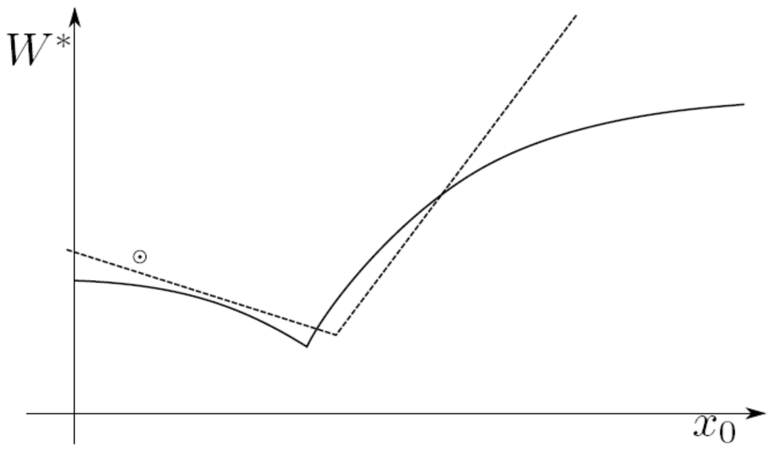
Problem: overshoot at active set changes
Can we do better?

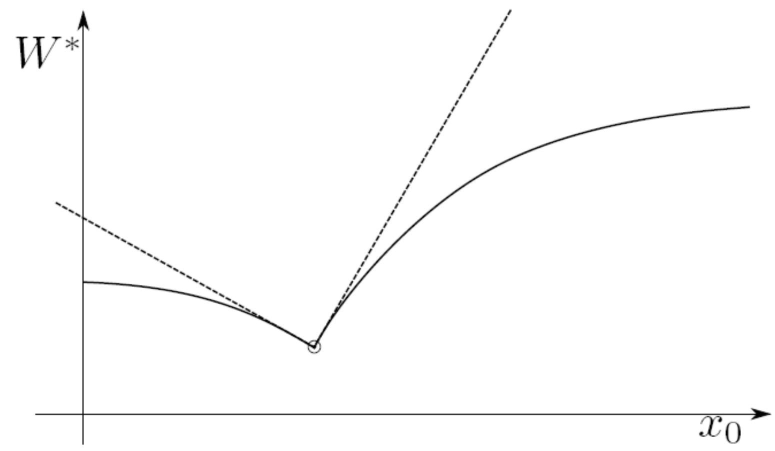
Generalized Tangential Predictor via SQP [D. 2001]

- In each iteration, solve parametric QP with inequalities

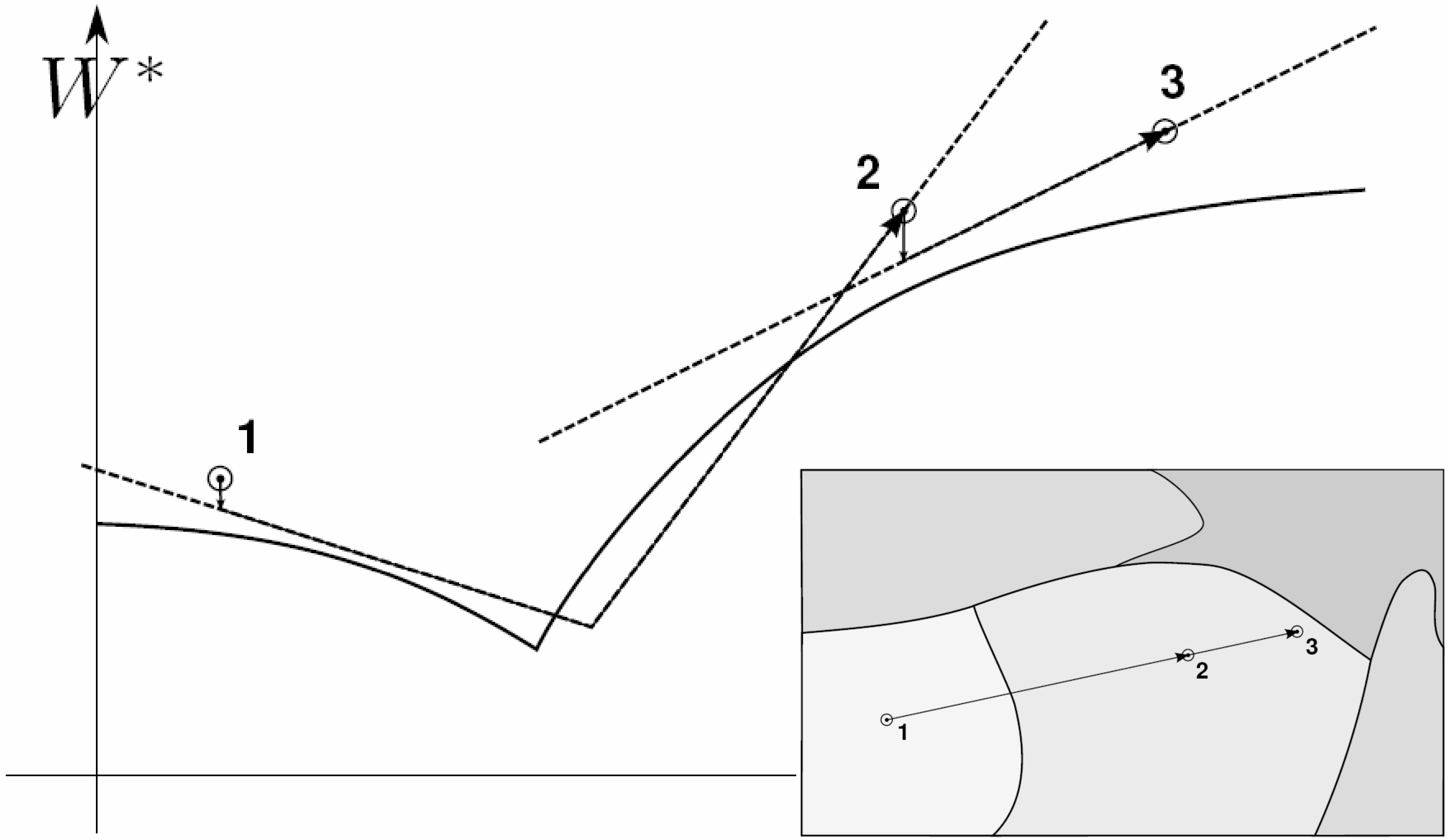
$$\begin{array}{lll} \underset{u}{\text{minimize}} & f_{\text{condQP},i}(\bar{x}_0, u) & \\ \text{subject to} & \bar{r} + \bar{R}^{x_0}\bar{x}_0 + \bar{R}^u u & \leq 0. \end{array}$$

- This "Generalized Tangential Predictor" delivers first order prediction also at active set changes [D. 2001].



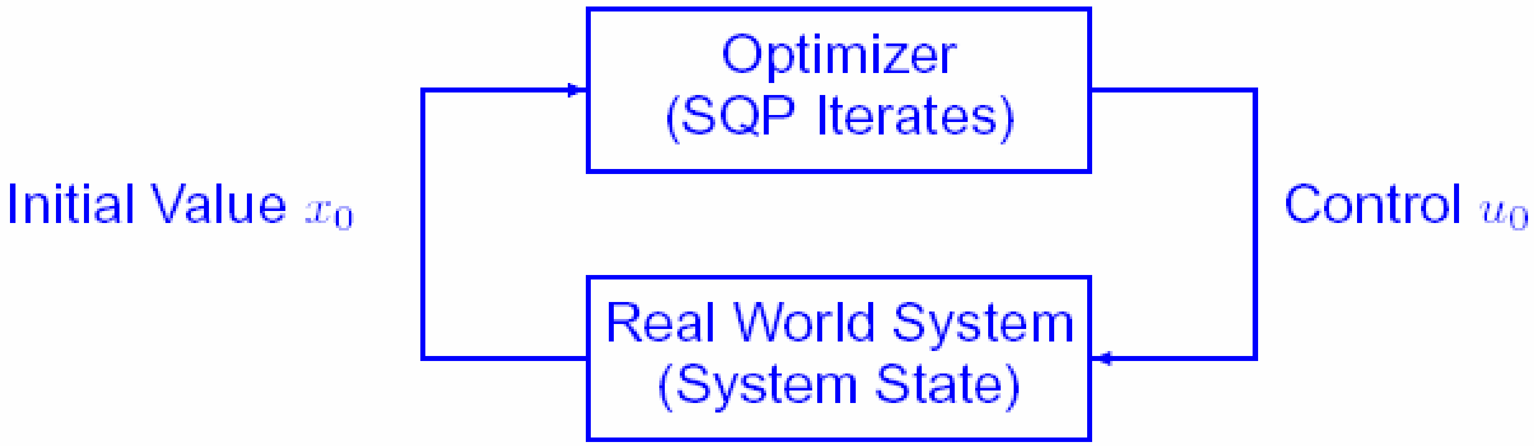


SQP Real-Time Iteration (RTI) [D. 2001]



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

Stability of System-Optimizer Dynamics?

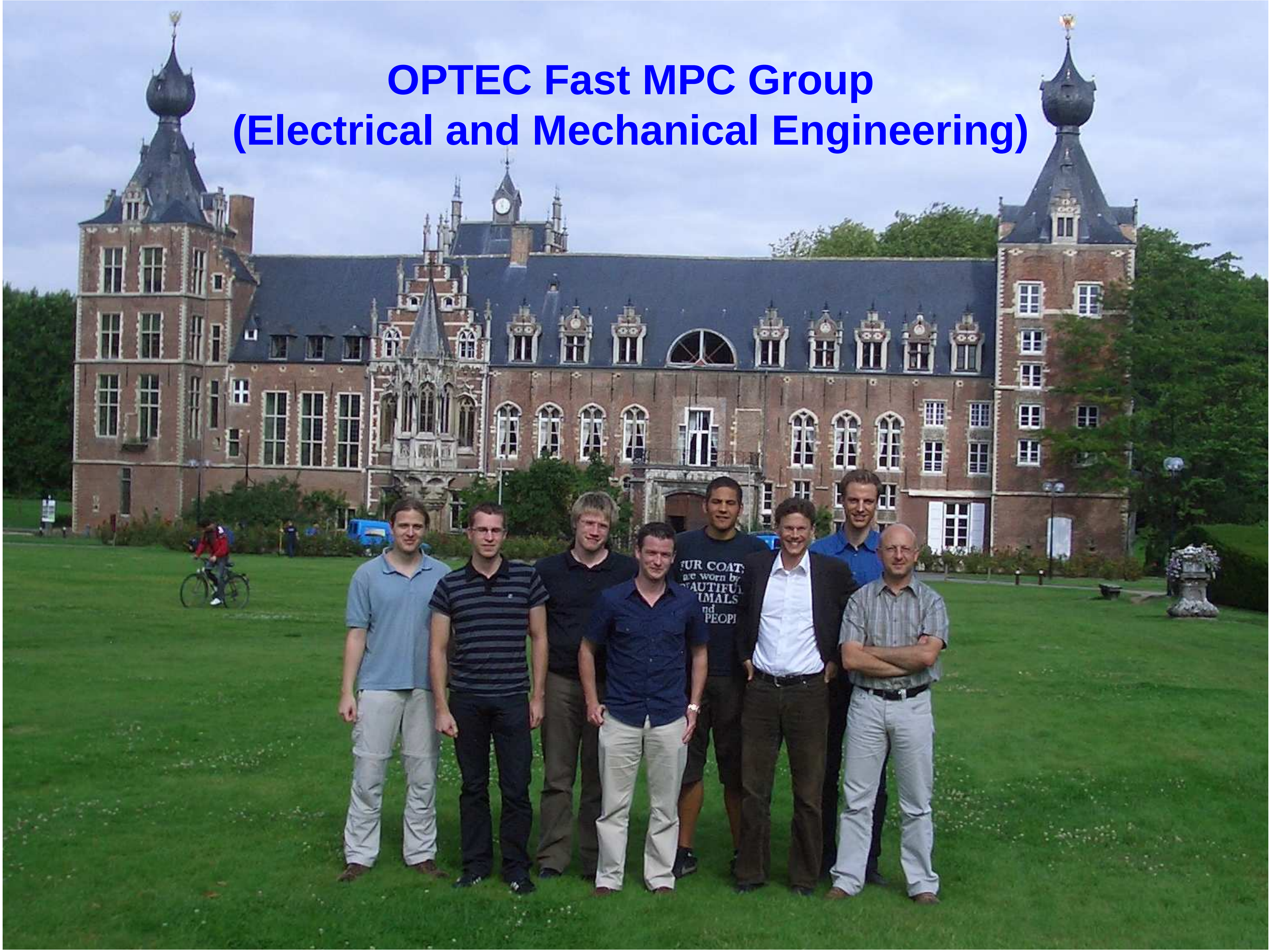


- System and optimizer are coupled: can numerical errors grow and destabilize closed loop?
- Stability analysis combines concepts from both, **NMPC stability theory** and **convergence theory of Newton-type optimization**.
- Stability shown under mild assumptions (short sampling times, stable NMPC scheme)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

Outline of the Talk

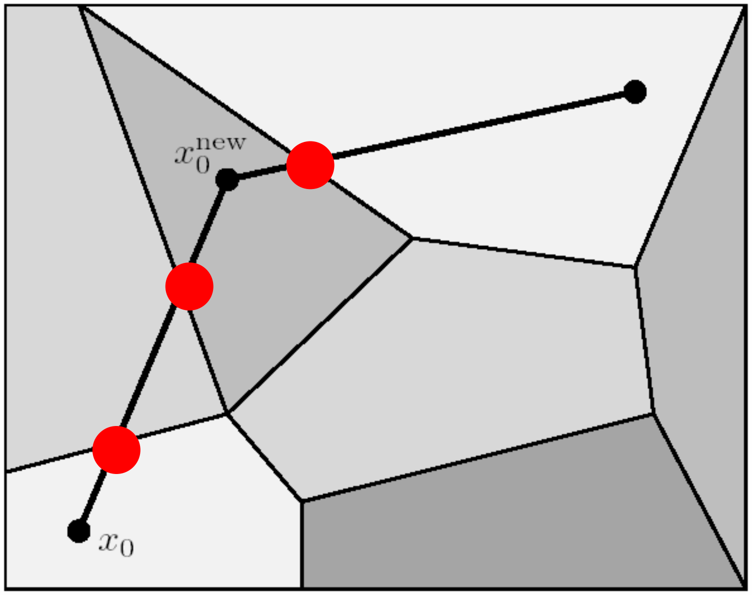
- Model Predictive Control: A Computational Challenge
- Real-Time Optimization Algorithms:
  - Newton Type Optimization
  - Parametric Sensitivities
- Software and Applications:
  - qpOASES: Predictive Prefilter, Engine Control
  - ACADO: Wind Power Generating Kites
  - TimeOpt: Time Optimal Robot Arm Control

OPTEC Fast MPC Group
(Electrical and Mechanical Engineering)



qpOASES: Tailored QP Solver

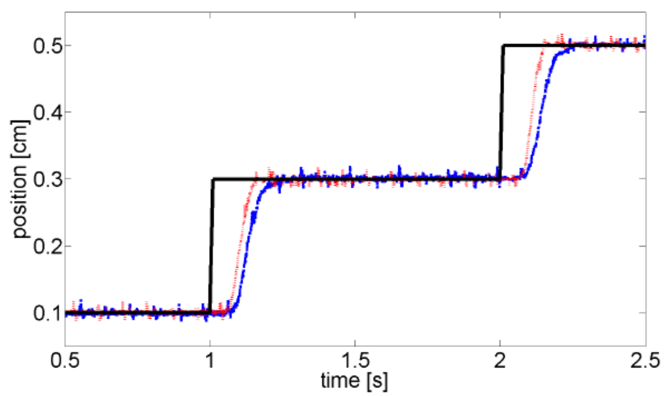
Solve p-QP via „Online Active Set Strategy“:



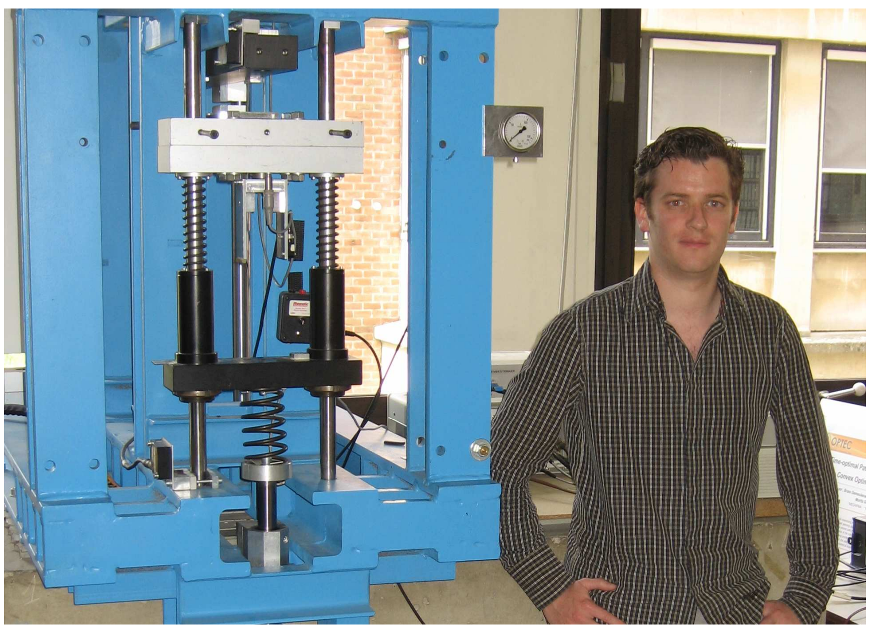
- go on straight line in parameter space from old to new problem data
- **solve each QP on path exactly (keep primal-dual feasibility)**
- Update matrix factorization at boundaries of critical regions
- Up to 10 x faster than standard QP

qpOASES: open source C++ code by Hans Joachim Ferreau

Time Optimal MPC: a 100 Hz Application

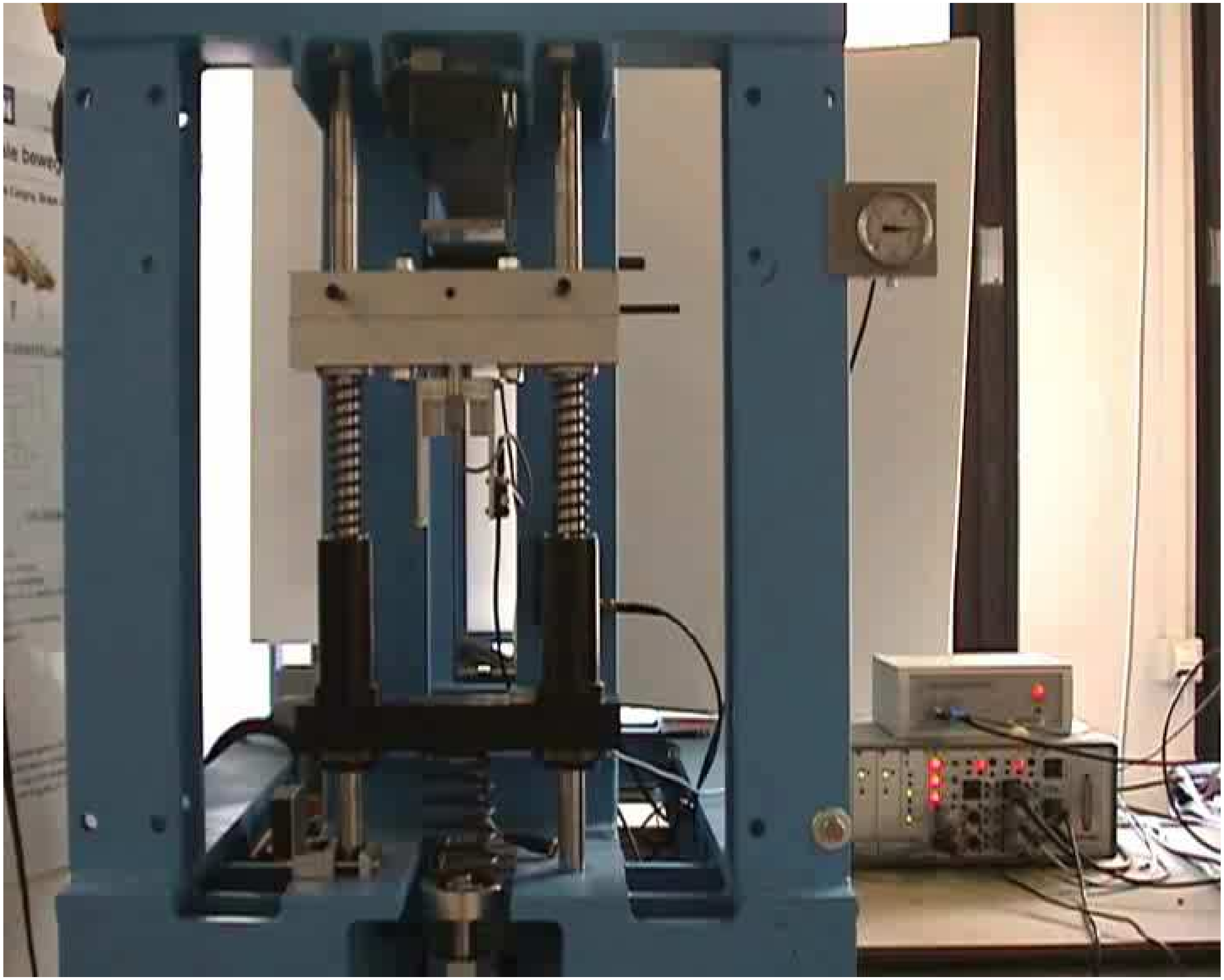


- Quarter car: oscillating spring damper system
- MPC Aim: settle at any new setpoint in ***in minimal time***
- Two level algorithm: MIQP
  - 6 online data
  - 40 variables + one integer
  - 242 constraints (in-&output)
- use qpOASES on dSPACE
- **CPU time: <10 ms**

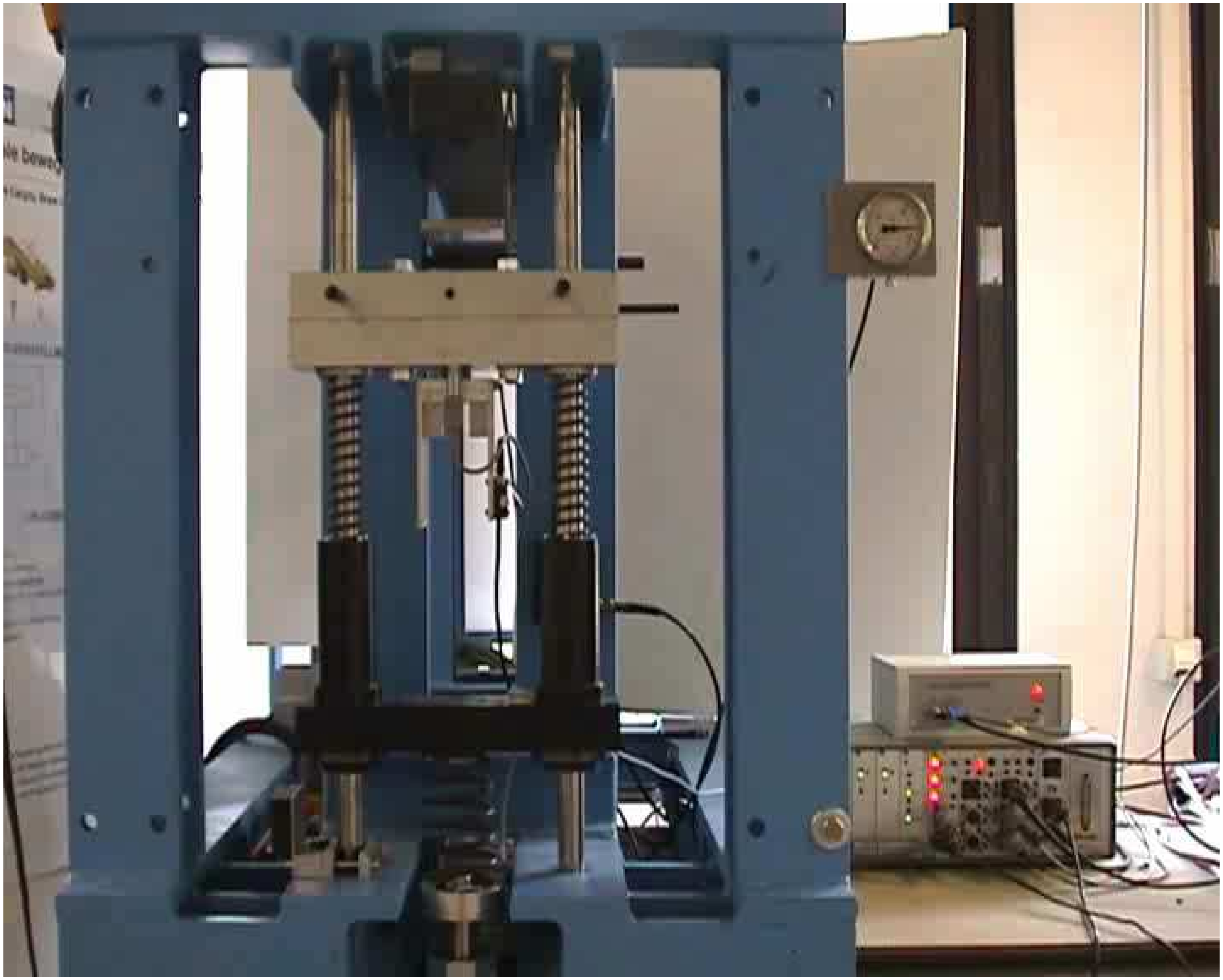


Lieboud Van den Broeck in front of quarter car experiment

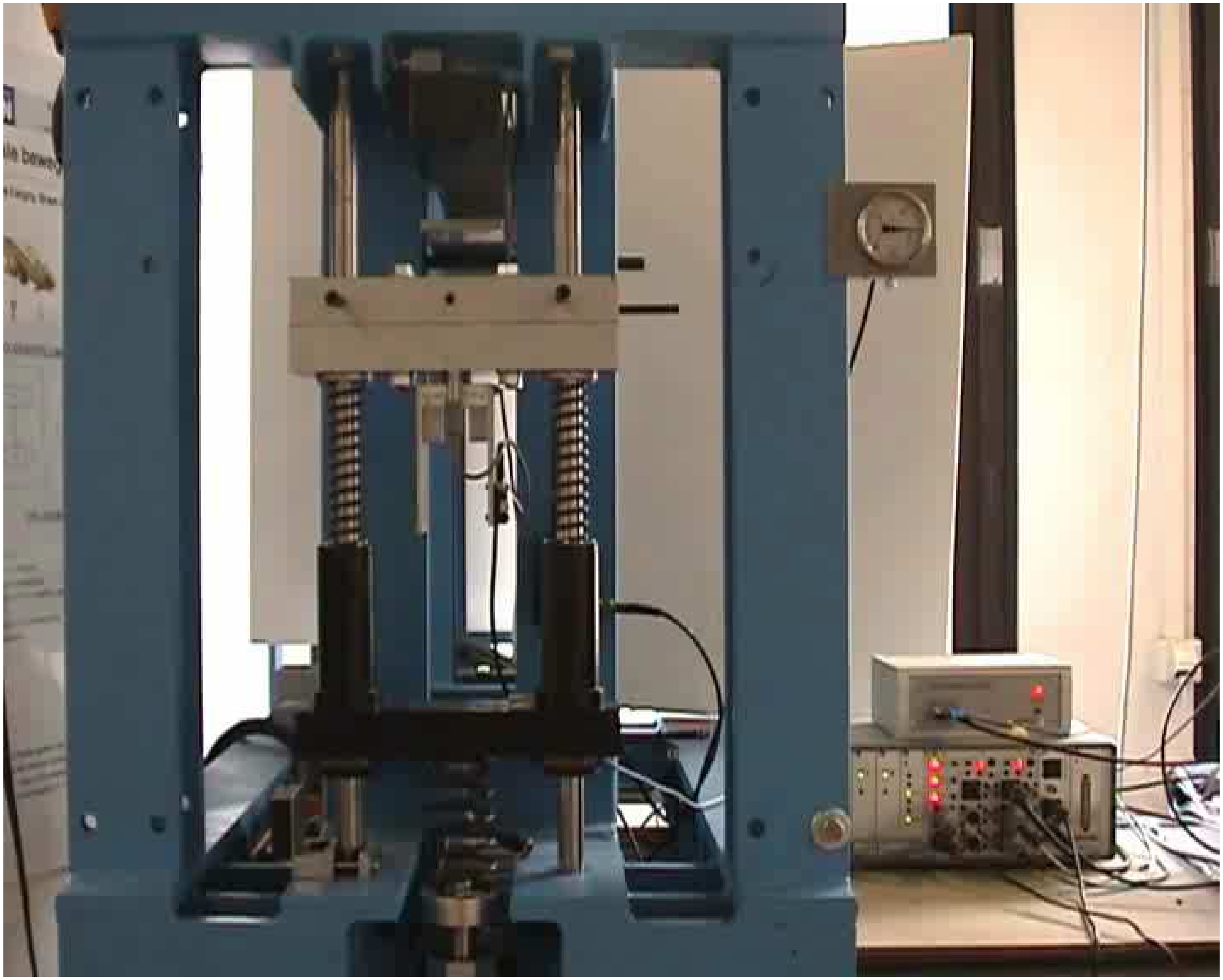
Setpoint change without control: oscillations



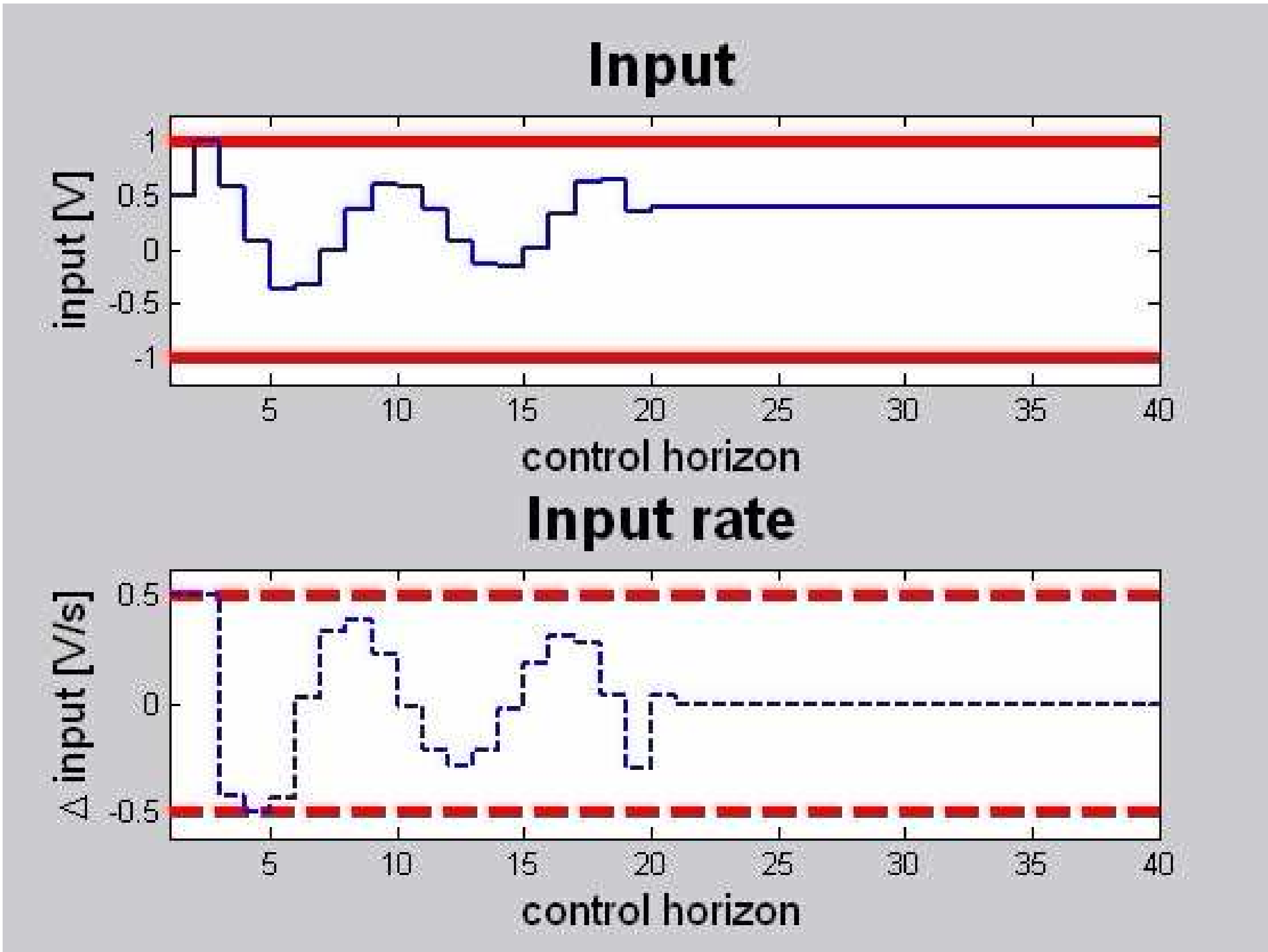
With LQR control: inequalities violated



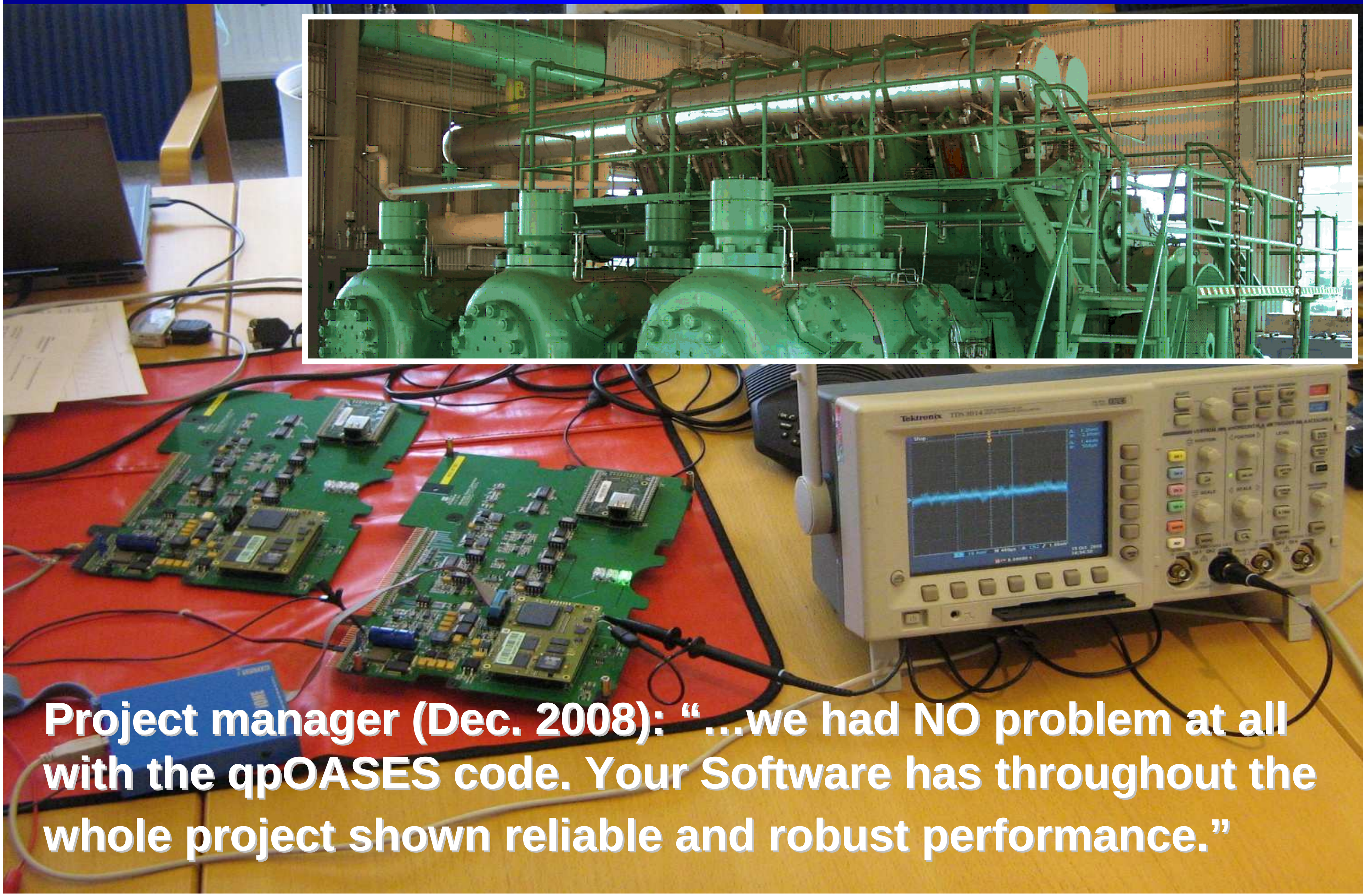
With Time Optimal MPC



Time Optimal MPC: qpOASES Optimizer Contents



qpOASES running on Industrial Control Hardware (20 ms)



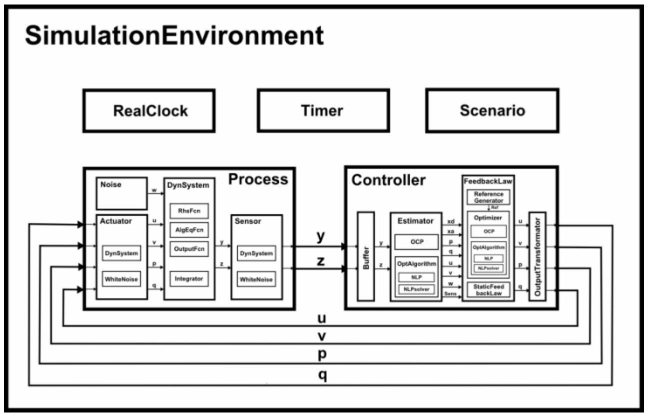
Project manager (Dec. 2008): "…we had NO problem at all with the qpOASES code. Your Software has throughout the whole project shown reliable and robust performance."

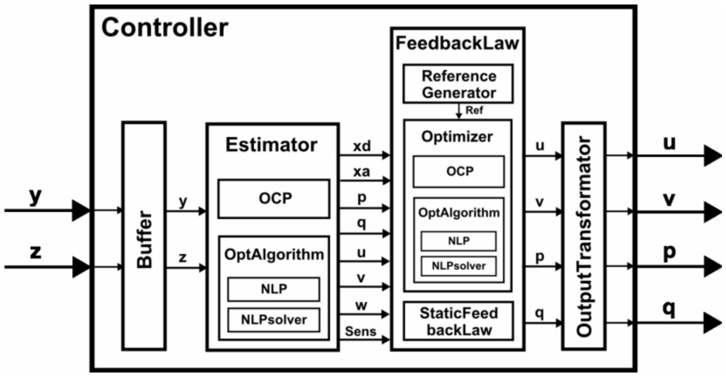
Outline of the Talk

- Model Predictive Control: A Computational Challenge
- Real-Time Optimization Algorithms:
  - Newton Type Optimization
  - Parametric Sensitivities
- Software and Applications:
  - qpOASES: Predictive Prefilter, Engine Control
  - ACADO: Wind Power Generating Kites
  - TimeOpt: Time Optimal Robot Arm Control

ACADO Toolkit

- A Toolkit for „Automatic Control and Dynamic Optimization“





- C++ code along with user-friendly Matlab interfaces
- Open-source software (LGPL 3)
- Since mid 2008 developed at OPTEC by Boris Houska and Hans Joachim Ferreau

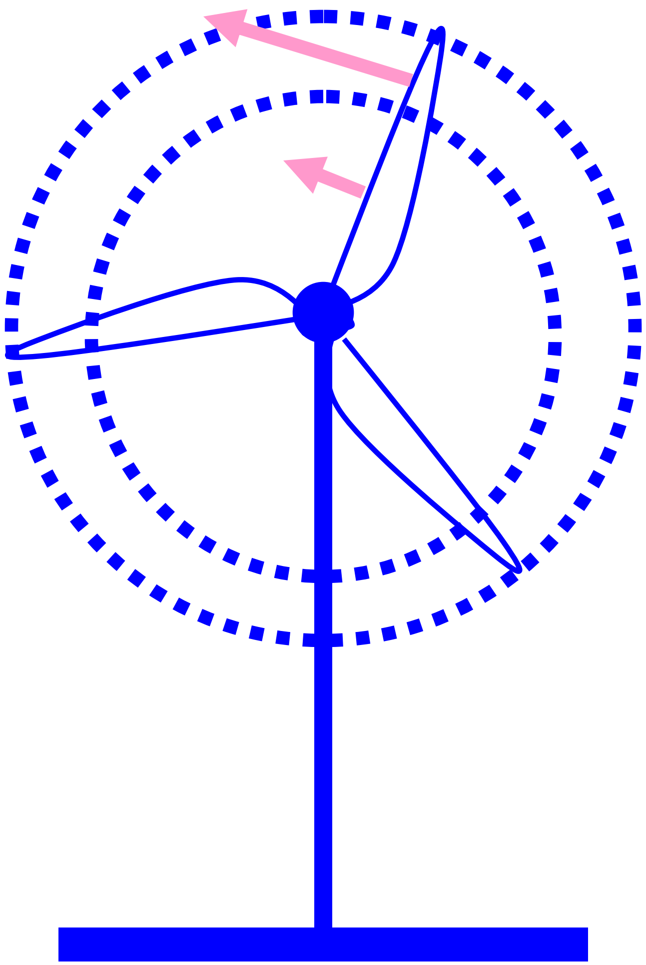
ACADO Toolkit – Main Features

- Problem Classes:
  - Optimal control
  - State & parameter estimation
  - Robust optimization
  - Model predictive control
- Dynamic Optimization:
  - Linear and Nonlinear
  - ODE and DAE
  - Continuous and discrete time
  - Automatic differentiation
  - Convexity detection
- Discretization Methods:
  - Single shooting
  - Multiple shooting
  - Collocation
- Integrators:
  - RKF and BDF methods
  - Efficient sensitivity generation
  - Second order sensitivities
- NLP solvers:
  - Adjoint-based SQP
  - Interior point methods

First beta release available since today on www.acadotoolkit.org
login "DYSCO", password: ask H. Joachim Ferreau at ACADO poster

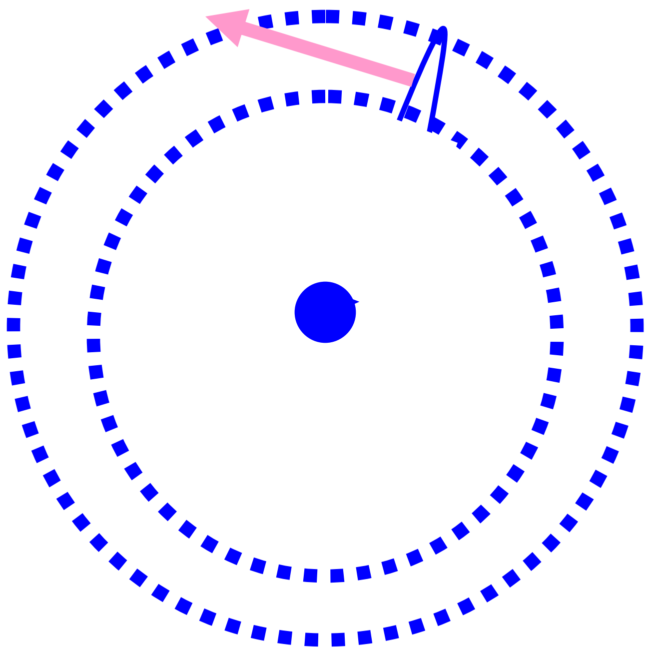
NMPC of Wind Power Generating Kites

Conventional Wind Turbines



- Due to high speed, wing tips are ***most efficient*** part of wing

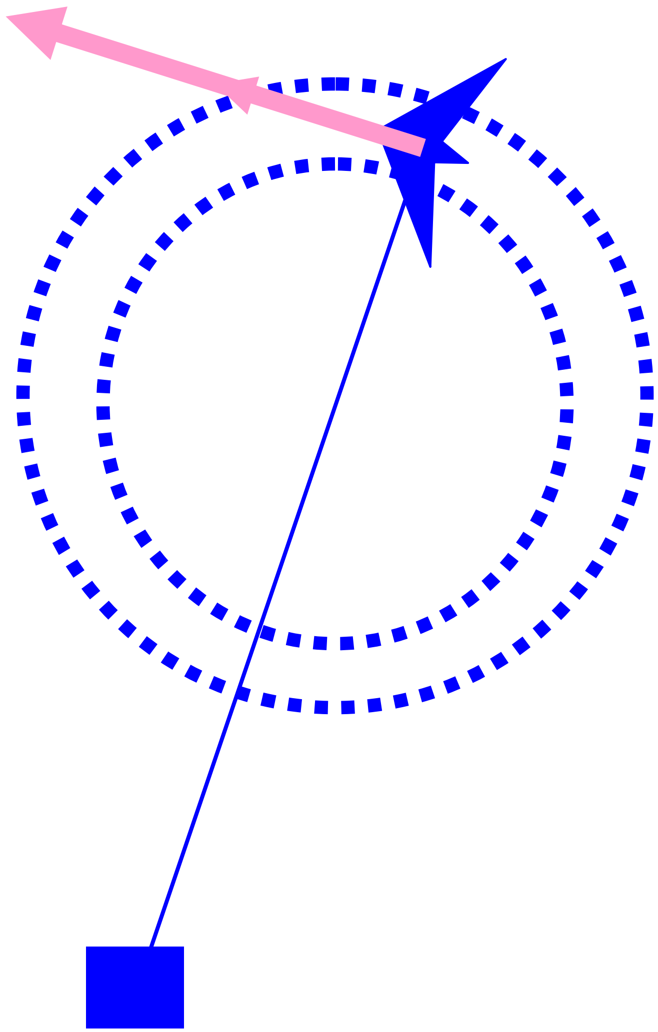
Conventional Wind Turbines



- Due to high speed, wing tips are ***most efficient*** part of wing

Could we construct a wind turbine with only wing tips and generator?

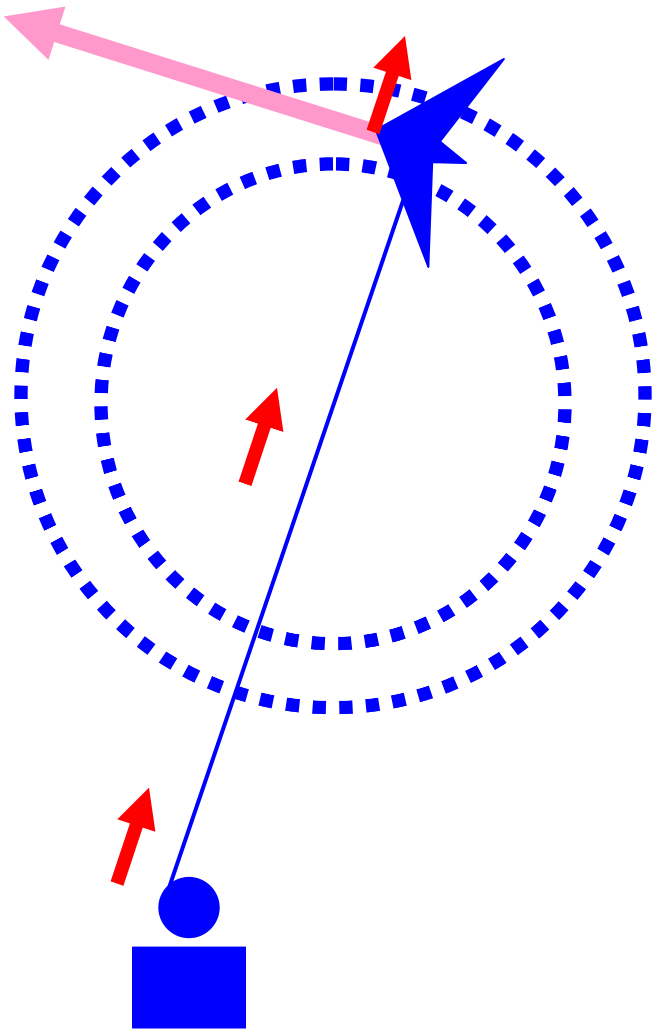
Crosswind Kite Power



- Fly kite fast in crosswind direction
- Very strong force

But where could a generator be driven?

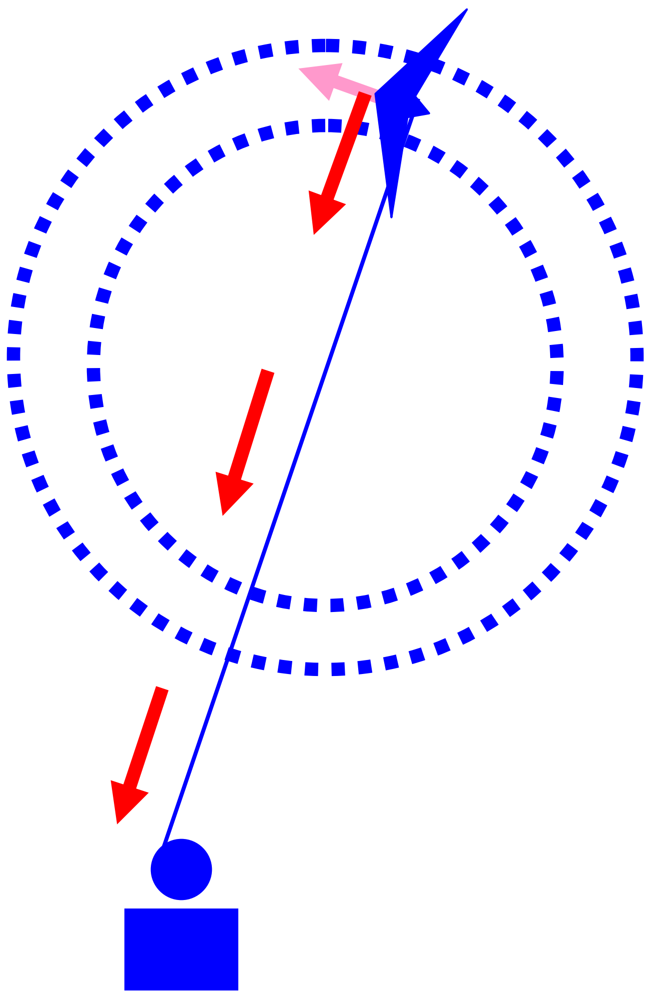
New Power Generating Cycle



New cycle consists of two phases:

- ***Power generation phase:***
  - ***unwind cable***
  - generate power

New Power Generating Cycle

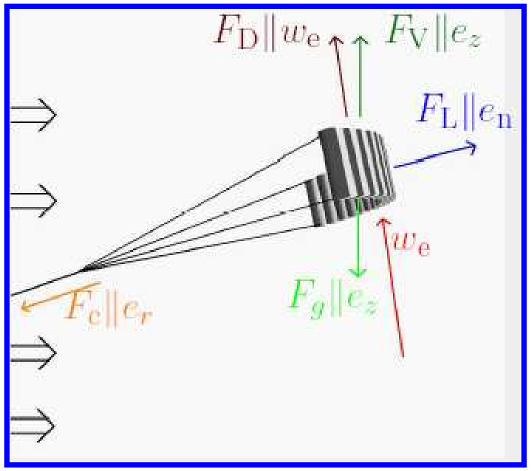


New cycle consists of two phases:

- ***Power generation phase:***
  - ***unwind cable***
  - generate power
- ***Retraction phase:***
  - Reduce tension
  - pull back line

Kite Modelling (Boris Houska)

Have to regard also cable elasticity



forces at kite

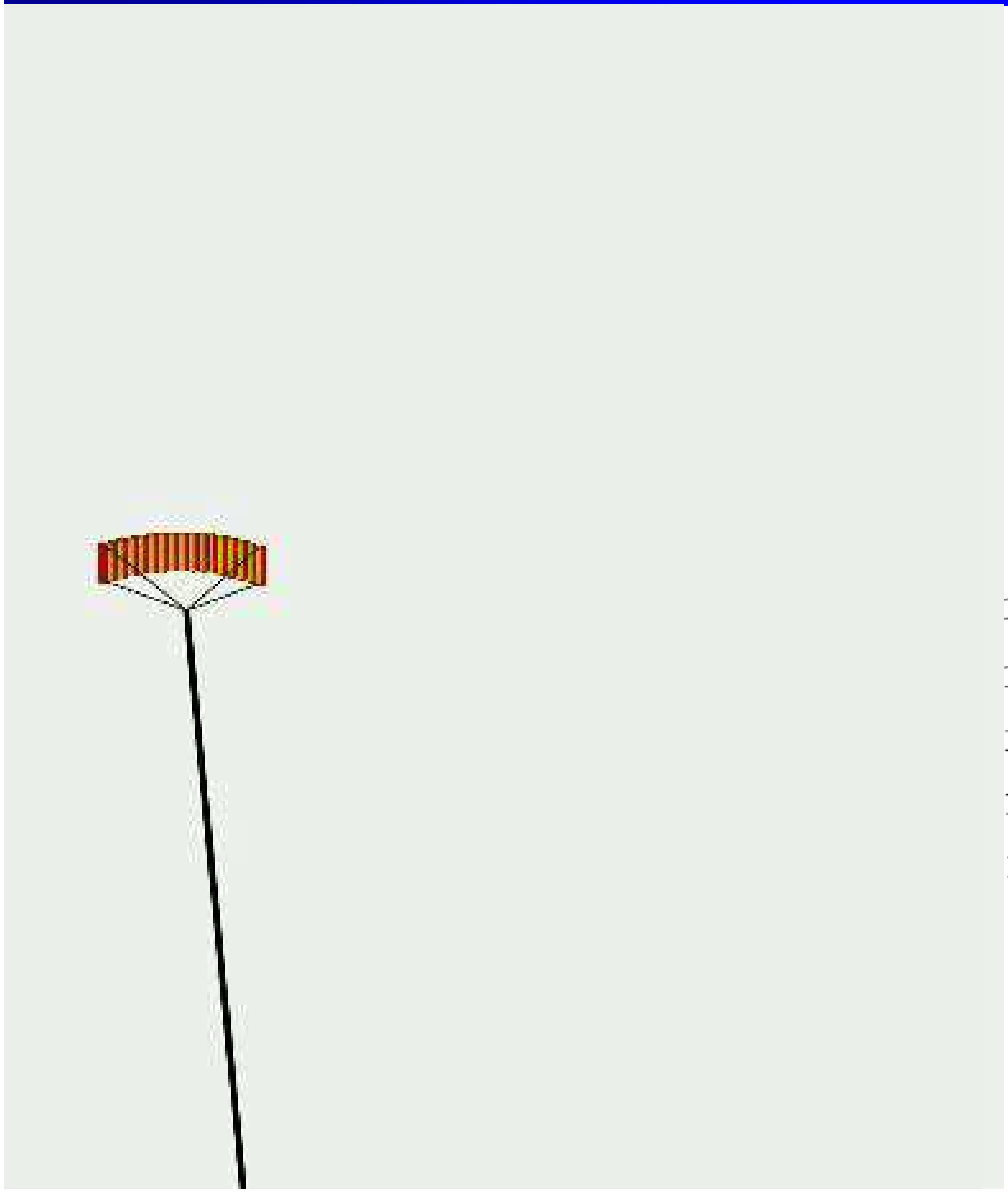
ODE Model with 12 states and 3 controls

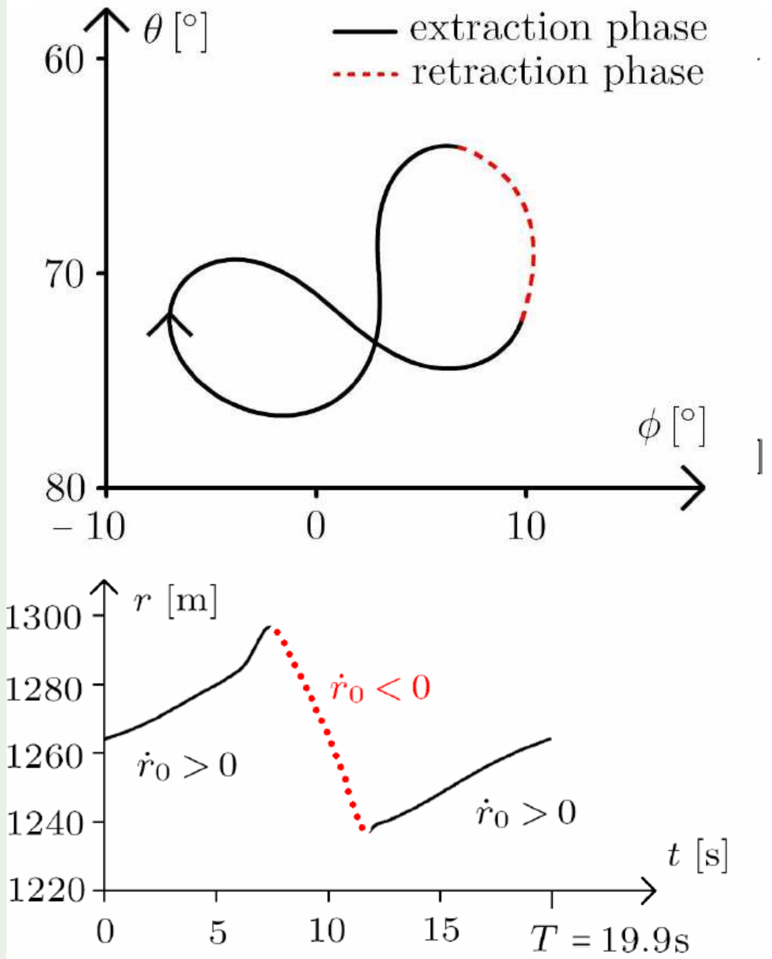
- Differential states:
  $x := (r_0\,,\, r\,,\, \phi\,,\, \theta\,,\, \dot{r}_0\,,\, \dot{r}\,,\, \dot{\phi}\,,\, \dot{\theta}\,,\, n\,,\, \Psi\,,\, C_{\mathrm{L}}\,,\, W\,)^T$
- Controls: $u := (\ddot{r}_0\,,\, \dot{\Psi}\,,\, \dot{C}_{\mathrm{L}})^T$

Control inputs:

- line length
- roll angle (as for toy kites)
- lift coefficient (pitch angle)

Solution of Periodic Optimization Problem





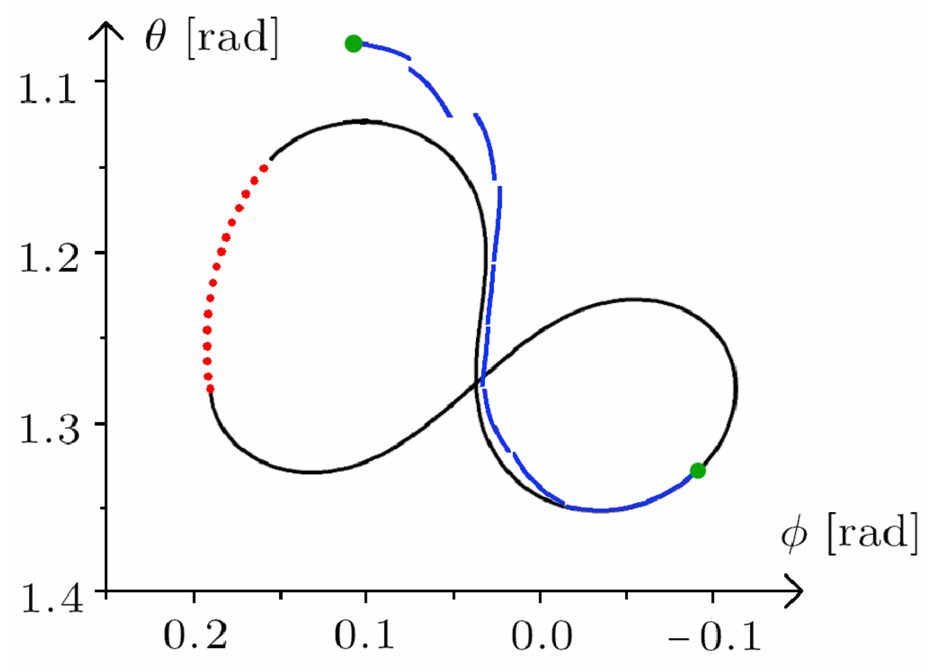
Cable 1.3 km long, 7 cm thick,

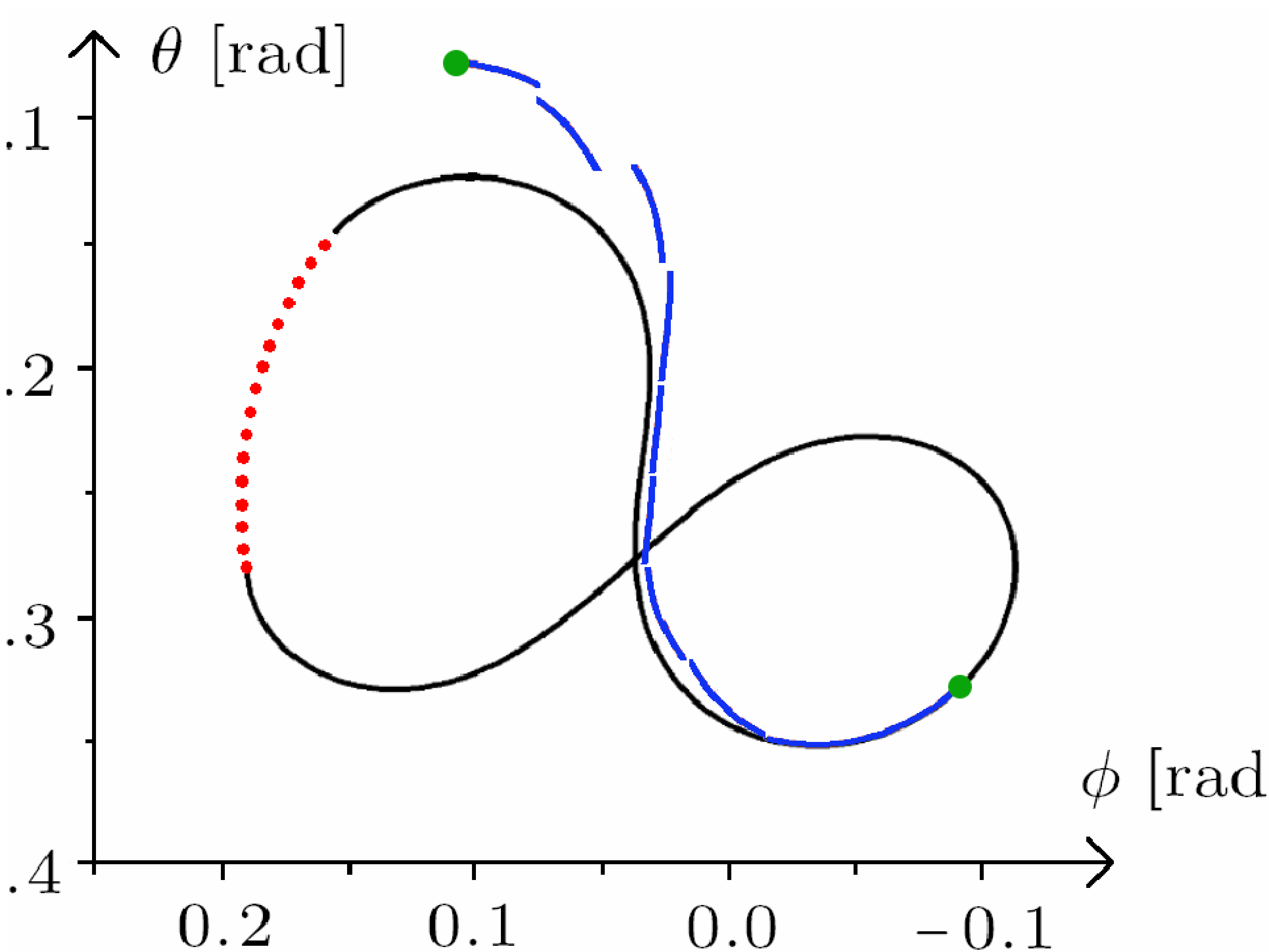
Kite Area 500 m^2, Power 5 MW.

Kite NMPC Problem solved with ACADO

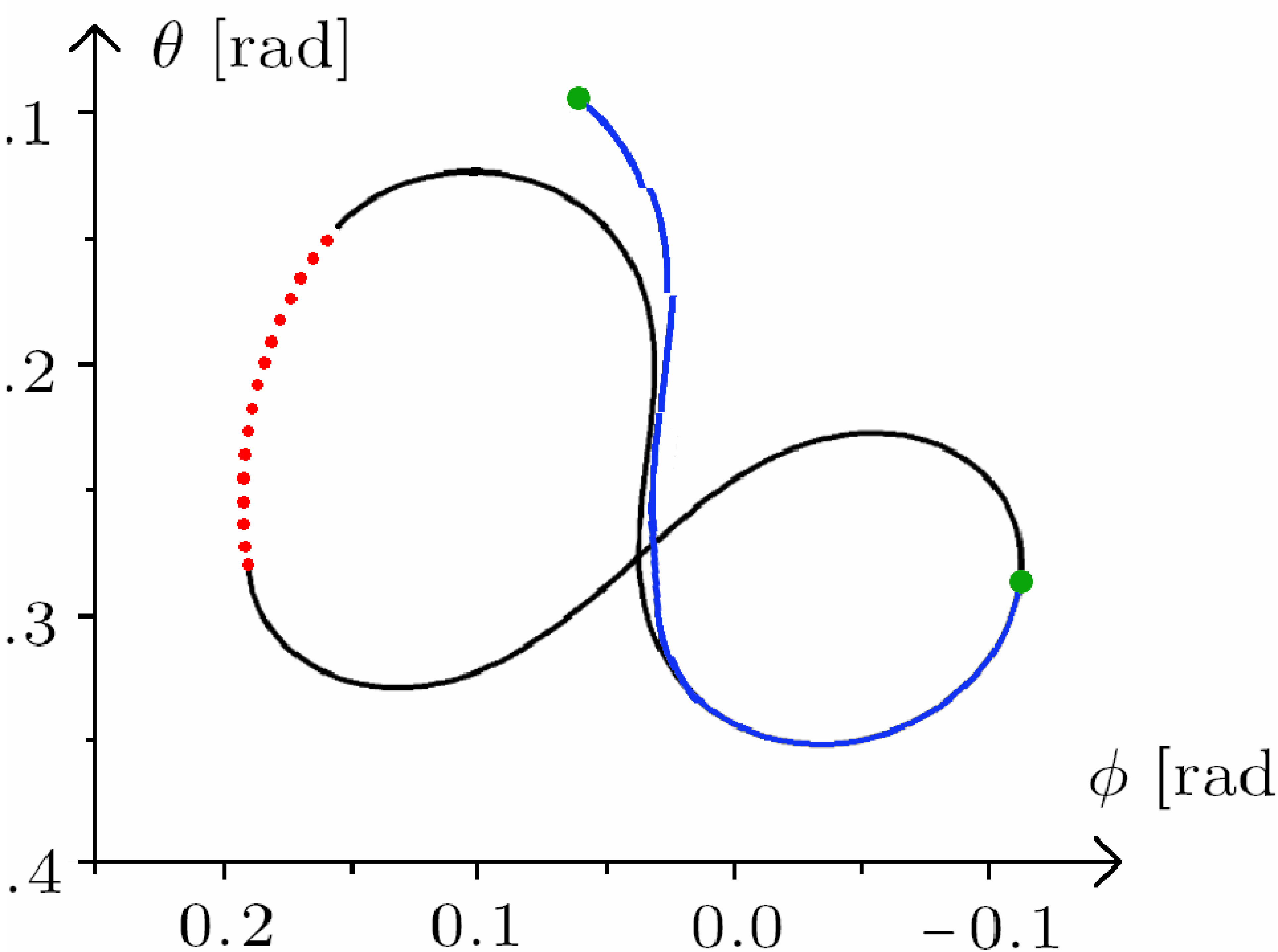
- 9 states, 3 controls
- Penalize deviation from “lying eight”
- Predict half period
- zero terminal constraint
- 10 multiple shooting intervals

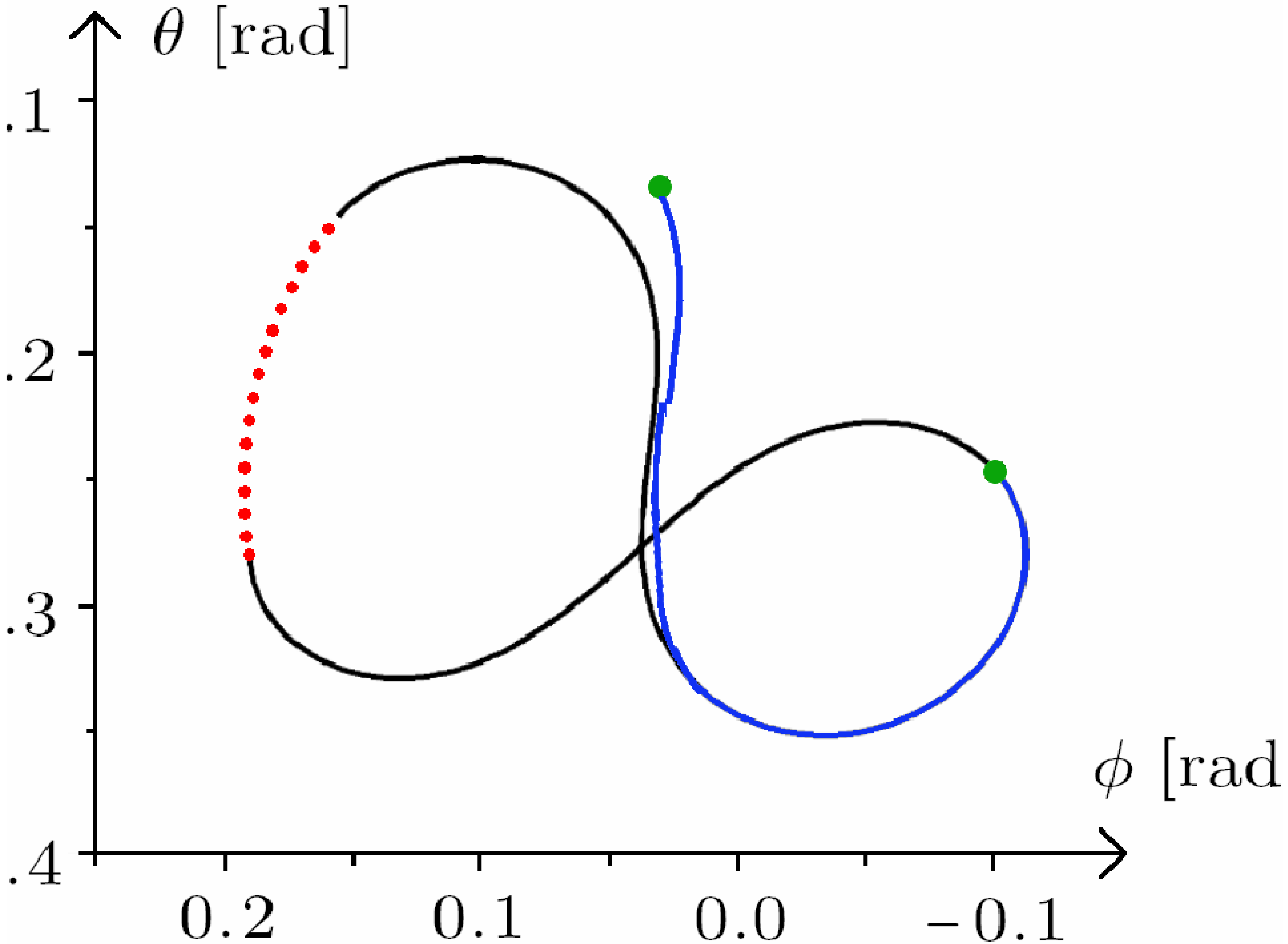
Solve with **SQP real-time iterations**



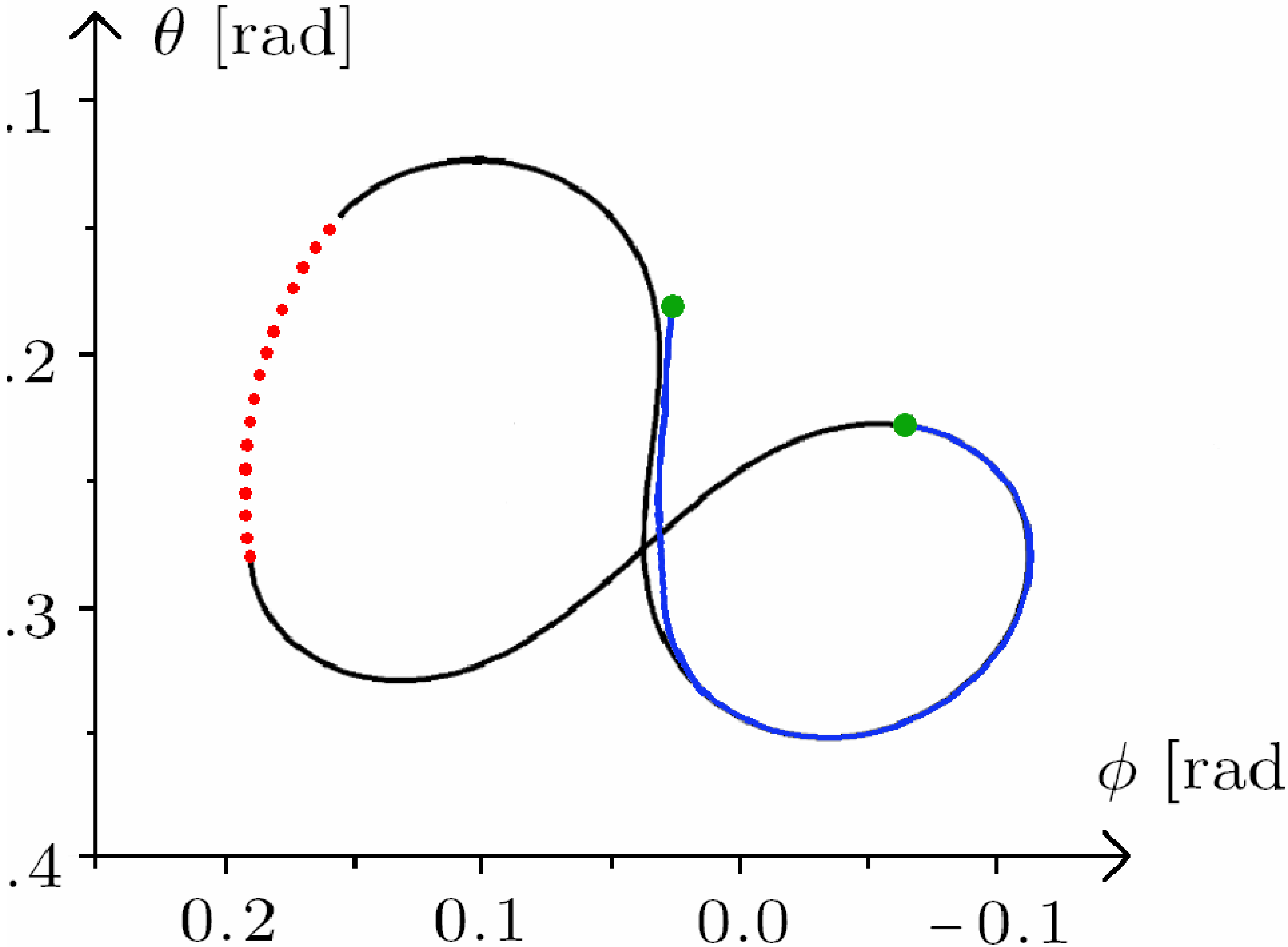


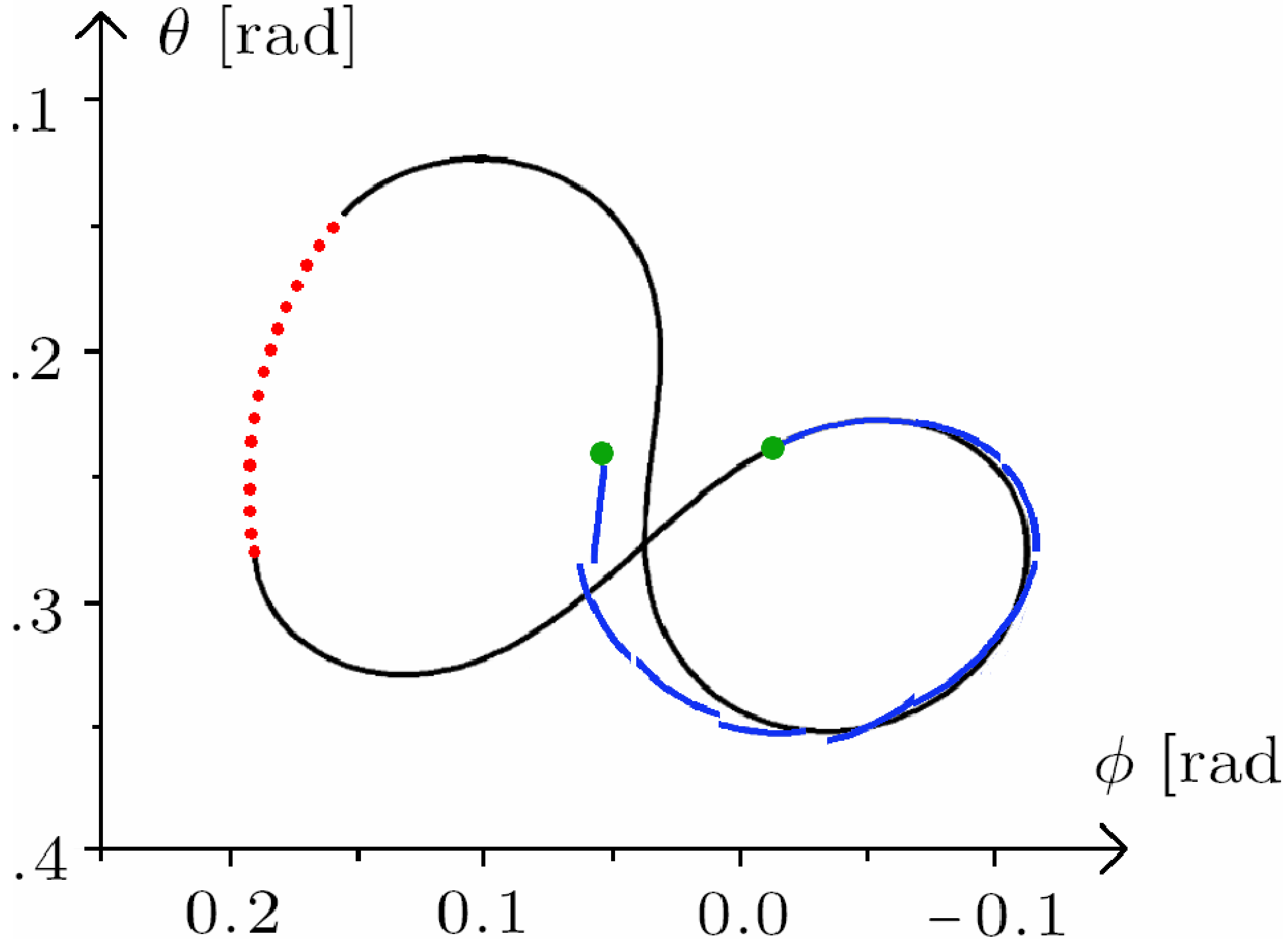
θ [rad]
.1
.2
.3
.4
φ [rad
0.2
0.1
0.0
−0.1

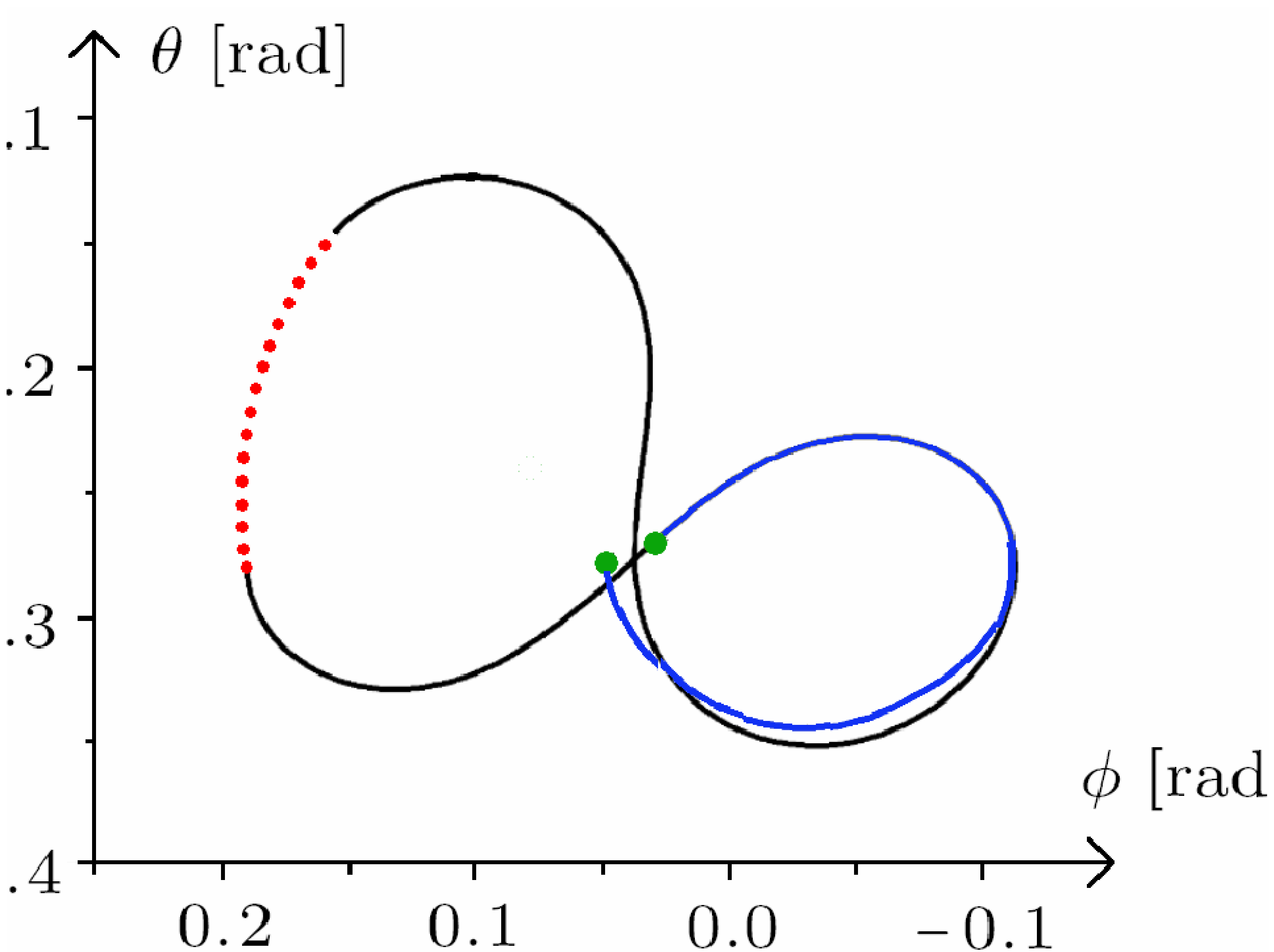

θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1



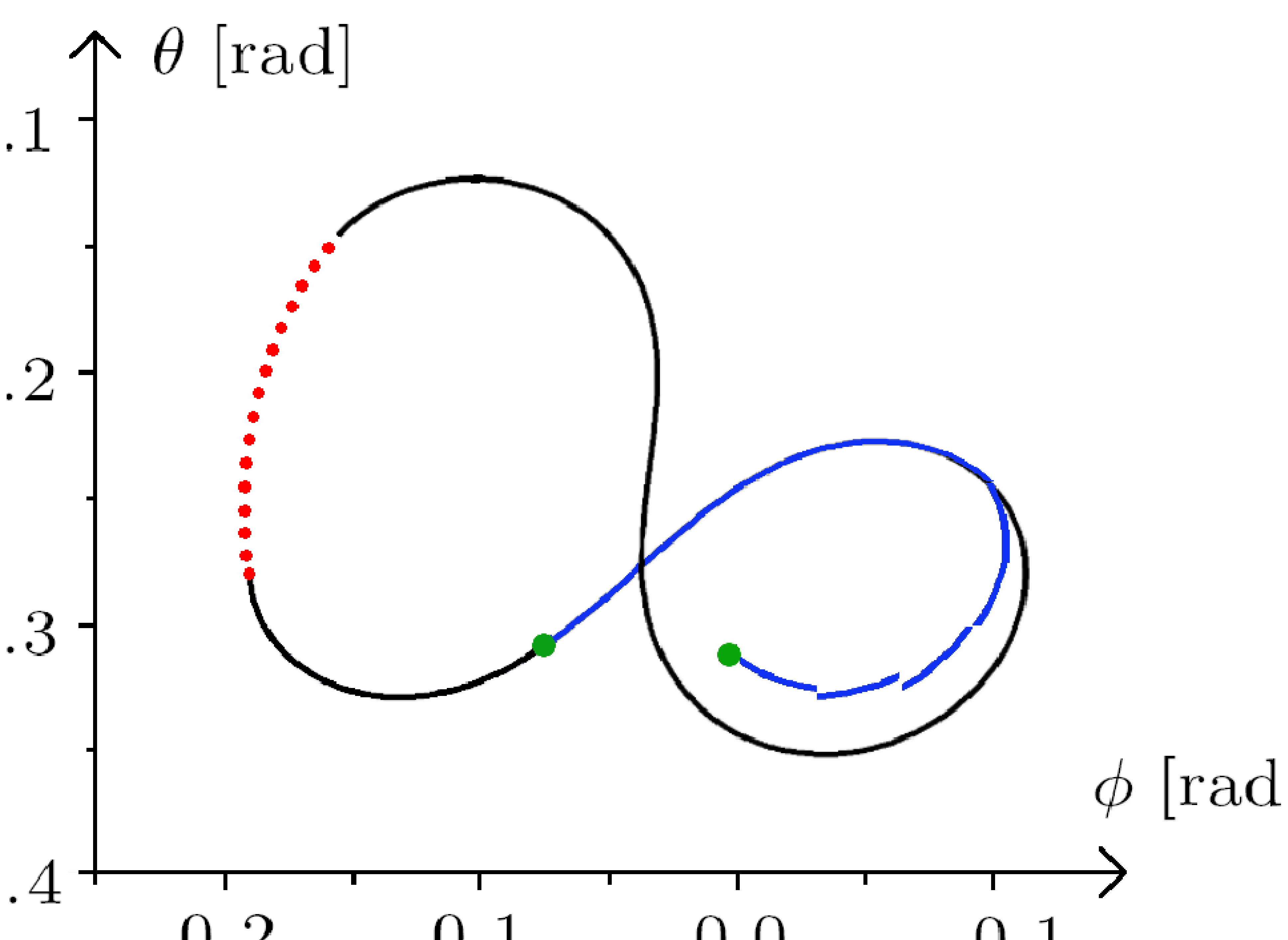
θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1

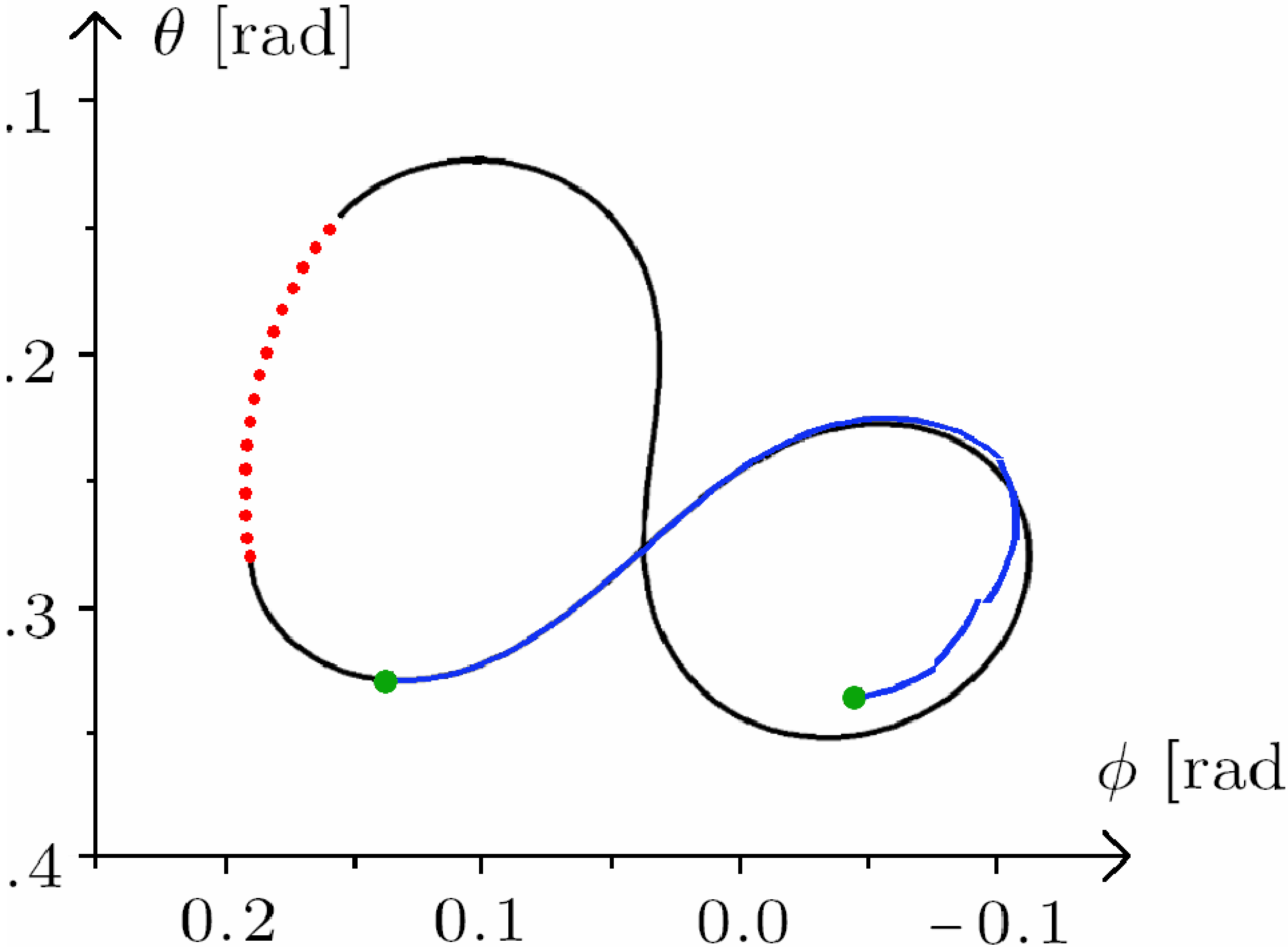

θ [rad]
.1
.2
.3
.4
ϕ [rad
0.2
0.1
0.0
−0.1


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: ACADO CPU Time per RTI below 50 ms

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

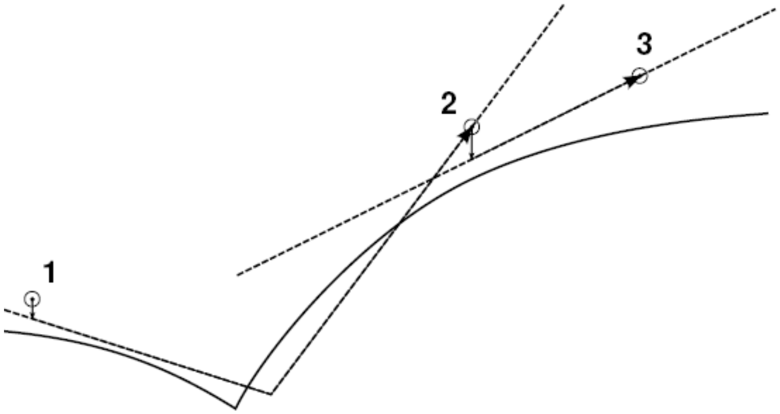
Feedback Phase: 3 ms

(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

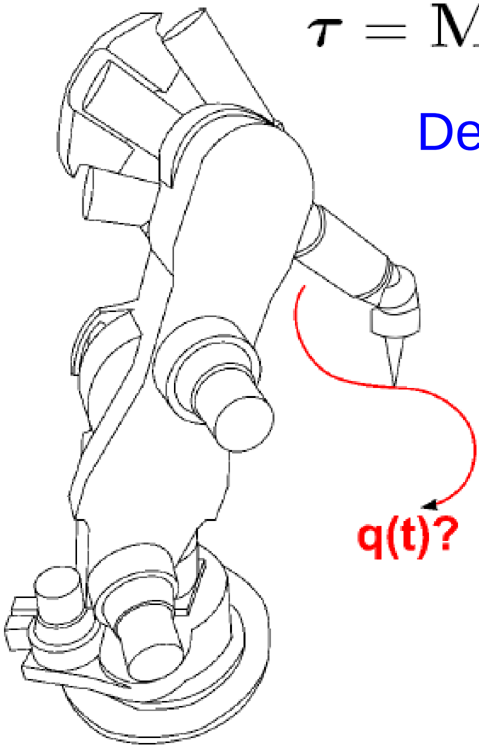
(on Intel Core 2 Duo CPU T7250, 2 GHz)



Model Validation Experiments at K.U. Leuven

Time Optimal Robot Motion (D. Verscheure et al.)

Nonlinear Dynamic Robot Model (6 DOF)



$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_s(\mathbf{q})\mathrm{sgn}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q})$$

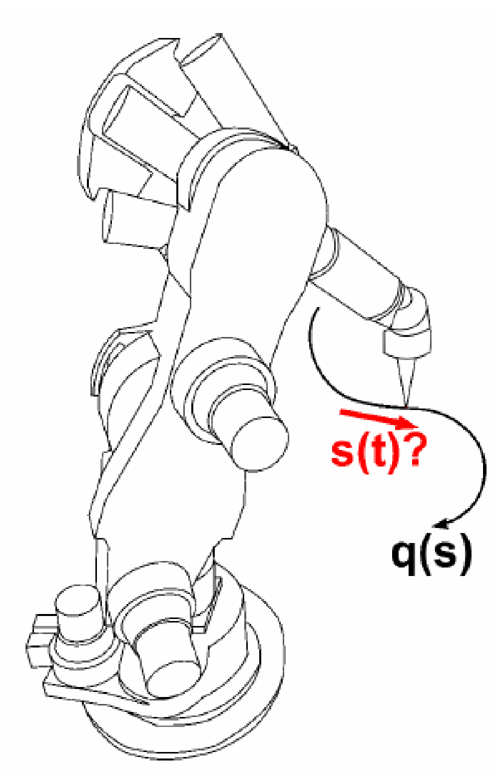
Desired: Time Optimal Trajectory

Often

- geometric path fixed
- velocity free for optimization

Then, model can be written as:

$$\boldsymbol{\tau}(s) = \mathbf{m}(s)\ddot{s} + \mathbf{c}(s)\dot{s}^2 + \mathbf{g}(s)$$

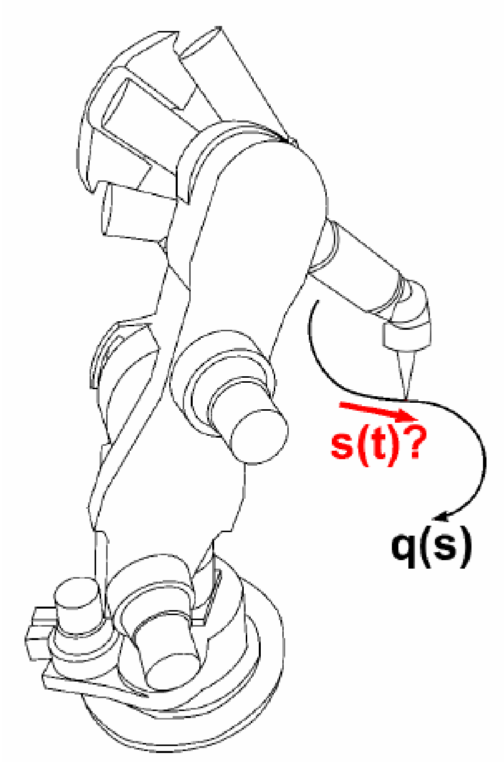


Nonlinear Time Optimal Robot Control

- Minimize time

subject to
- Boundary conditions
- **Torque limits**

$$
\begin{aligned}
&\min_{T, s(\cdot), \boldsymbol{\tau}(\cdot)} \quad T, \\
&\text{subject to } \boldsymbol{\tau}(t) = \mathbf{m}(s(t))\ddot{s}(t) \\
&\qquad\qquad + \mathbf{c}(s(t))\dot{s}(t)^2 + \mathbf{g}(s(t)), \\
&\qquad s(0) = 0, \\
&\qquad s(T) = 1, \\
&\qquad \dot{s}(0) = \dot{s}_0, \\
&\qquad \dot{s}(T) = \dot{s}_T, \\
&\qquad \dot{s}(t) \geq 0, \\
&\qquad \underline{\boldsymbol{\tau}}(s(t)) \leq \boldsymbol{\tau}(t) \leq \overline{\boldsymbol{\tau}}(s(t)), \\
&\qquad \text{for } t \in [0, T],
\end{aligned}
$$

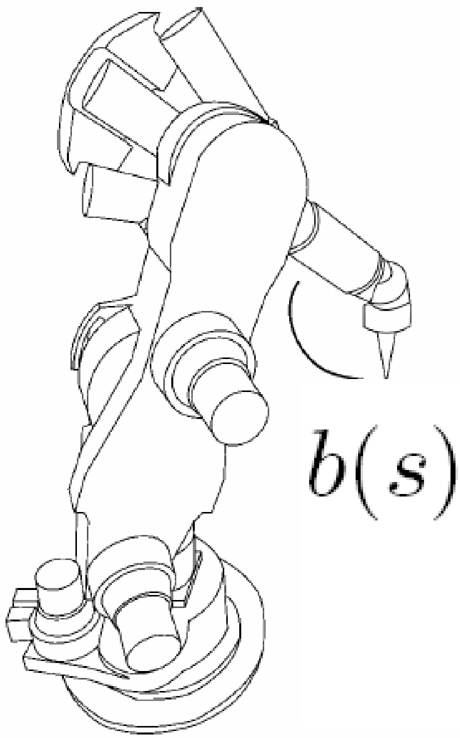


Nonlinear model → non-convex problem

Transformation into Convex Problem

After time transformation, previous problem is equivalent to:

$$
\begin{aligned}
\min_{a(\cdot),b(\cdot),\boldsymbol{\tau}(\cdot)} & \int_0^1 \frac{1}{\sqrt{b(s)}} ds, \\
\text{subject to } & \boldsymbol{\tau}(s) = \mathbf{m}(s)a(s) + \mathbf{c}(s)b(s) + \mathbf{g}(s), \\
& b(0) = \dot{s}_0^2, \\
& b(1) = \dot{s}_T^2, \\
& b'(s) = 2a(s), \\
& b(s) \geq 0, \\
& \underline{\boldsymbol{\tau}}(s) \leq \boldsymbol{\tau}(s) \leq \overline{\boldsymbol{\tau}}(s), \\
& \text{for } s \in [0,1].
\end{aligned}
$$



Linear model, convex cost → convex problem

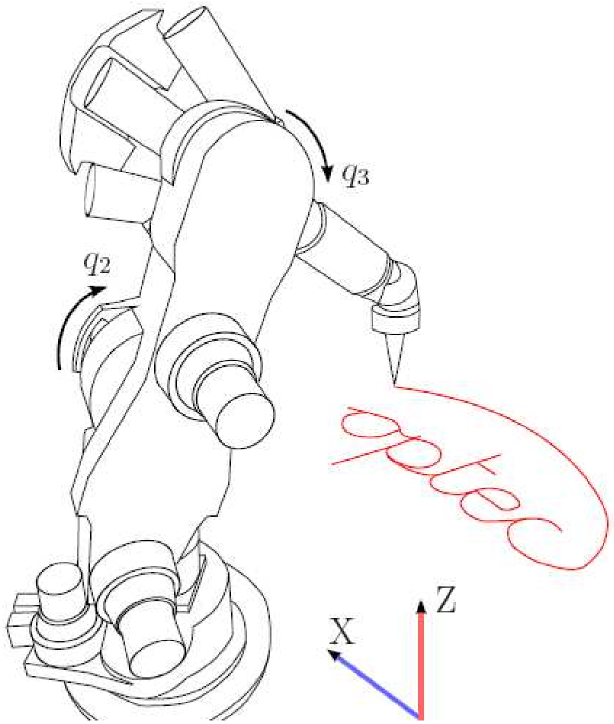
TimeOpt Software: Real-Time Control Setup

- Variable horizon length for future path (user decides online)
- require that robot rests at end of horizon (for safety)
- Solve with Interior Point formulation after discretization

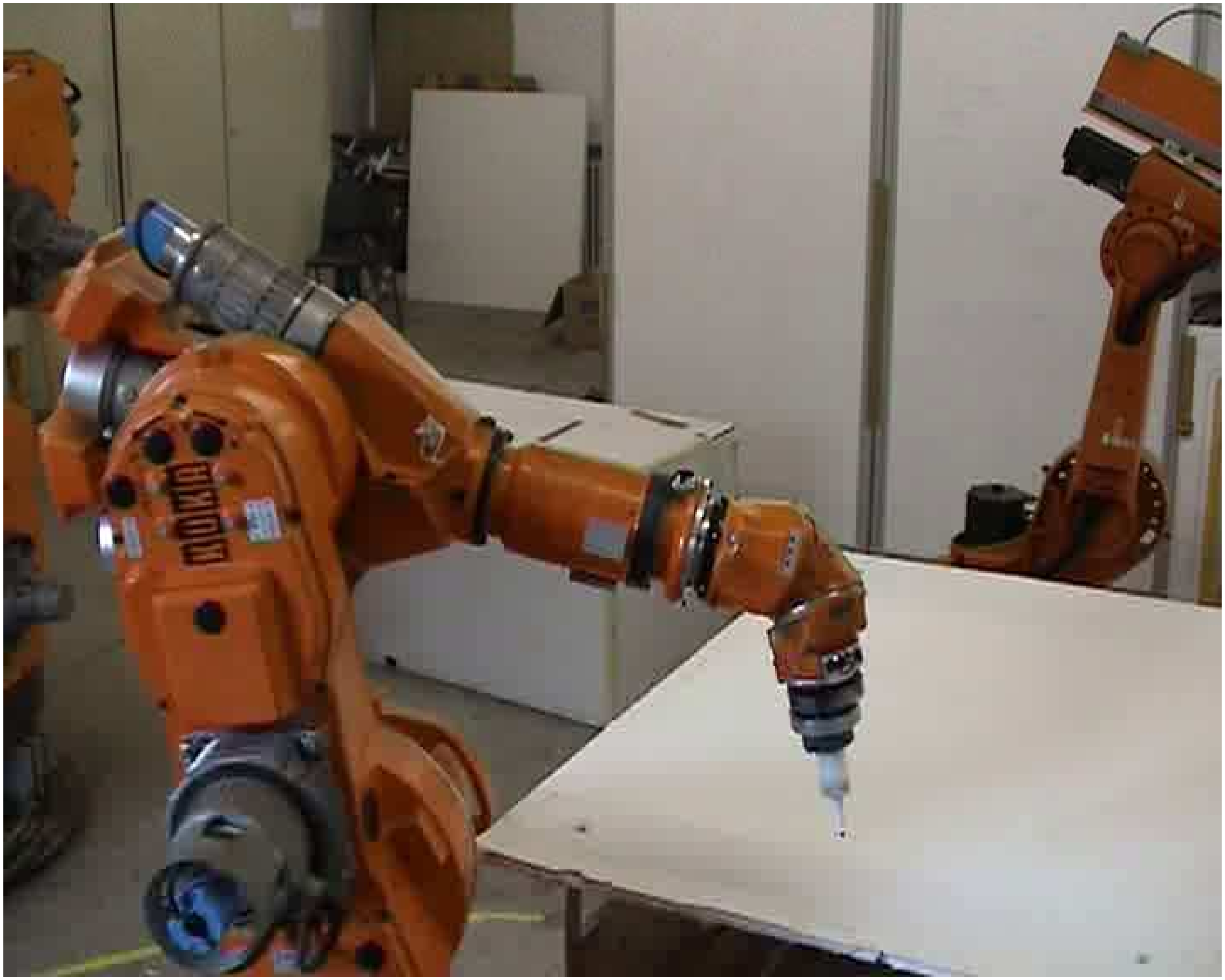
$$\min_{b^k} \sum_{k=0}^{K-1} \left[f_o^k(b^k, b^{k+1}) - \kappa \sum_{i=1}^{n} \log \left((\overline{\tau}_i(s^{k+1/2}) - f_{c,i}^k(b^k, b^{k+1}))(-\underline{\tau}_i(s^{k+1/2}) + f_{c,i}^k(b^k, b^{k+1})) \right) \right]$$

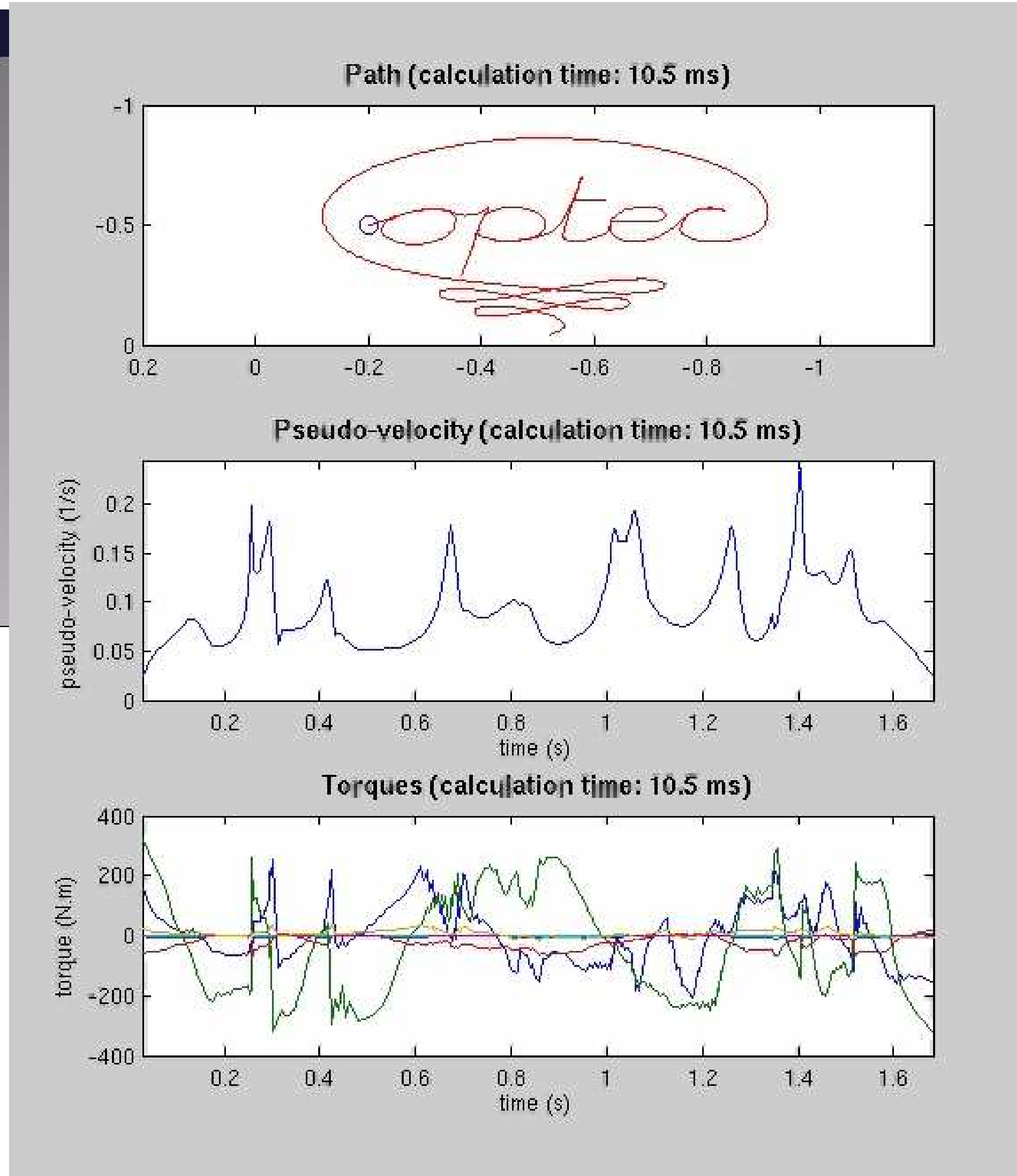
  - use FIXED barrier parameter
  - exploit banded structure
  - use „IP real-time iterations“ for approximate path-following
  - Implement in C on OROCOS control software

→ 2 ms CPU time per 900 var. problem



Result: Online Optimization with 500 Hz




Path (calculation time: 10.5 ms)
-1
-0.5
0
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
optec
Pseudo-velocity (calculation time: 10.5 ms)
pseudo-velocity (1/s)
0.2
0.15
0.1
0.05
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
time (s)
Torques (calculation time: 10.5 ms)
torque (N.m)
400
200
0
-200
-400
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
time (s)

Summary

- Real-Time Optimization needs sophisticated numerical methods
- OPTEC develops open source software for nonlinear dynamic optimization
- Real-Time Optimization powerful tool in mechatronic MPC applications
  - qpOASES: TOMPC (100 Hz), industrial gas engine
  - ACADO: Kite NMPC (20 Hz)
  - TimeOpt: Convex time optimal robot NMPC (500 Hz)
- Lots of exciting applications in engineering that need ultra-fast real-time optimization algorithms

Invitation to Leuven: July 8, 5 p.m.

12th Simon Stevin Lecture on Optimization in Engineering

Simon Stevin,
(1548-1620),
Flemish mathematician
and engineer

Lieven Vandenberghe:
"Convex techniques for sparse and low-order model selection"

July 8, 2009, 5 p.m., Aud. CS, KUL
followed by a reception

All DYSCO members and friends are most welcome!

'09
BFG
LEUVEN

14th Belgian-French-German Conference on Optimization

Leuven, September 14-18, 2009

Special Topic: Optimization in Engineering