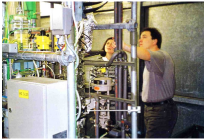
Nonlinear Model Predictive Control

Moritz Diehl

Dynamic Technical Processes

SMB process (Dortmund)



Distillation column (Stuttgart)

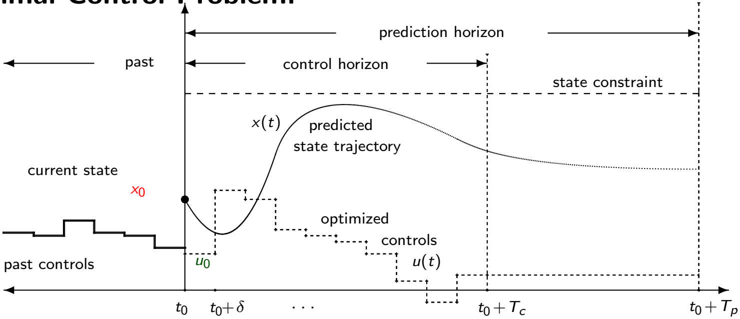
Power Plant (Pavia)

Polymer Reactor (BASF)

- Idea: use model to optimally operate plants e.g. with respect to
  - productivity,
  - product purity,
  - energy consumption,
  - safety, ...
- Problem: offline optimal control cannot cope with model-plant mismatch and disturbances
- Need *closed loop* controls!

Nonlinear Model Predictive Control (NMPC)

- Each sampling time, solve for given system state x_0 an **Optimal Control Problem:**

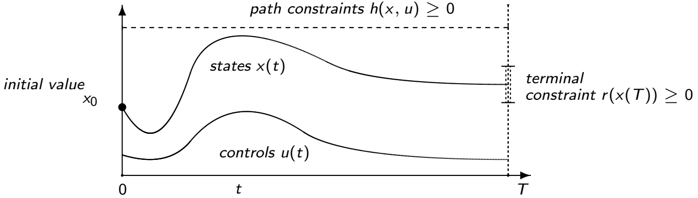


- Give first control move u_0 back to real-world system. Move horizon.
- Result: **Feedback law** $u_0(x_0)$. Can compensate for disturbances and modelling errors.

Example: Distillation Column (ISR, Stuttgart)

- Aim: to ensure product purity, keep two temperatures (T_{14}, T_{28}) constant despite disturbances
- least squares objective:
  $$\min \int_{t_0}^{t_0+T_p} \left\| \begin{array}{l} T_{14}(t) - T_{14}^{\mathrm{ref}} \\ T_{28}(t) - T_{28}^{\mathrm{ref}} \end{array} \right\|_2^2 dt$$
- control horizon 10 min
- prediction horizon 10 h
- stiff DAE model with 82 differential and 122 algebraic state variables
- Desired sampling time: 30 seconds.

NMPC Optimal Control Problem



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

$$\begin{array}{rcll}
\text{subject to} \quad x(0) - {\color{red}x_0} & = & 0, & & \text{(fixed initial value)} \\
\dot{x}(t) - f(x(t), z(t), u(t)) & = & 0, & t \in [0, T], & \text{(DAE model)} \\
g(x(t), z(t), u(t)) & = & 0, & t \in [0, T], & \\
h(x(t), z(t), u(t)) & \geq & 0, & t \in [0, T], & \text{(path constraints)} \\
r\,(x(T)) & \geq & 0 & & \text{(terminal constr.).}
\end{array}$$

Online Optimization Algorithm

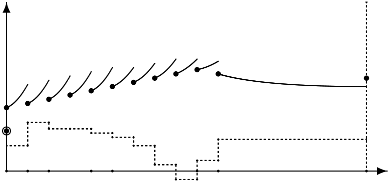
Basis:

- **Direct Multiple Shooting for DAE**

Online Features:

- Initialization of subsequent problems by **Initial Value Embedding**.
- **Real-Time Iterations** optimize while problem is changing.
- Proof of nominal stability of combined **System-Optimizer Dynamics**.

NLP in Direct Multiple Shooting



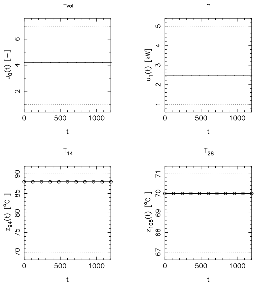
$$\underset{s,q}{\text{minimize}} \quad \sum_{i=0}^{N-1} l_i(s_i, q_i) \quad + \quad E\left(s_N\right)$$

subject to

$$\begin{aligned}
s_0 - x_0 &= 0, & & \text{(initial value)} \\
s_{i+1} - x_i(t_{i+1}; s_i, q_i) &= 0, \;\; i = 0, \dots, N-1, & & \text{(continuity)} \\
h(s_i, q_i) &\geq 0, \;\; i = 0, \dots, N, & & \text{(discretized path constr.)} \\
r\left(s_N\right) &\geq 0. & & \text{(terminal constraints)}
\end{aligned}$$

Distillation Online Scenario

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

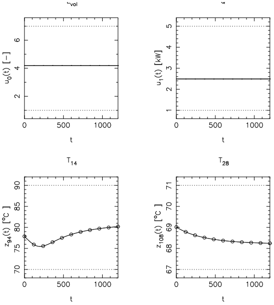


- **Suddenly**, system state x_0 is disturbed.
- What to do with optimizer?

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.

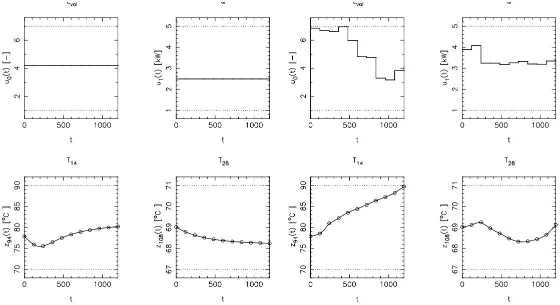
Initialization



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.

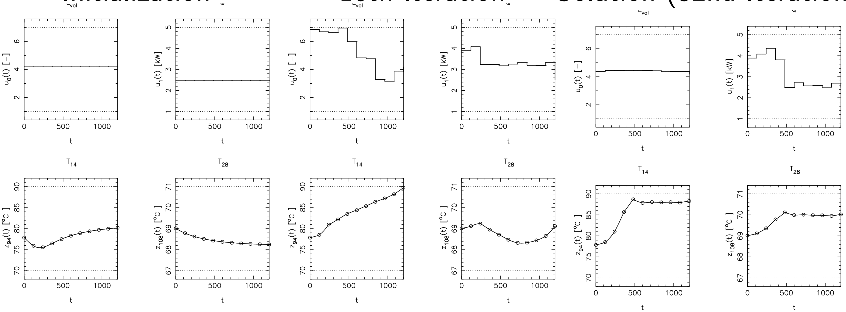
Initialization 16th Iteration



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)

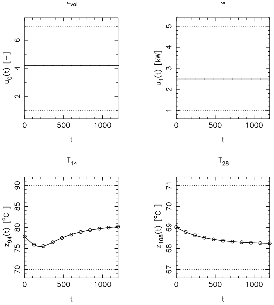


Solution only after 600 seconds - much too late!

Conventional, but with Gauss-Newton Hessian

- use Gauss-Newton method for least-squares integrals (Diehl, 2001)

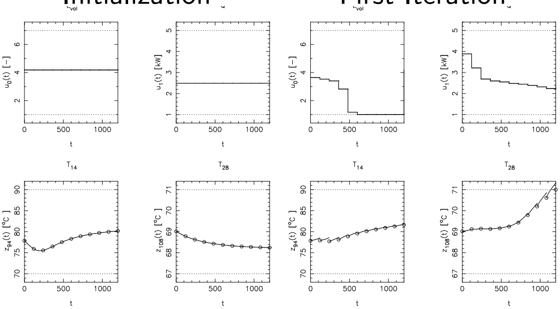
Initialization



Conventional, but with Gauss-Newton Hessian

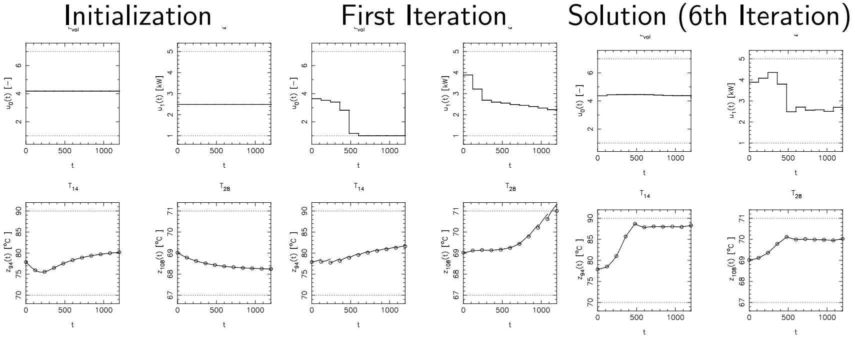
- use Gauss-Newton method for least-squares integrals (Diehl, 2001)

Initialization First Iteration



Conventional, but with Gauss-Newton Hessian

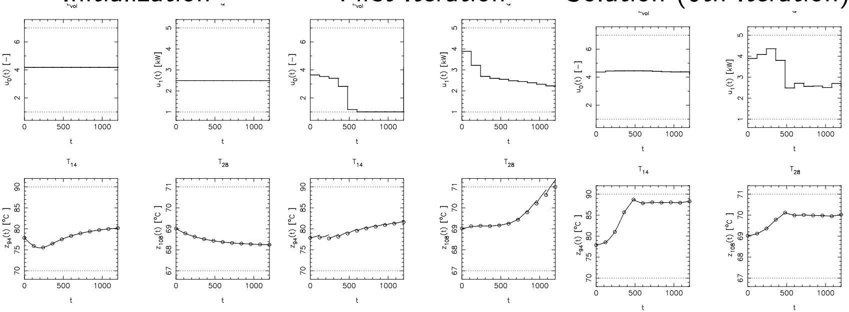
- use Gauss-Newton method for least-squares integrals (Diehl, 2001)



Conventional, but with Gauss-Newton Hessian

- use Gauss-Newton method for least-squares integrals (Diehl, 2001)

Initialization | First Iteration | Solution (6th Iteration)

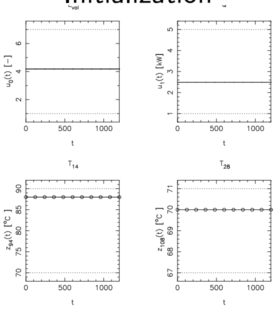


Solution still takes two minutes - can't we do better?

New Approach: Initial Value Embedding

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

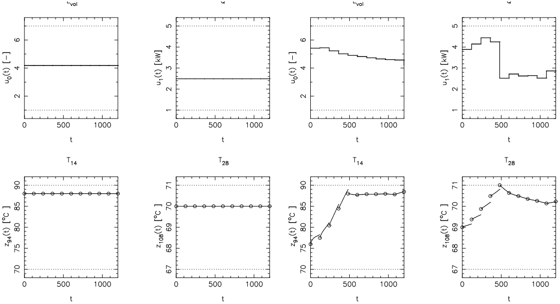
Initialization



New Approach: Initial Value Embedding

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

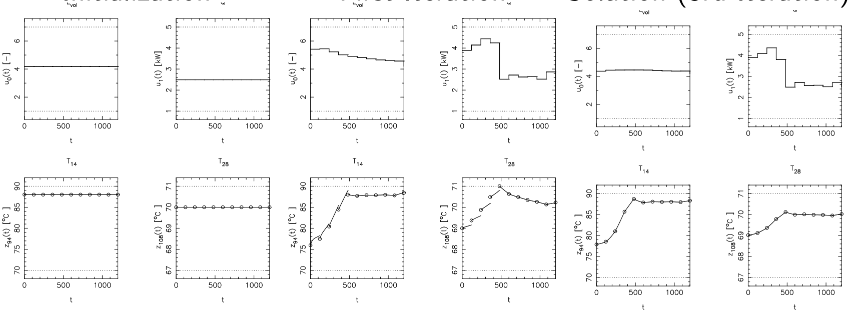
Initialization First Iteration



New Approach: Initial Value Embedding

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

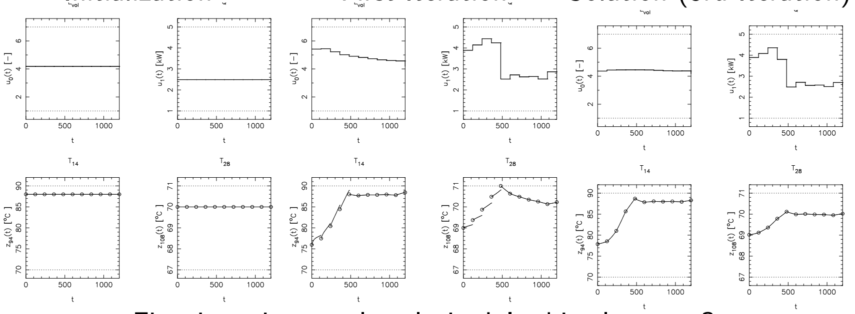
Initialization First Iteration Solution (3rd Iteration)



New Approach: Initial Value Embedding

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

Initialization First Iteration Solution (3rd Iteration)



First iteration nearly solution! Is this always so?

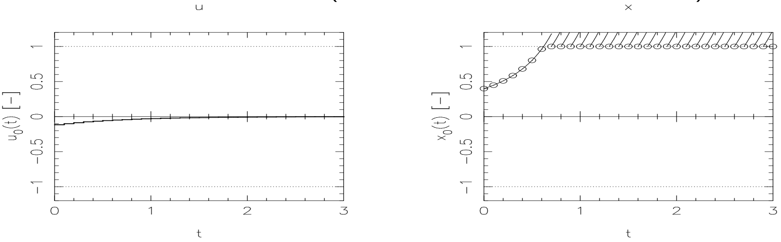
Test with NMPC Example Problem

$$\underset{x(\cdot),\,u(\cdot)}{\text{minimize}} \int_0^3 x(t)^2 + u(t)^2 \, dt \quad \text{s.t.} \quad \begin{cases} x(0) = & x_0, \\ \dot{x} = & (1+x)x + u, \quad t \in [0,3], \\ |x| \leq & 1, \; |u| \leq 1, \quad t \in [0,3], \\ x(3) = & 0. \end{cases}$$

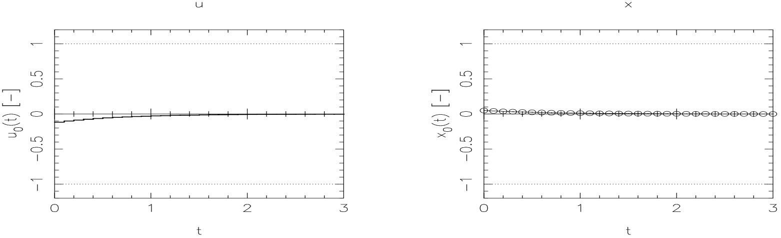
- Before, system was in state $x_0 = 0.05$
- Optimizer had found solution for $x_0 = 0.05$
- After disturbance, new state is $x_0 = 0.40 \gg 0.05$
- How to compute new solution?

Transition from $x_0 = 0.05$ to $x_0 = 0.4$

Conventional Initialization (old controls, new initial value):

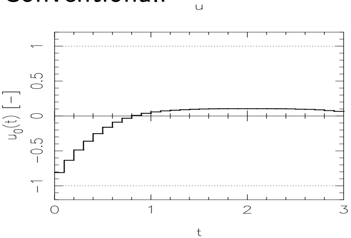


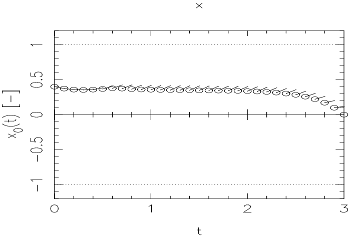
Initial Value Embedding (old solution, violates $s_0 - x_0 = 0$):



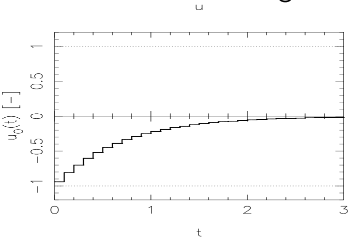
First Iteration

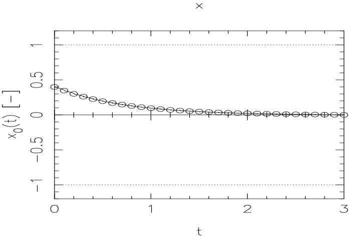
Conventional:





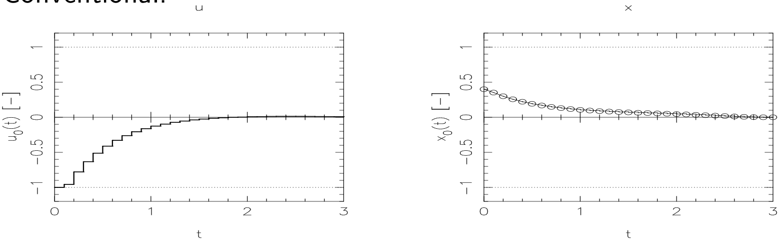
Initial Value Embedding:



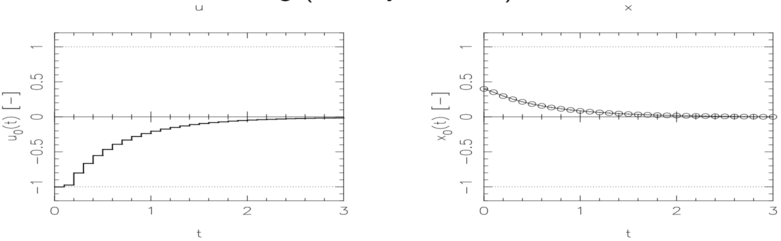


2nd Iteration

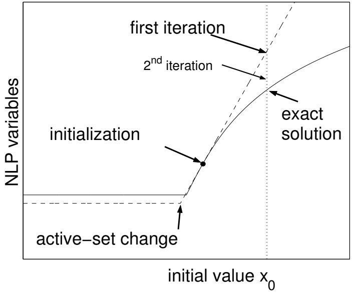
Conventional:



Initial Value Embedding (already solution):

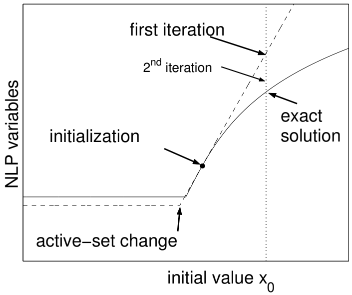


Initial Value Embedding



- 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

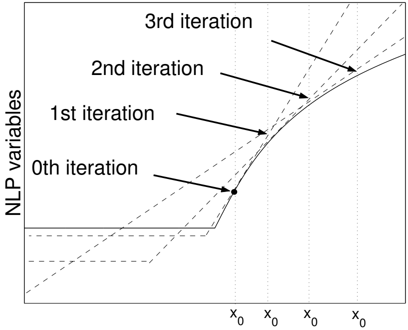


- first iteration is tangential predictor for exact solution (for exact Hessian SQP)
- also valid for active set changes
- derivative can be computed *before* x_0 is known: first iteration nearly without delay

Why wait until convergence and do nothing in the meantime?

Real-Time Iterations

Iterate, *while* problem is changing!



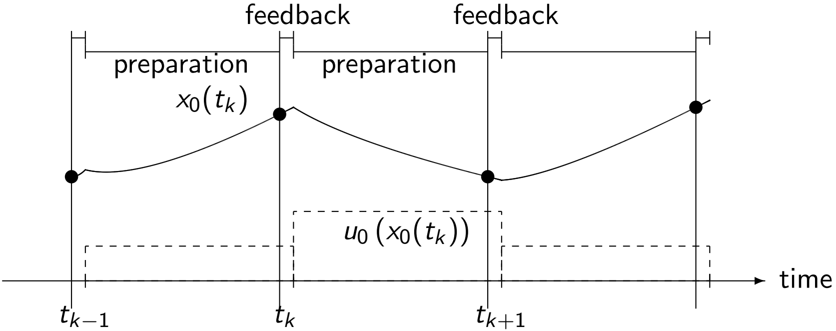
- tangential prediction after each change in x_0
- solution accuracy is increased with each iteration when x_0 changes little
- iterates stay close to solution manifold

Real-Time Iteration Algorithm:

1. **Preparation Step (long):**
   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.

- minimal cycle-duration (as **one** SQP iteration)
- negligible feedback delay (≈ 1 % of cycle)
- nevertheless fully nonlinear optimization

Real-time iterations minimize feedback delay

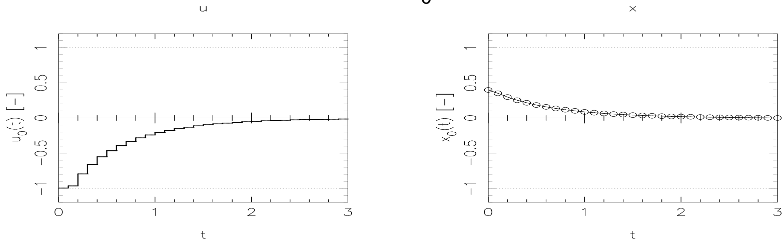


For distillation model:

- preparation time: $\approx$ 20.0 seconds
- feedback delay: $\approx$ 0.2 seconds ($\approx$1%)

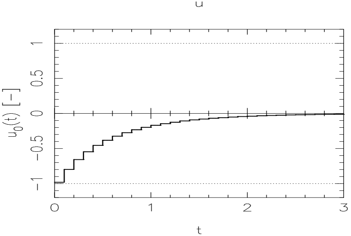
Real-Time Iterations with NMPC Example

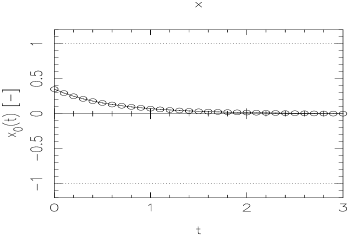
- go through initial values $x_0 = 0.40, 0.35, \ldots 0.05$,
- then jump to $-0.50, -0.55, \ldots, -0.70$
- Start with exact solution of $x_0 = 0.40$:



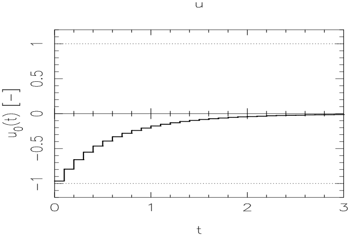
1st Real-Time Iteration, $x_0 = 0.35$

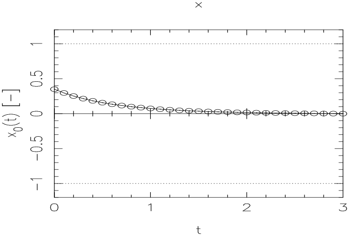
Real-time iterations:





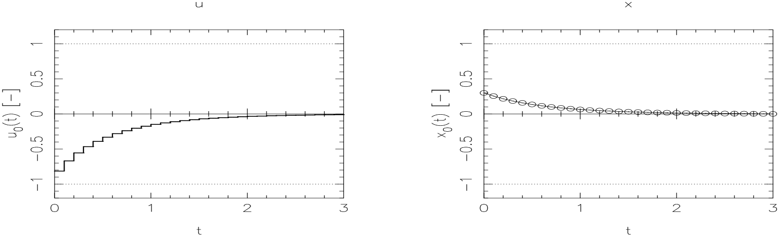
Exact solution for comparison:



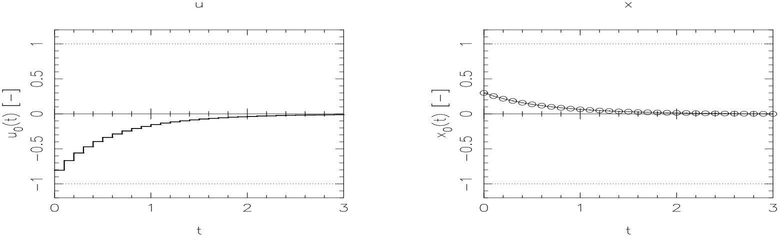


2nd Real-Time Iteration, $x_0 = 0.30$

Real-time iterations:

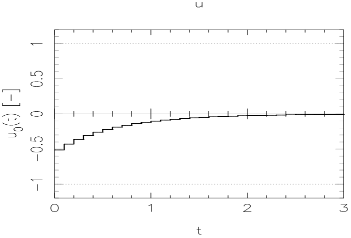


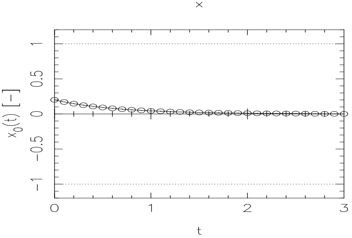
Exact solution for comparison:



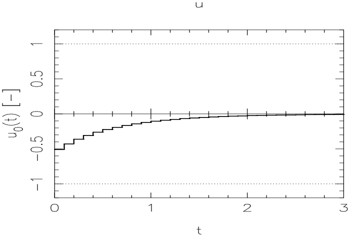
4th Real-Time Iteration, $x_0 = 0.20$

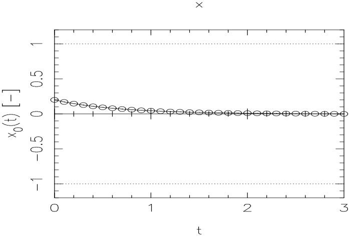
Real-time iterations:





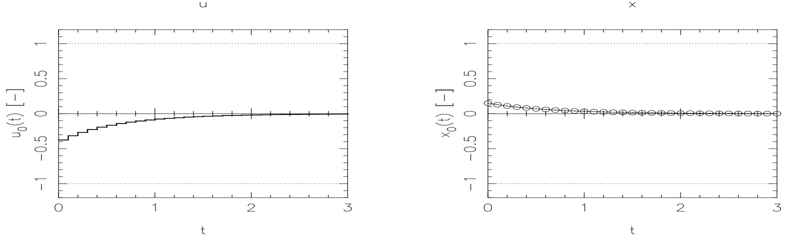
Exact solution for comparison:



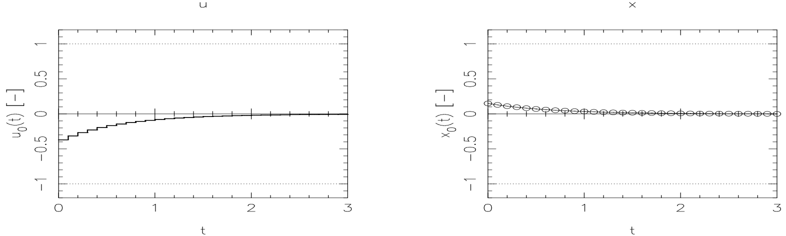


5th Real-Time Iteration, $x_0 = 0.15$

Real-time iterations:

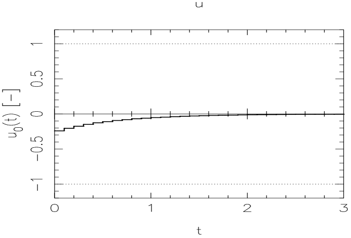


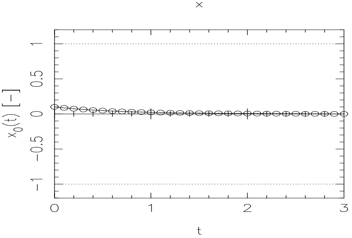
Exact solution for comparison:



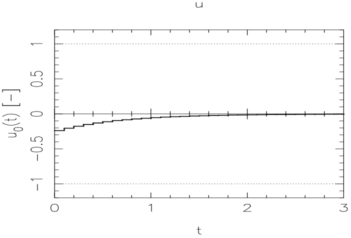
6th Real-Time Iteration, $x_0 = 0.10$

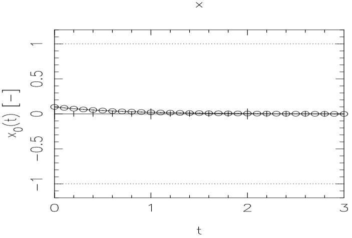
Real-time iterations:





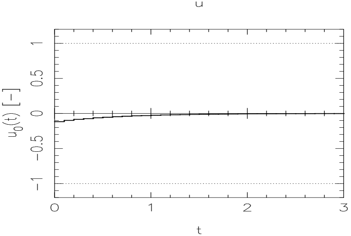
Exact solution for comparison:

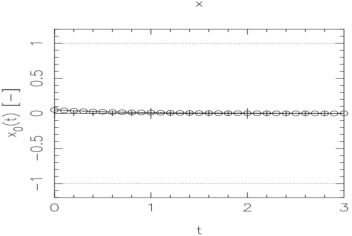




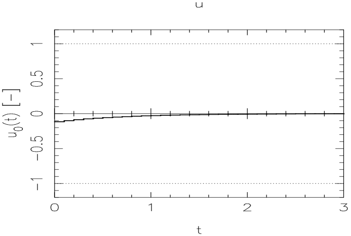
7th Real-Time Iteration, $x_0 = 0.05$

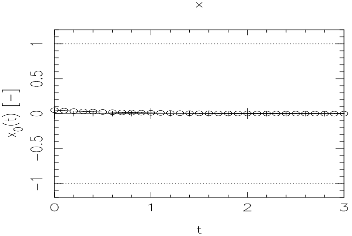
Real-time iterations:





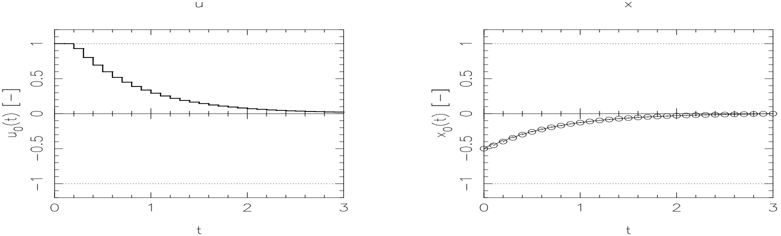
Exact solution for comparison:



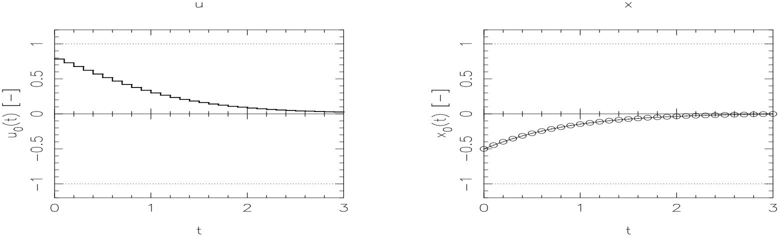


8th Real-Time Iteration, $x_0 = -0.50$

Real-time iterations:

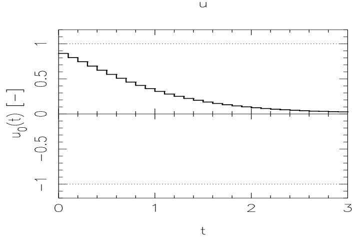


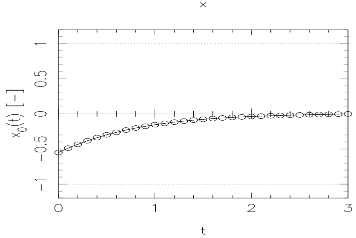
Exact solution for comparison:



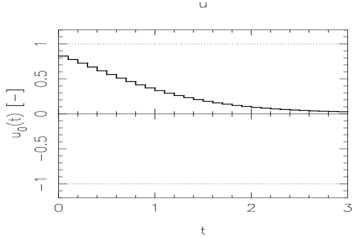
Next Real-Time Iteration, $x_0 = -0.55$

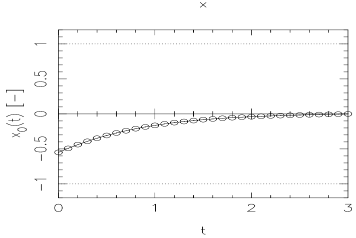
Real-time iterations:





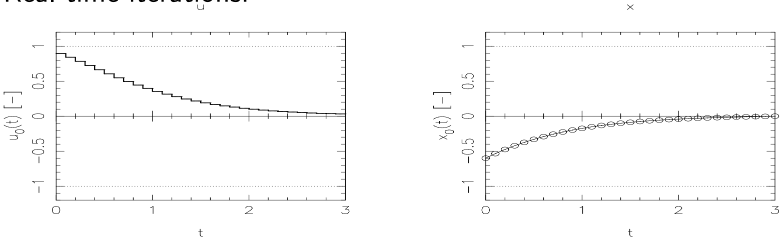
Exact solution for comparison:



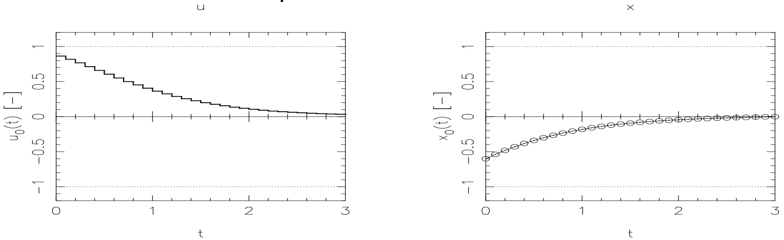


Next Real-Time Iteration, $x_0 = -0.60$

Real-time iterations:

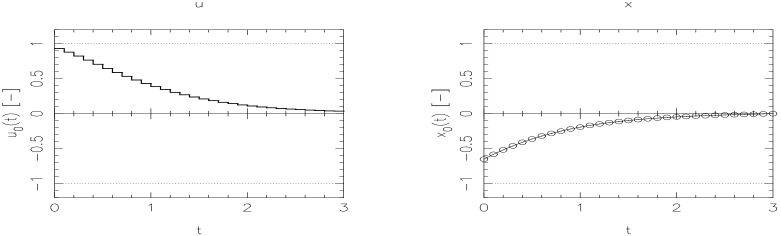


Exact solution for comparison:

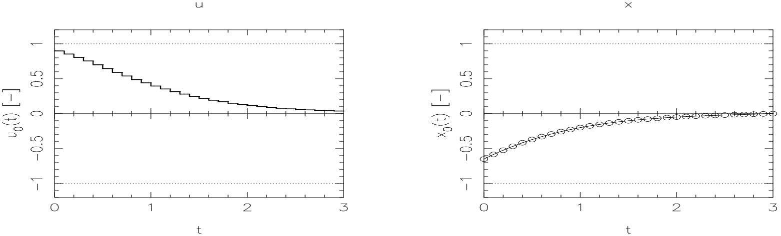


Next Real-Time Iteration, $x_0 = -0.65$

Real-time iterations:

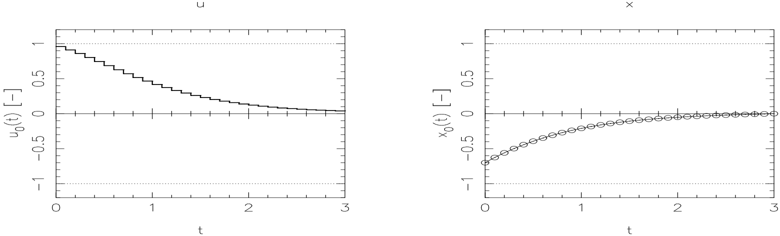


Exact solution for comparison:

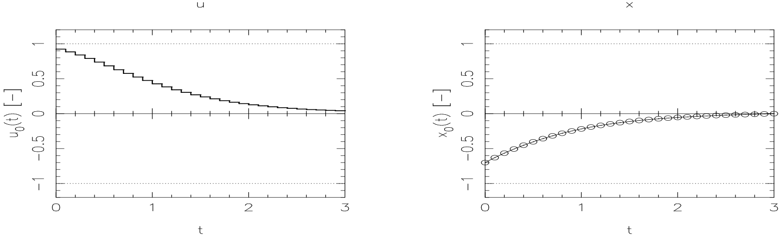


Next Real-Time Iteration, $x_0 = -0.70$

Real-time iterations:



Exact solution for comparison:



Nominal Stability of Closed Loop?

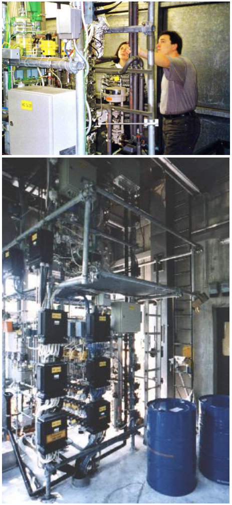
- Real process and optimizer are coupled with each other. Can numerical errors grow and destabilize closed loop?
- Stability analysis combines concepts from both, **NMPC stability theory** and **convergence theory of nonlinear optimization**.
- Nominal stability shown under realistic assumptions.

  [Diehl, Findeisen, Bock, Schlöder, Allgöwer: Nominal stability of the real-time iteration scheme for nonlinear model predictive control. IEE Control Theory Appl. (2005)]
- After disturbance of size ϵ: loss of optimality is of order $O(\epsilon^2)$ for Gauss-Newton, and $O(\epsilon^4)$ for exact Hessian.

  [Diehl, Bock, Schlöder: A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control. SIAM J. Control & Opt. (2005)]

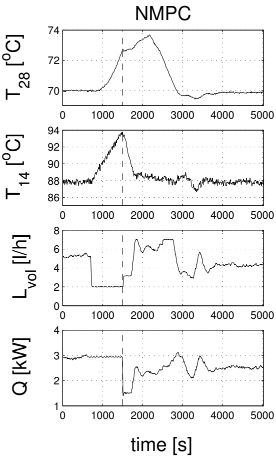
Realization at Distillation Column

(with Allgöwer, Findeisen, Nagy, Schwarzkopf, Uslu)



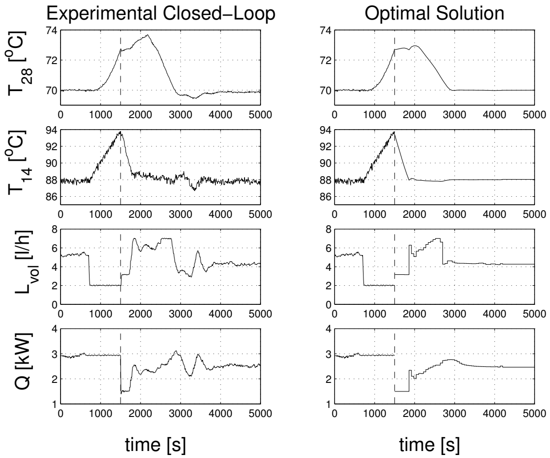
- Parameter estimation using dynamic experiments
- Online state estimation with Extended Kalman Filter variant, using only 3 temperature measurements to infer all 82 system states
- Implementation of estimator and optimizer on Linux Workstation.
- Communication with Process Control System via FTP all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

Large Disturbance (Heating), then NMPC



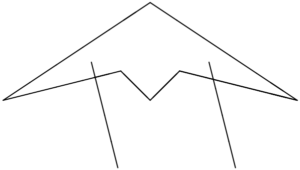
- Overheating by manual control
- NMPC only starts at $t = 1500$ s
- PI-controller not implementable, as disturbance too large (valve saturation)
- NMPC: at start control bound active
  $\Rightarrow$ T_{28} rises further
- Disturbance attenuated after half an hour

Comparison with Theoretical Optimal Solution

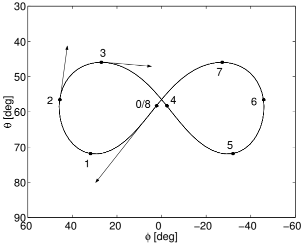


Simulated Control of a Looping Kite

Kite can be controlled by two lines:

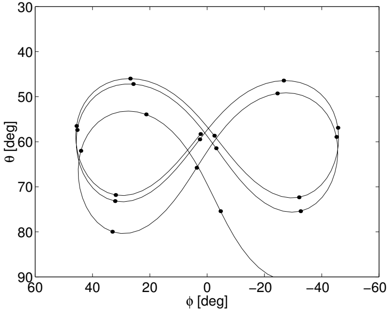


Control aim is to fly a "lying eight":



Period duration: 8 seconds

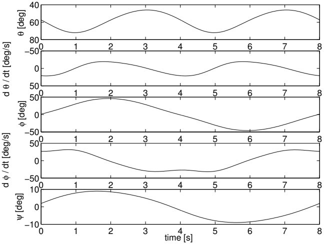
Orbit is Open Loop Unstable



Simulated open loop controlled kite crashes onto ground after 25 seconds!

$\Rightarrow$ feedback necessary

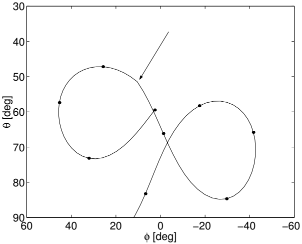
Nonlinear Model Predictive Control Setup



- predict two full periods (16 seconds)
- optimize quadratic deviation from "lying eight"
- choose one second sampling time
- use real-time iterations
  recall: negligible feedback delay

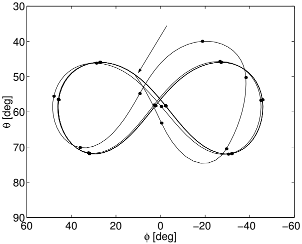
Weak Kick

Open loop controlled system:

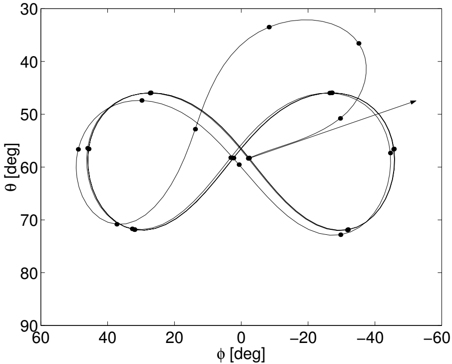


Crash after 5 seconds

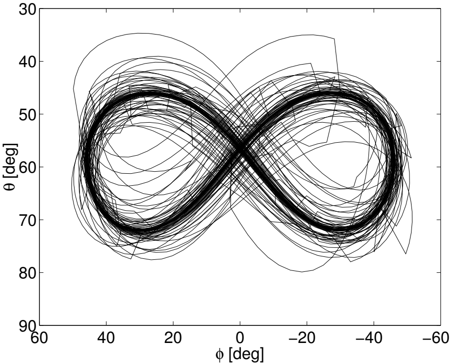
NMPC controlled system:



Strong Kick



Robustness Test with Strong Random Kicks



Summary

- **Nonlinear Model Predictive Control (NMPC)** allows optimal control of real world processes. Requires online optimization.
- Online optimization by no means just an application of fast offline optimization methods!
- **Direct, simultaneous** optimal control algorithms favourable for NMPC.
- Our algorithm based on:
  - **direct multiple shooting** with **Gauss-Newton** algorithm
  - **initial value embedding** to deliver tangential predictor
  - **real-time iterations** to have minimal cycle times and negligible feedback delay
- Nominal stability can be guaranteed.
- Thourougly tested numerically and experimentally.

References

- M. Diehl, H. J. Ferreau and N. Haverbeke: Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation. In: Nonlinear model predictive control, Eds. L. Magni, M.D. Raimondo and F. Allgöwer. Series: Lecture Notes in Control and Information Sciences, Vol 384, pp. 391–417, Springer, 2009.
- Diehl, M., Bock, H.G., Schl oder, J.P.: A real-time iteration scheme for nonlinear optimization in optimal feedback control. SIAM Journal on Control and Optimization 43(5), 17141736 (2005)
- M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer: Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. Journal of Process Control 12, pp. 577-585, 2002.
- M. Diehl, R. Findeisen, S. Schwarzkopf, I. Uslu, F. Allgöwer, H. G. Bock, E. D. Gilles, and J. P. Schlöder: An efficient algorithm for nonlinear model predictive control of large-scale systems, Automatisierungstechnik.
  Part I: Description of the method, 50(12), 2002.
  Part II: Application to a Distillation Column, 51(1), 2003.