Inexact SCP Methods for Hierarchical Optimization of Decomposable Systems

Moritz Diehl

Optimization in Engineering Center OPTEC

and Electrical Engineering Department ESAT

K.U. Leuven, Belgium

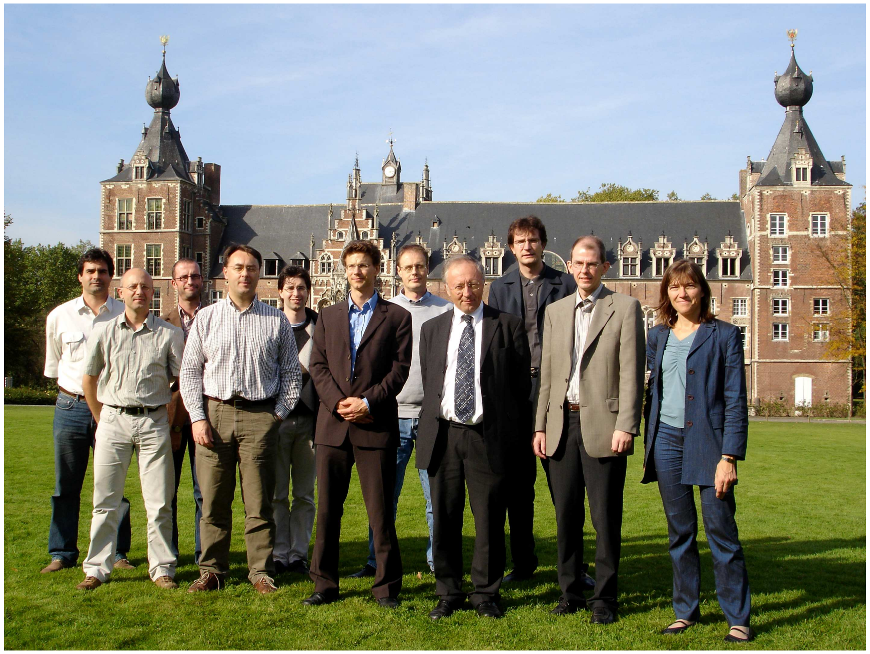
Joint work with **Carlo Savorgnan, Attila Kozma**, L. Van den Broeck, J. Swevers, H.J. Ferreau, B. Houska, Q. Tran Dinh, I. Necoara, C. Romani, R. Scattolini

LCCC, Lund, May 21, 2010

OPTEC - Optimization in Engineering Center

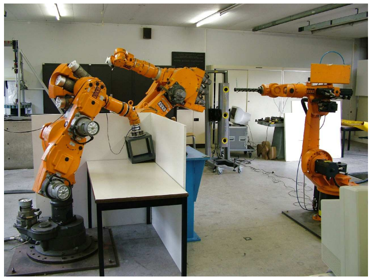
Center of Excellence of K.U. Leuven, since 2005, until 2017

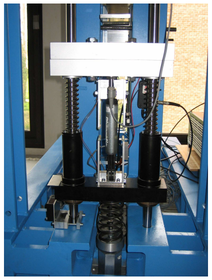
About 15 professors, 15 postdocs, and 40 PhD students involved in OPTEC research

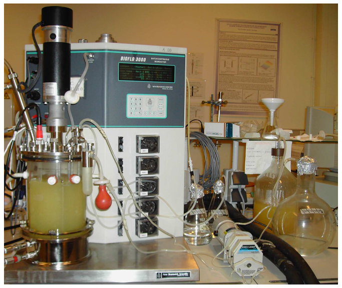


Founded by five Departments:

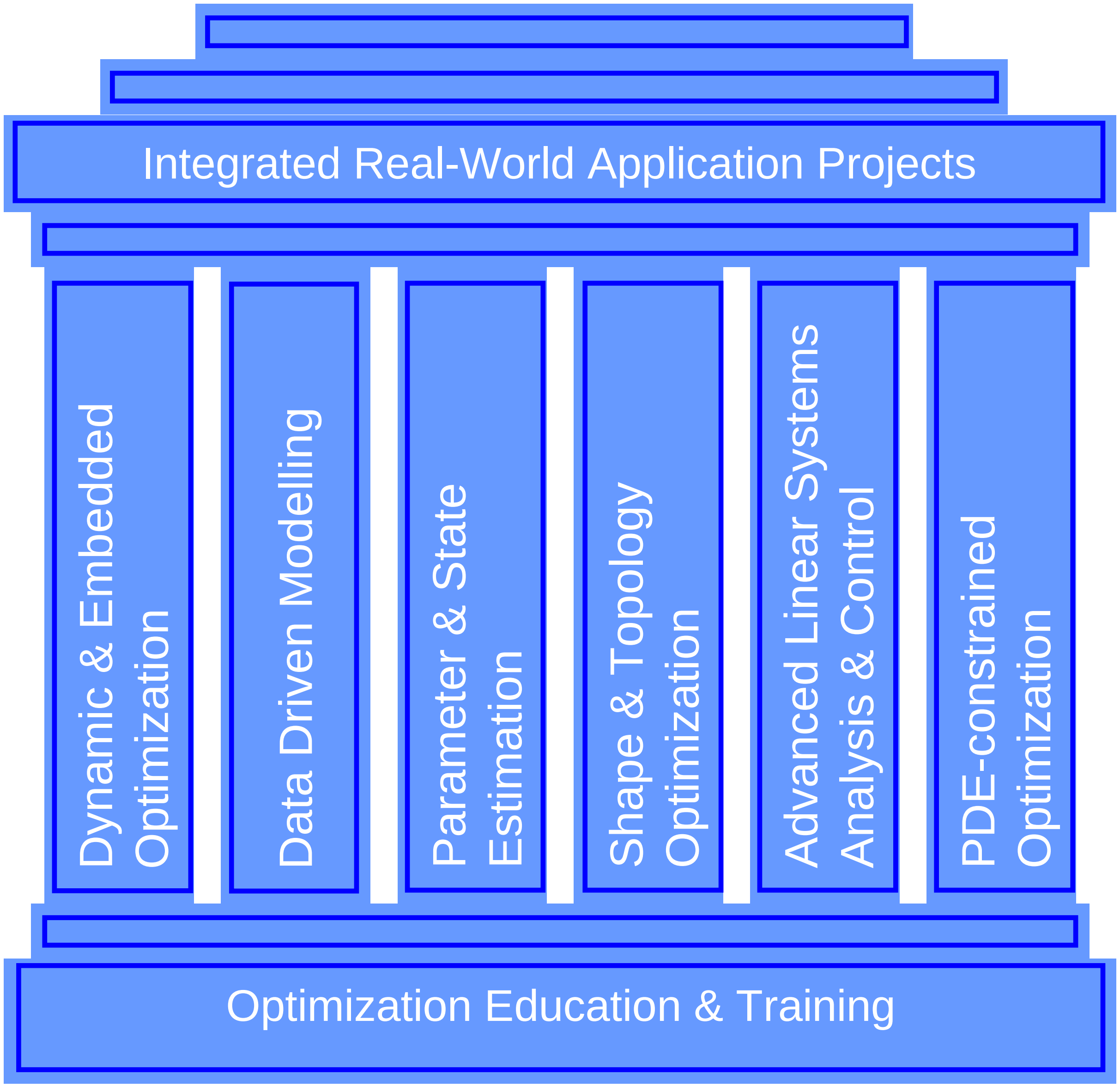
- Electrical Engineering
- Mechanical Engineering
- Chemical Engineering
- Computer Science
- Civil Engineering







OPTEC: 70 people in six methodological working groups



European project HD-MPC

HD-MPC: Hierarchical and distributed model predictive control for large-scale systems

- European STREP Project 2008 - 2011
- 10 research teams: Delft (co-ordinator), EDF*, Leuven*, Milano*, Aachen, Sevilla, UN Columbia, Supelec, Inocsa, Madison (* involved in hydro power valley problem)

Overview

- Large Decomposable Systems: the Hydro Power Valley
- Multidimensional Multiple Shooting
- Dual decomposition and online active set strategy
- Optimizing MPC of mechatronic systems

Motivation

Large-scale systems in engineering

- composed of multiple subsystems
- complex nonlinear dynamics and
- mutual influences

E.g. river networks, chemical production sites, airflow in buildings.

How to compute optimal controls e.g. for transients?

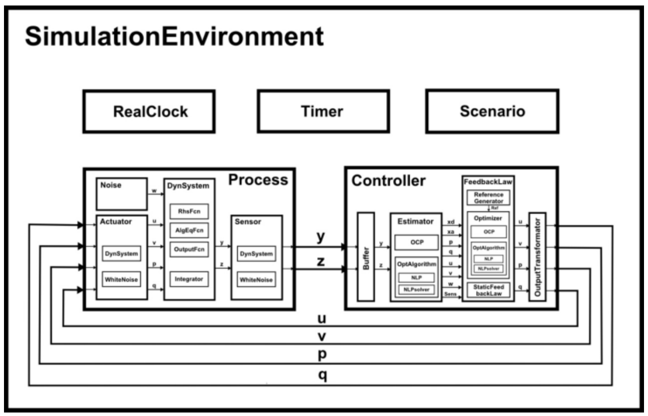
Motivation

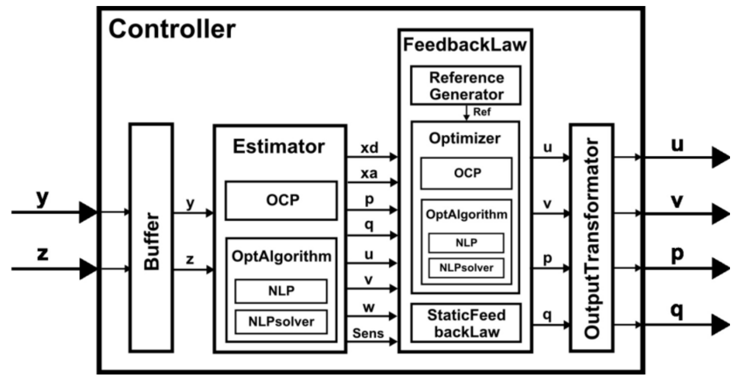
Assumption: simulators for individual subsystems exist

- use their own adaptive numerical integration schemes
- based on possibly different modelling languages
- can provide derivatives in forward and reverse mode (not yet standard, but provided e.g. by SUNDIALS, DASPK, DAESOL-II, ACADO Integrators, ...)

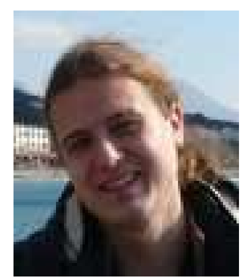
(ACADO Toolkit)

- 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



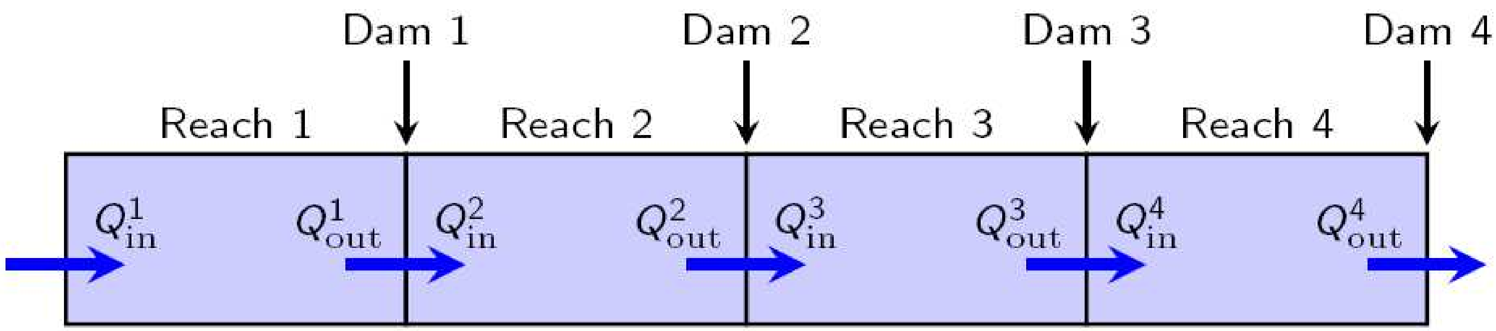
Available since September 2009, on www acado.org (→ google: "acado")

Hydro Power Valley (HPV)

River reaches connected by dams and hydro power units.
NMPC control aims:

- strictly respect level constraints
- match total power demand
- keep levels as constant as possible

Model system composed of 4 reaches.

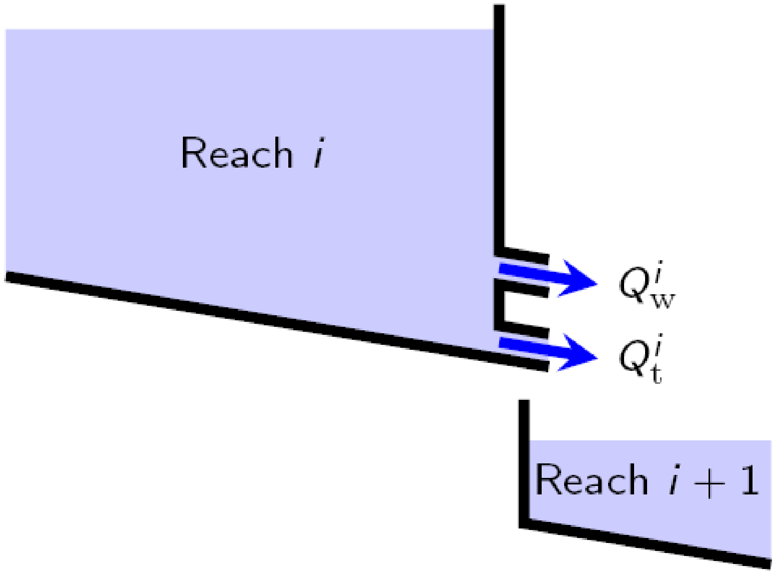


Hydro Power Valley (HPV)

Water flow in reaches modeled by Saint Venant PDE:

$$\begin{cases} \frac{\partial Q(z,t)}{\partial z} + w\frac{\partial H(z,t)}{\partial t} = 0 \\ \frac{1}{gw}\frac{\partial}{\partial t}\left(\frac{Q(z,t)}{H(z,t)}\right) + \frac{1}{2gw^2}\frac{\partial}{\partial z}\left(\frac{Q^2(t,z)}{H^2(t,z)}\right) + \frac{\partial H(t,z)}{\partial z} + I_f(z) - I_0 = 0 \end{cases}$$

Transform PDE into ODE by spatial discretization.



Output discharge given by:

- turbine discharge Q_t^i
- weir discharge Q_w^i

Power produced by turbine

$$P^i = k_t^i Q_t^i H_t^i$$

Coupling between subsystems

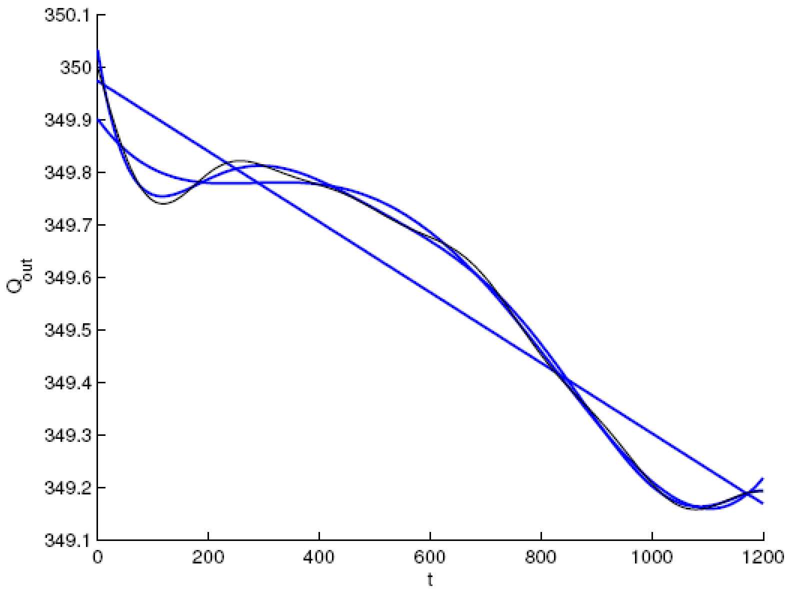
Dynamics of subsystems are coupled via in-/output profiles of "coupling variables". Infinite dimensional coupling.

Coupling between subsystems

Dynamics of subsystems are coupled via in-/output profiles of "coupling variables". Infinite dimensional coupling.

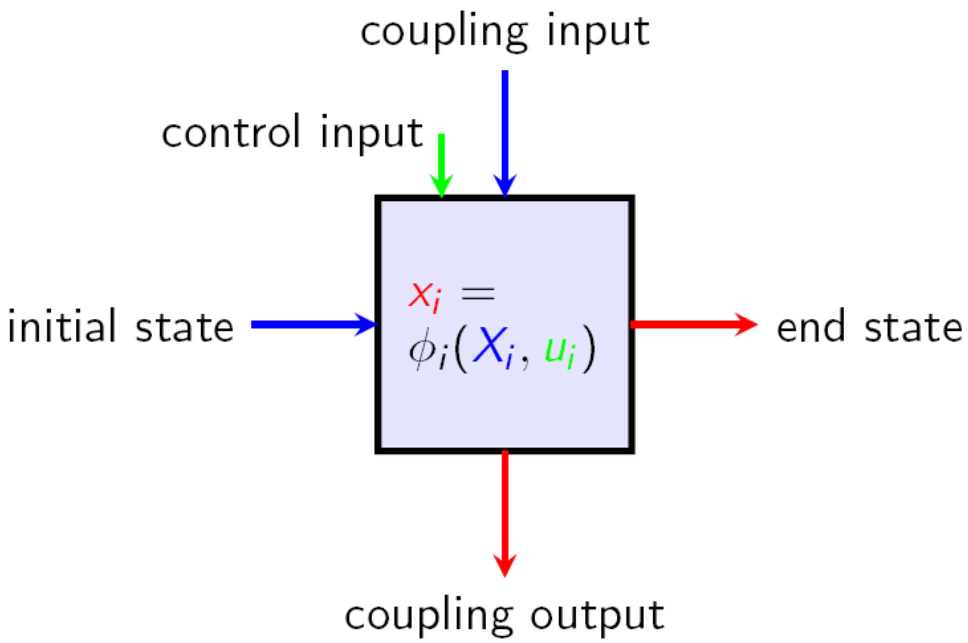
Can approximate coupling profile by orthogonal polynomials:

$$\int_a^b P_i(t)P_j(t)dt = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{otherwise} \end{cases}$$



Approximation of typical output water discharge profile (black) by polynomials of degree 1, 4 and 7.

The "Simulation Box"

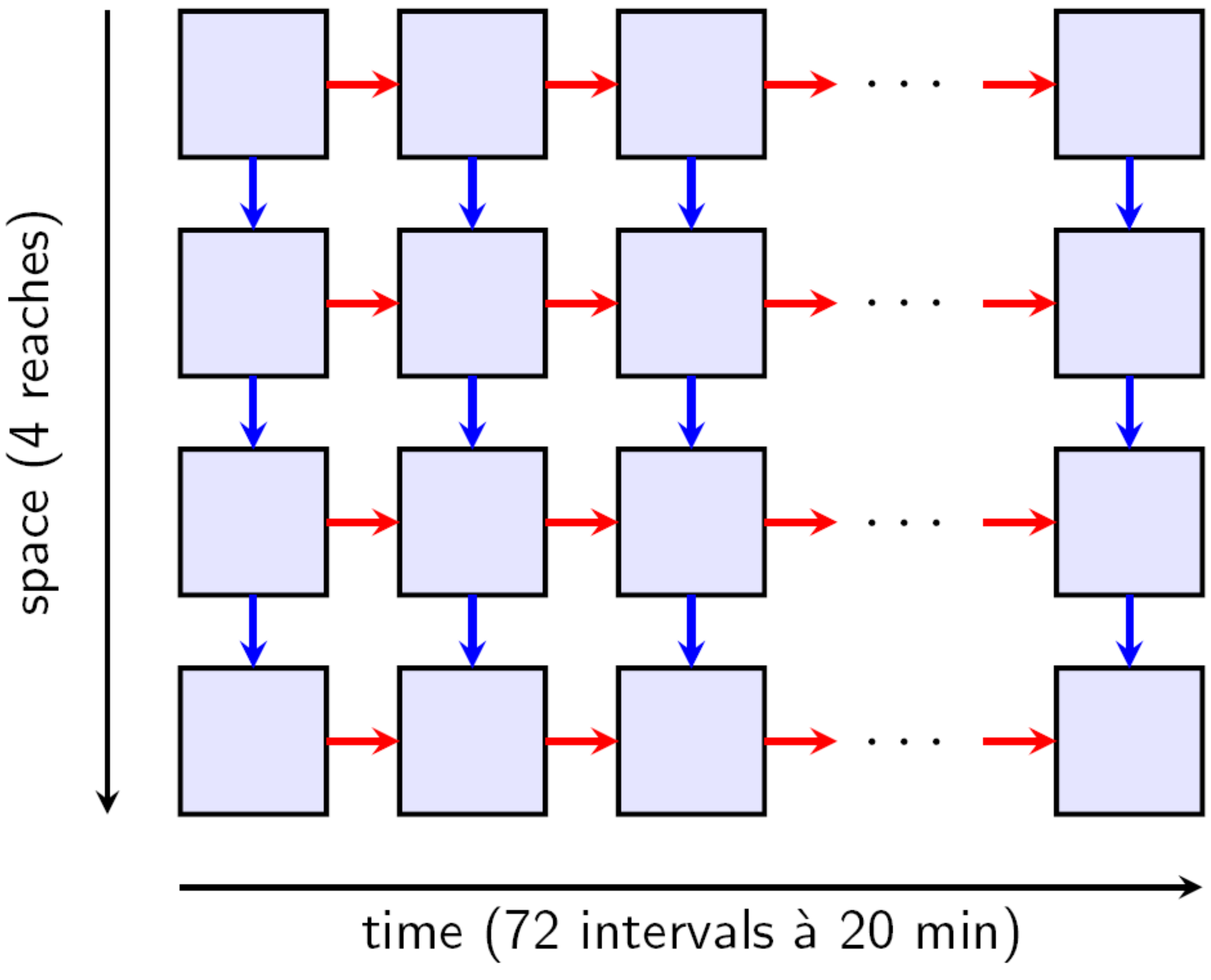


For ONE reach on ONE time interval:

- input $X_i \in \mathbb{R}^{12}$: 10 initial states + 2 water inflow coefficients
- control input $u_i \in \mathbb{R}$: constant turbine discharge
- output $x_i \in \mathbb{R}^{12}$: 10 end states + 2 outflow coefficients
- about 2000 hidden variables (steps of ACADO Integrator)

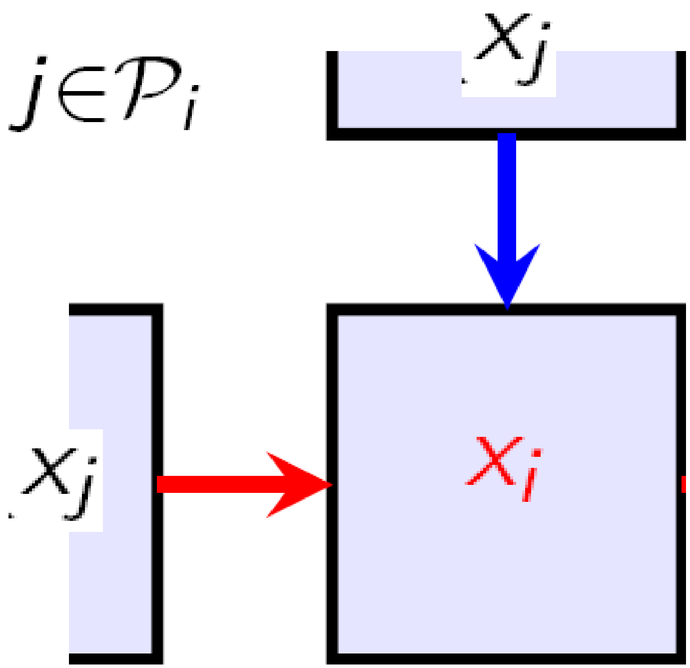
All Simulation Boxes for Optimal Control Problem

The overall system consists of $N = 288$ coupled simulation boxes



Coupling inputs X_i are outputs x_j of other boxes

Total system state: $x^T = (x_1^T, \ldots, x_N^T)$. Box input X_i is shorthand notation for outputs from **parents** $\mathcal{P}_i$, i.e. $X_i = (x_j)_{j \in \mathcal{P}_i}$.



Parents are neighbours in space (upper reach) or in time (previous interval).
Simulation box solves initial value problem with given boundary conditions.

Large Scale Nonlinear Program (NLP)

Each simulation box $x_i = \phi_i(X_i, u_i)$ also evaluates an objective $f_i(X_i, u_i)$ and inequality constraints $g_i(X_i, u_i)$.

$$
\begin{aligned}
\text{minimize}_{x,u} \quad & \sum_{i=1}^{N} f_i(X_i, u_i) \\
\text{subject to} \quad & \phi_i(X_i, u_i) - x_i = 0, \\
& g_i(X_i, u_i) \le 0, \quad i = 1, \dots, N.
\end{aligned}
$$

Note: coupling constraints only feasible in solution!
Simultaneous **method for simulation and optimization.**

We do "Multidimensional Multiple Shooting"

Generalization of **direct multiple shooting** [Bock and Plitt, 1984], widespread tool for dynamic optimization. People use multiple shooting because it

- allows us to use adaptive integrators
- leads to favourably structured NLPs
- is easy to parallelize
- allows to initialize all states
- can treat unstable systems well
- often shows faster convergence than single shooting, cf. the analysis of **lifted Newton methods** in [Albersmeyer and D., SIAM Opt, 2010]

In addition, our spatio-temporal NLP decomposition

- solves the problem of co-simulation of different subsystems.

Sequential Convex Programming (SCP)

Assuming f_i, g_i convex and known to central optimizer, can linearize simulation boxes at linearization points $\bar{X}_i, \bar{u}_i$.

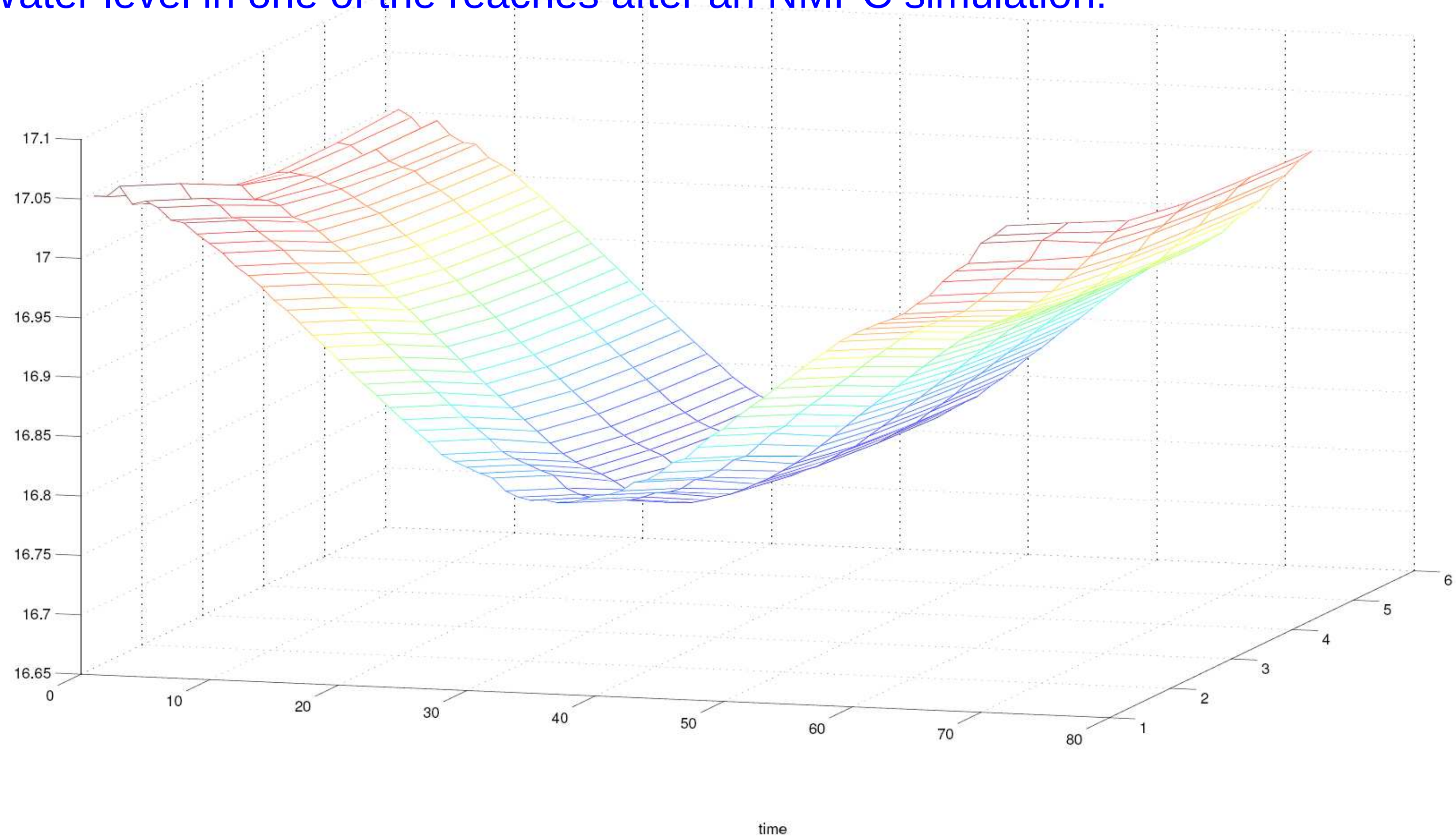
$$\begin{aligned} \text{minimize}_{x,u} \quad & \sum_{i=1}^{N} f_i(X_i, u_i) \\ \text{subject to} \quad & \boxed{\phi_i(\bar{X}_i, \bar{u}_i) + \frac{\partial \phi_i(\bar{X}_i, \bar{u}_i)}{\partial(X, u)} \begin{bmatrix} X_i - \bar{X}_i \\ u_i - \bar{u}_i \end{bmatrix}} - x_i = 0 \\ & g_i(X_i, u_i) \leq 0, \; i \in [1, N]. \end{aligned}$$

Iteratively solving linearized convex problems for obtaining the next linearization point yields a generalization of SQP, **Sequential Convex Programming (SCP)**. Can prove linear convergence towards local minima [Necoara et al, CDC, 2009], [T. D. Quoc and MD, BFG, 2010].

NMPC Simulation of Hydro Power Valley [C. Savorgnan]

- Aim: track a sinusoidal power profile (two turbines, two reaches)
- use SCP algorithm with real-time iteration [D. 2001]

Water level in one of the reaches after an NMPC simulation:



Inexact SCP Method

Approximate $\frac{\partial \phi_i(\bar{X}_i, \bar{u}_i)}{\partial (X,u)}$ by cheaper A_i. Add gradient correction to objective.

$$\begin{aligned} \text{minimize}_{x,u} \quad & \sum_{i=1}^{N} f_i(X_i, u_i) + [X_i^T | u_i^T] \frac{\partial \phi_i(\bar{X}_i, \bar{u}_i)}{\partial (X,u)}^T \bar{\lambda}_i \\ \text{subject to} \quad & \phi_i(\bar{X}_i, \bar{u}_i) + A_i \begin{bmatrix} X_i - \bar{X}_i \\ u_i - \bar{u}_i \end{bmatrix} - x_i = 0, \\ & g_i(X_i, u_i) \leq 0, \; i \in [1, N]. \end{aligned}$$

Solution x^*, u^* and equality multipliers δ^* yield next linearization point $\bar{x}^+, \bar{u}^+$ and multiplier guess, $\bar{\lambda}^+ = \bar{\lambda} + \delta^*$.
Linear convergence proven [D., Walther, Bock, Kostina, OMS, 2009], [Quoc et al. 2010].

Overview

- Large Decomposable Systems: the Hydro Power Valley
- Multidimensional Multiple Shooting
- **Dual decomposition and online active set strategy**
- Optimizing MPC of mechatronic systems

How to solve a decomposable convex QP ?

$$\min_{\underline{x}_1,\ldots,\underline{x}_N} \quad \sum_{i=1}^{N} \frac{1}{2} \underline{x}_i^T Q_i \underline{x}_i + \underline{c}_i^T \underline{x}_i$$

$$\text{s.t.} \quad H_i \underline{x}_i \leq \underline{d}_i \quad i = 1, \ldots, n$$

$$\sum_{i=1}^{N} A_i \underline{x}_i = \underline{b}$$

Only equality constraints interconnect subsystems

→ dual decomposition might be beneficial

Distributed QP Solution with DQP [Attila Kozma]

- DQP: MPI / C++ Framework for Distributed QP solution
- Treats Lagrange dual of QP

$$\max_{\underline{\lambda}} \quad \sum_{i=1}^{N} \underbrace{\begin{pmatrix} \min\limits_{\underline{x}_i} & \left(\frac{1}{2} \underline{x}_i^T Q_i \underline{x}_i + \left(\underline{c}_i^T + \underline{\lambda}^T A_i \right) \underline{x}_i - \underline{\lambda}^T \frac{\underline{b}}{N} \right) \\ \text{s.t.} & H_i \underline{x}_i \leq \underline{d}_i \end{pmatrix}}_{P_i(\underline{\lambda})}$$

- Local problems $P_i(\underline{\lambda})$ solved in parallel, with qpOASES (→next slide)
- Use Nesterov's optimal gradient scheme

$$\underline{\lambda}_x^{(k)} = \underline{\lambda}_y^{(k-1)} - t \nabla f(\underline{\lambda}_y^{(k-1)})$$

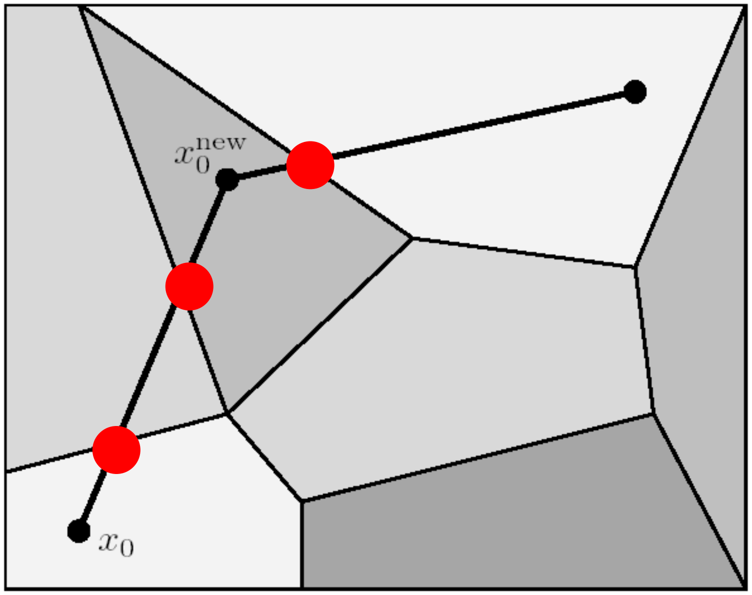
$$\underline{\lambda}_y^{(k)} = \underline{\lambda}_x^{(k)} + \frac{k-1}{k+2} \left(\underline{\lambda}_x^{(k)} - \underline{\lambda}_x^{(k-1)} \right)$$

- step depends on Hessians' smallest eigenvalue, complexity is $O(\frac{1}{\sqrt{\epsilon}})$

qpOASES: online QP solver

Only gradient in QP changes.
Solve p-QP via „Online Active Set Strategy“:

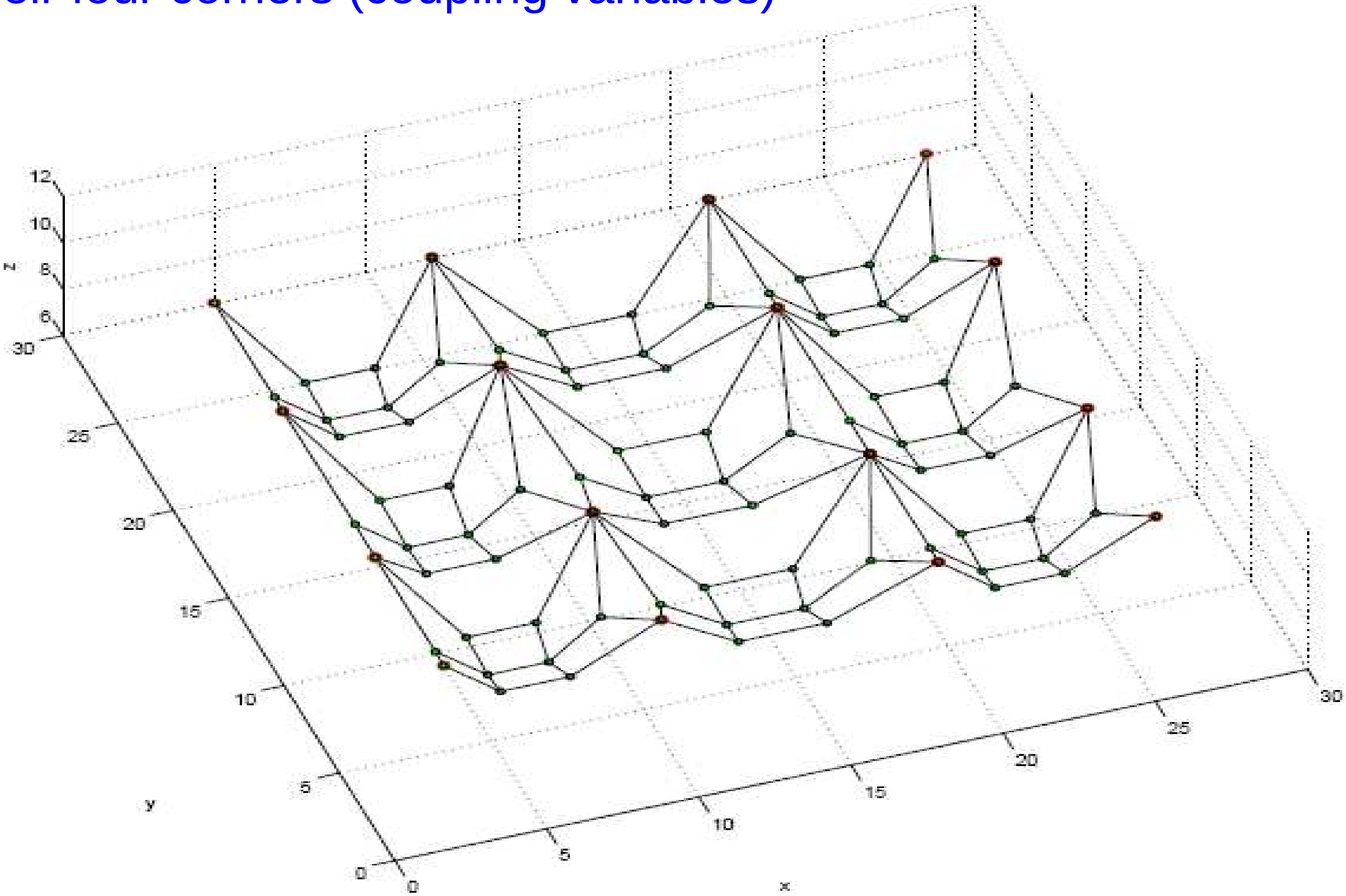
- go on straight line in parameter space from old to new problem data
- **solve each QP on path exactly (keep primal-dual feasibility)**
- Update matrix factorization at boundaries of critical regions
- Up to 10 x faster than standard QP



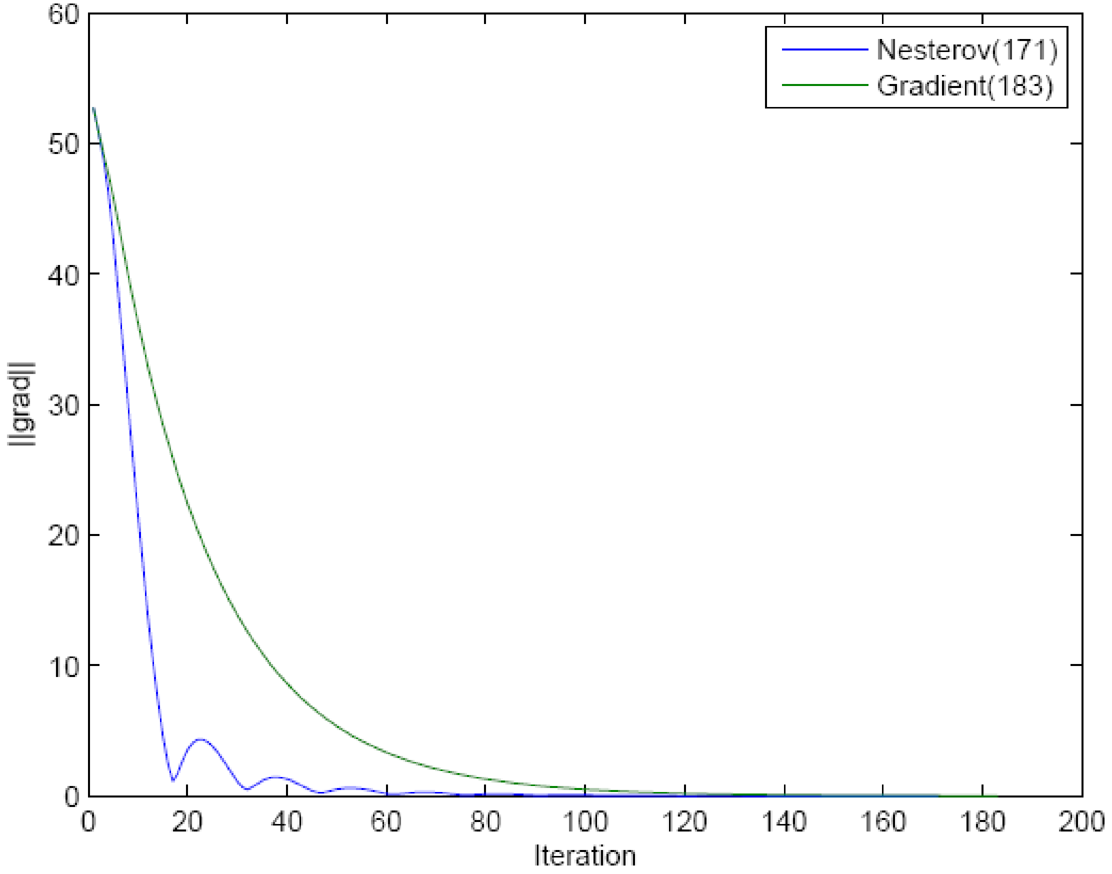
qpOASES: open source C++ code by Hans Joachim Ferreau

Static QP Test Example: Connected Hammocks

- Large QP arising from minimizing potential energy of "hammocks" that can touch the ground and are connected at their four corners (coupling variables)

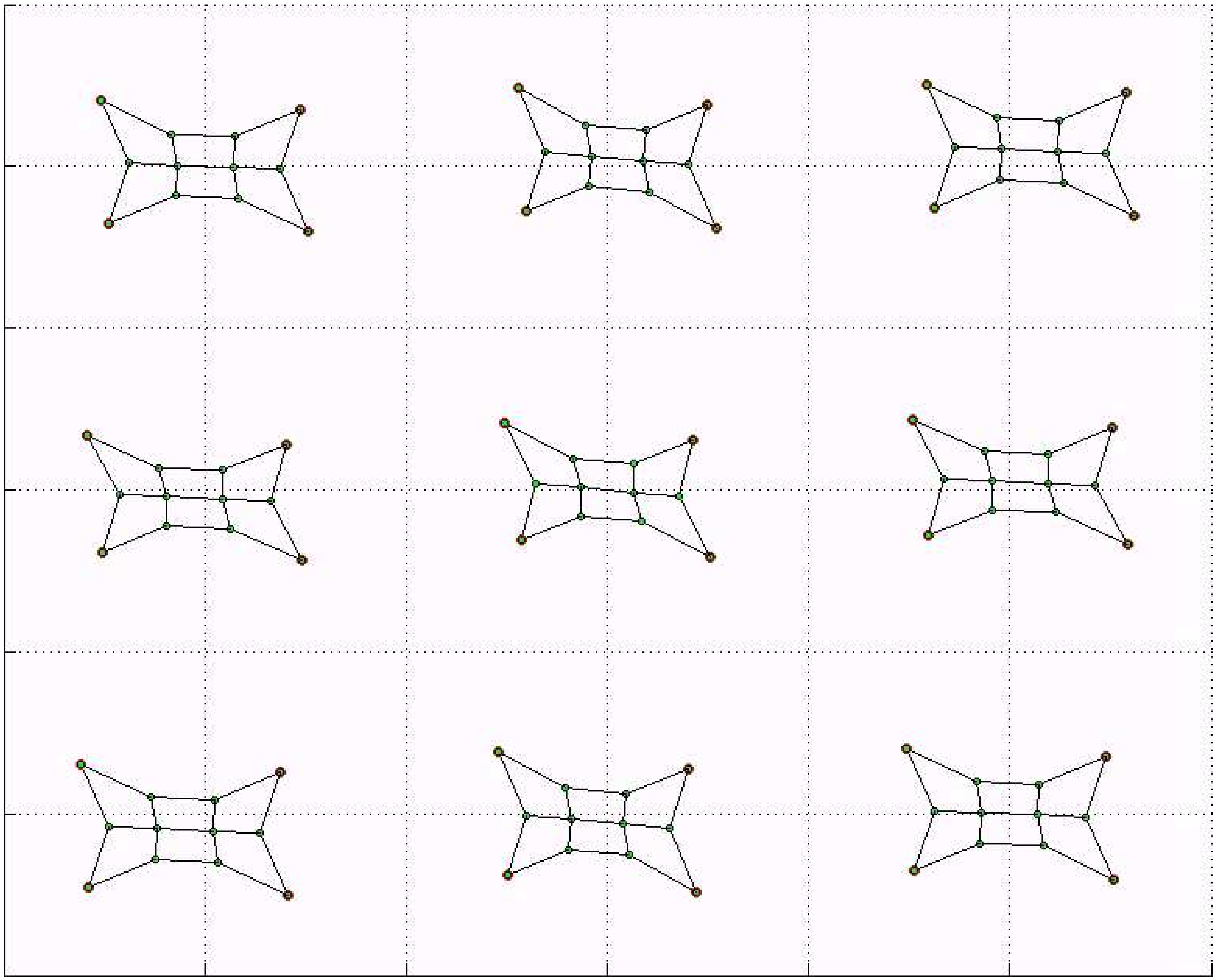


Comparison of Usual Gradient and Nesterov's Scheme



- Nesterov's optimal scheme is not monotonously decreasing the dual gradient, but converges much faster than pure gradient method ($\underline{\lambda}^{(k)} = \underline{\lambda}^{(k-1)} - t\nabla f(\underline{\lambda}^{(k-1)})$)

Hammocks Optimized with DQP



Runtime Comparison

Larger instance of test QP:

- 10 x 10 hammocks of each 10 x 10 mass points = 30 000 variables
- run on 101 cores of cluster from Flemish supercomputing center

Wall clock:

δ	Nesterov	Gradient
10^{-3}	0:55	02:58
10^{-4}	1:55	03:59
10^{-5}	2:52	04:56
10^{-6}	3:29	05:52

Runtime Comparison

Larger instance of test QP:

- 10 x 10 hammocks of each 10 x 10 mass points = 30 000 variables
- run on 101 cores of cluster from Flemish supercomputing center

Wall clock:

δ	Nesterov	Gradient
10^{-3}	0:55	02:58
10^{-4}	1:55	03:59
10^{-5}	2:52	04:56
10^{-6}	3:29	05:52

Attention: same problem takes 0:03 seconds on a single CPU when solved with a sparse IP method (OOQP from S. Wright).

Problem of all gradient methods: no second order information, slow linear convergence. Better parallelize IP solver! (→ C. Laird)

Workshop Invitation

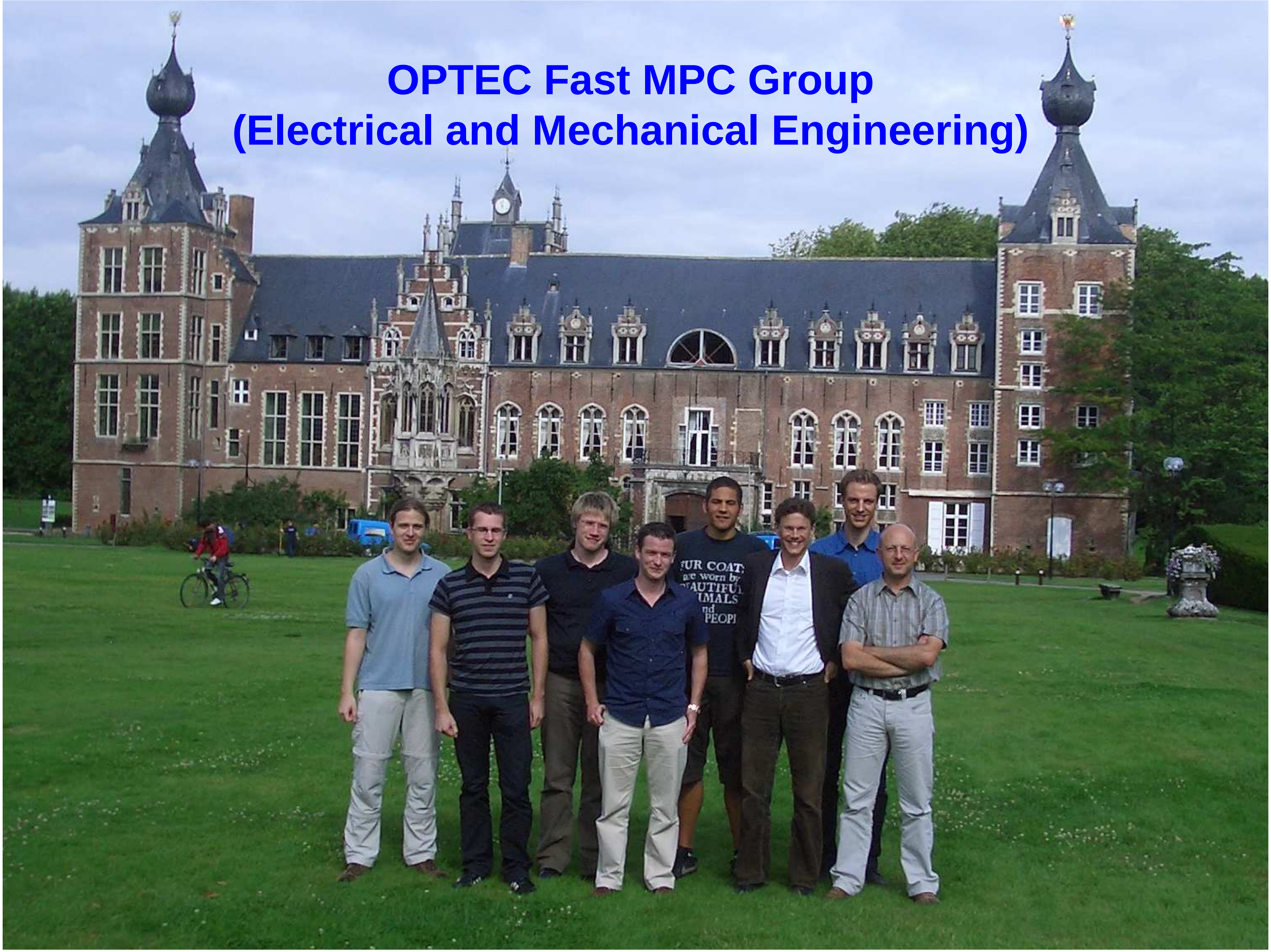
OPTEC Workshop on
Large Scale Convex Quadratic Programming
November 25-26, 2010,
Leuven

- Invited Keynote Speakers:
  - Y. Nesterov (fast gradient methods)
  - M. Saunders (sparse active set strategies)
  - S. Wright (sparse interior point methods)
- Topics:
  - decomposition algorithms
  - hierarchical and distributed QP solutions
  - fast gradient methods
  - structure exploiting interior-point and active set methods
  - parallel linear algebra
  - engineering applications

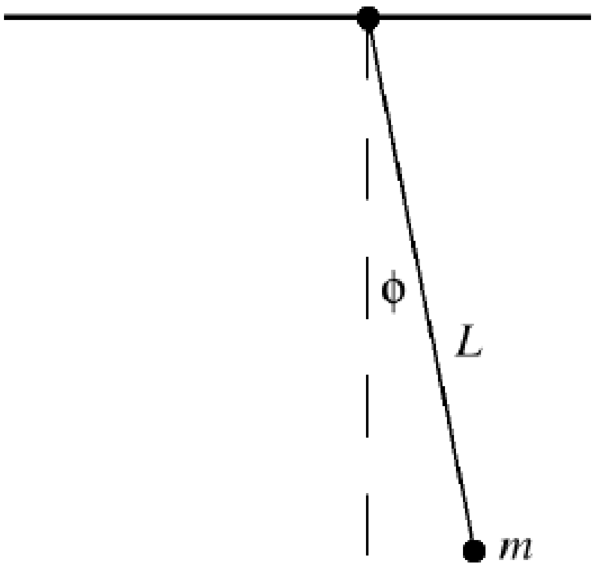
Overview

- Large Decomposable Systems: the Hydro Power Valley
- Multidimensional Multiple Shooting
- Dual decomposition and online active set strategy
- **Optimizing MPC of mechatronic systems**

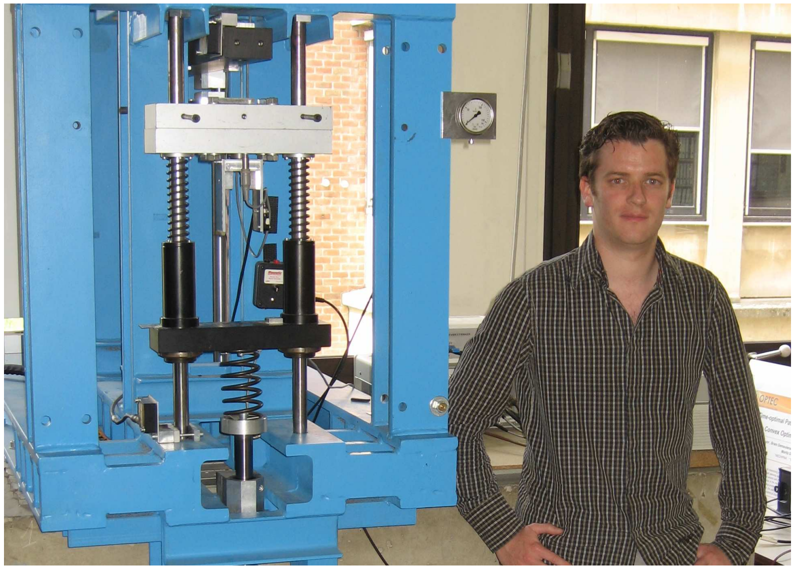
OPTEC Fast MPC Group
(Electrical and Mechanical Engineering)



Time Optimal MPC: a 60 Hz Application

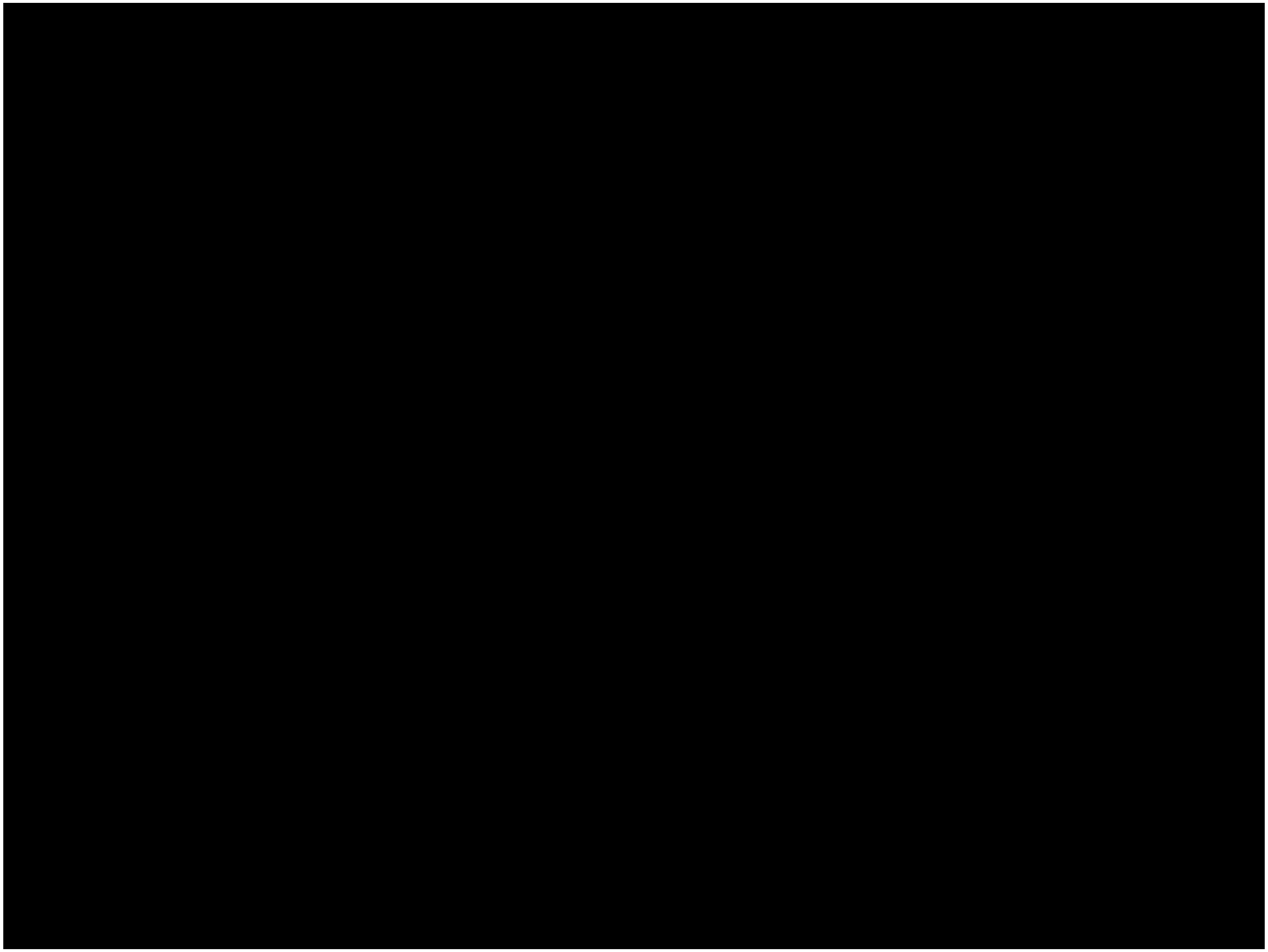


- Overhead crane
- MPC Aim: settle at any new setpoint in ***in minimal time***
- Two level algorithm: MIQP
  - 6 online data
  - 40 variables + one integer
  - 242 constraints (in-&output)
- use qpOASES on dSPACE
- **CPU time: <10 ms**

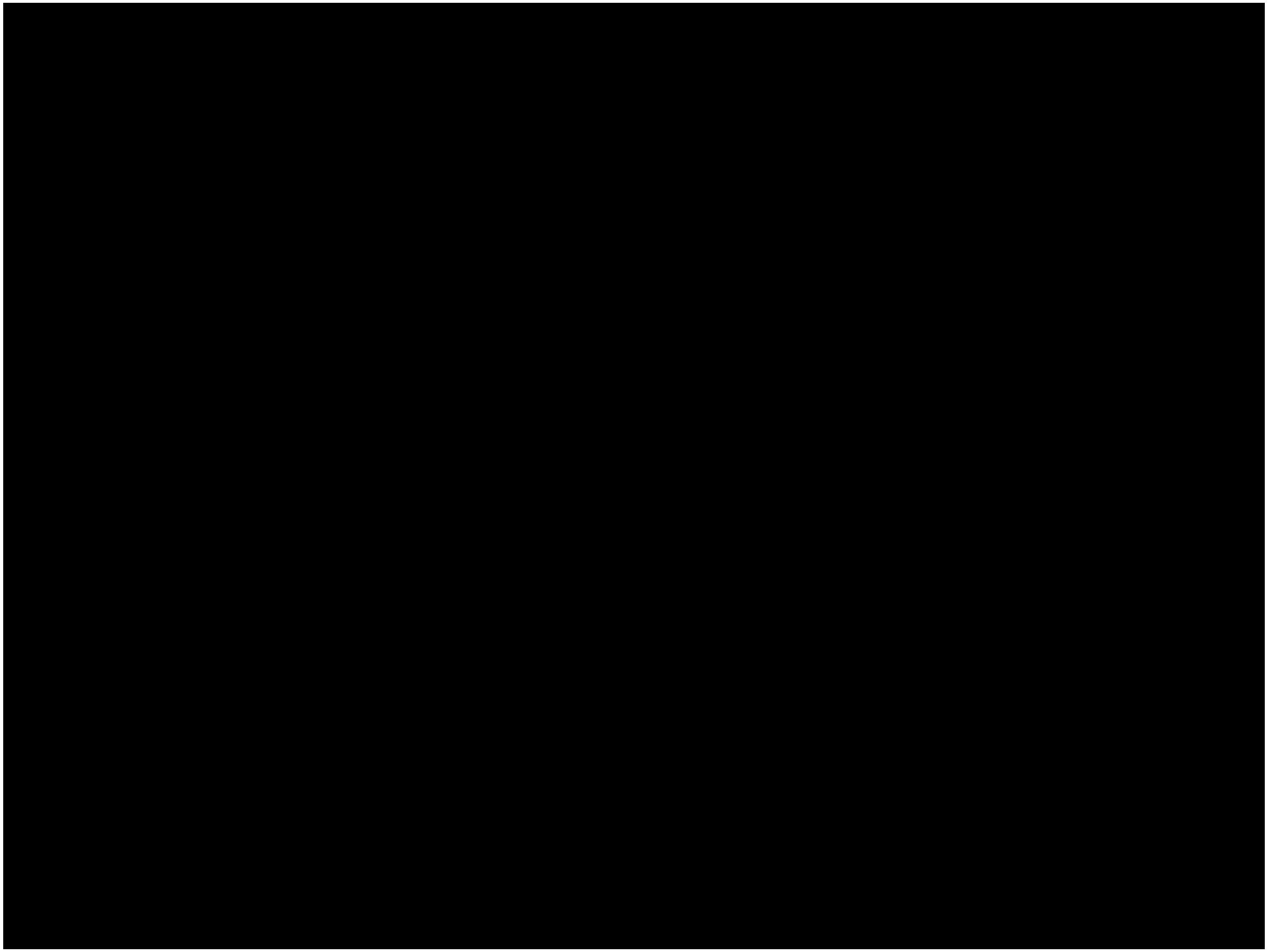


Lieboud Van den Broeck in mechanical engineering department

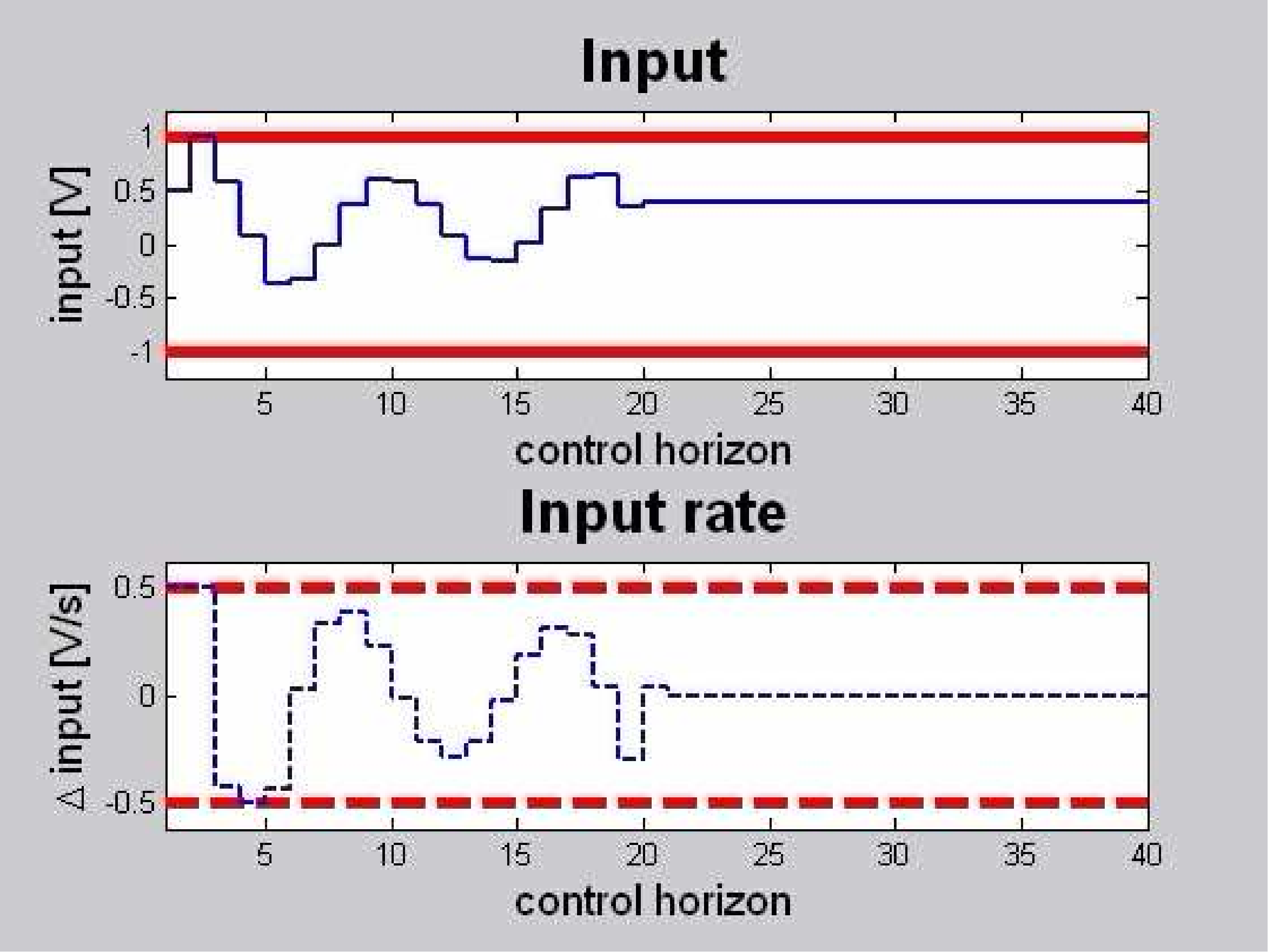
Overhead crane: feedback response



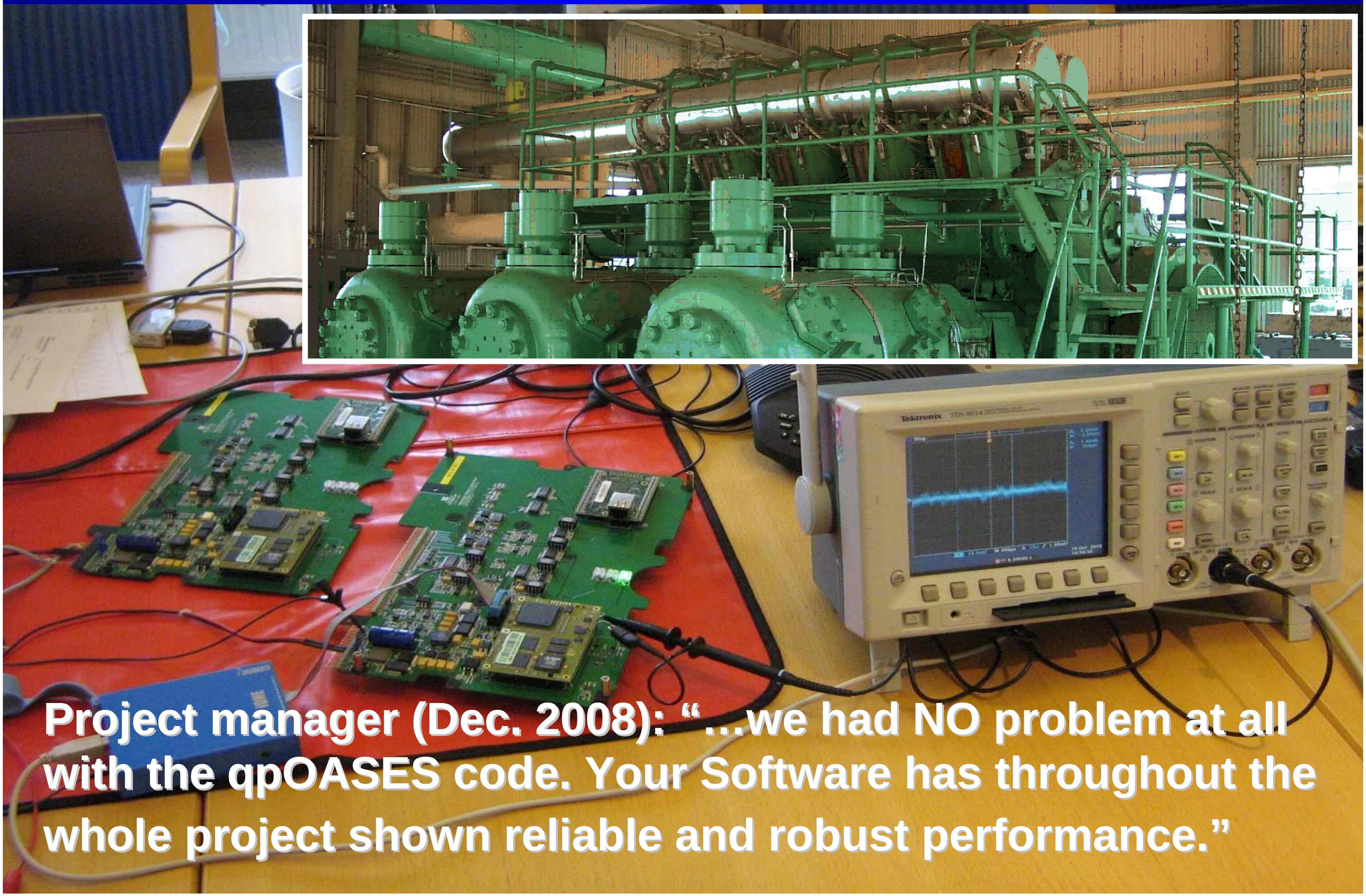
Overhead crane: multiple set point changes



Time Optimal MPC: qpOASES Optimizer Contents



qpOASES running on Industrial Control Hardware (20 ms)

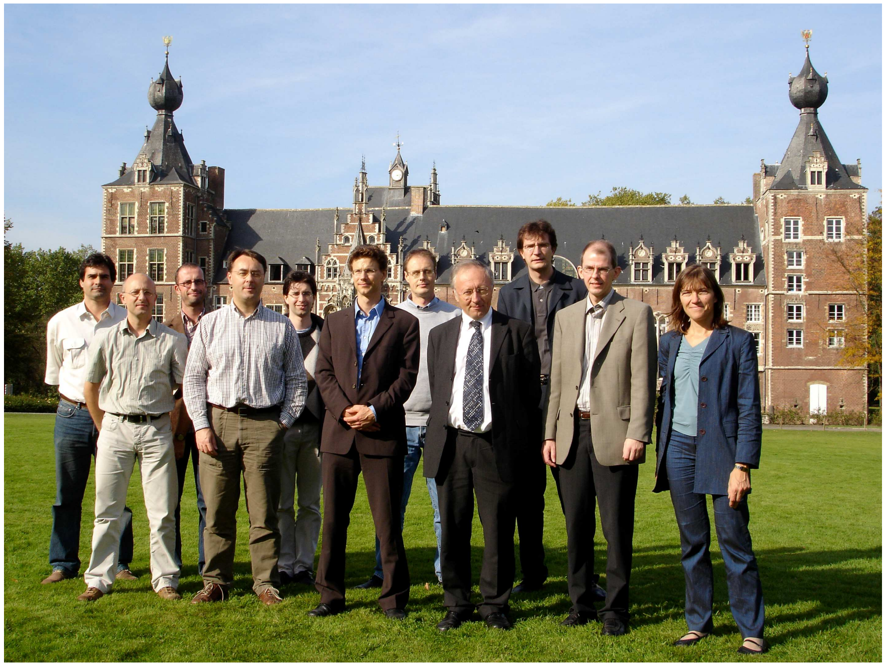


Project manager (Dec. 2008): "…we had NO problem at all with the qpOASES code. Your Software has throughout the whole project shown reliable and robust performance."

Summary

- Interconnected nonlinear systems allow "simulation box" decomposition: **multidimensional multiple shooting**, with exact or inexact SCP
- Dual decomposition often suffers from slow (sub)linear convergence
- OPTEC develops open source software for embedded optimization (ACADO, qpOASES, DQP, ...)
- Lots of exciting applications in engineering that need ultra-fast real-time optimization algorithms (convex and non-convex)

Open Positions at Optimization in Engineering Center OPTEC, Leuven



- OPTEC successfully secured funding until 2017
- Three open positions in MD's group:
  - **Embedded Optimization for Control (EMBOCON)**
  - **Distributed Optimization (HD-MPC)**
  - **Modelling and Optimal Control of Kite Energy Systems**

Indoors Test Flights, Leuven, April 2010

Software and Hardware Environment of DQP

- Software used within DQP:

  C/C++

  MPI (OpenMPI): Middleware framework for message passing between processes (distributed memory)

  μBLAS: C++ matrix arithmetics using templates

  qpOASES: Implementation of online active set strategy in C++

- Computations on Flemish Supercomputer Center:

  112 nodes with two quad-core Xeon 5420 2.5GHz CPUs
  80 nodes with two quad-core Xeon 5560 2.8GHz CPUs

  SUSE Linux Enterprise Server 10 SP2 (x86_64)

Why are inexact derivatives interesting ?

- derivative $\frac{\partial \phi_i(\bar{X}_i, \bar{u}_i)}{\partial (X,u)}$ is a large dense matrix, expensive to compute (cf. Schur-complements in C. Laird's talk)
- often, only few strongly coupling variables X_i^A in $X_i = (X_i^A, X_i^B)$, so can cheaply approximate derivative:

$$\left[\frac{\partial \phi_i}{\partial X_i^A} \quad | \quad \frac{\partial \phi_i}{\partial X_i^B} \quad | \quad \frac{\partial \phi_i}{\partial u_i} \right] \approx \left[\frac{\partial \phi_i}{\partial X_i^A} \quad | \quad 0 \quad | \quad \frac{\partial \phi_i}{\partial u_i} \right] =: A_i$$

- evaluate gradient correction $\frac{\partial \phi_i(\bar{X}_i, \bar{u}_i)}{\partial (X,u)}^T \bar{\lambda}_i$ by reverse differentiation, only 4 times more expensive than simulation $\phi_i(X_i, u_i)$. One single **extended simulation box** call.
- Less communication: variables x^B and multipliers λ^B only passed between child and parent nodes. Central optimizer works with **aggregate model** in x^A and u only.

Slight improvement without guarantee: L-BFGS

Limited Memory BFGS on parallel cluster:

δ	CPU-time	Runtime	Iterations
10^{-3}	00:01:16	00:00:04	10
10^{-4}	00:01:24	00:00:05	13
10^{-5}	00:01:24	00:00:09	14
10^{-6}	00:01:26	00:00:05	15

As fast as sparse QP, but with 101 CPUs!

Surely better to parallelize the sparse QP solver (cf. C. Laird's talk)

Solution of Hydro Power Valley Problem

- Two control variables: turbine discharges
- Aim was to track a sinusoidal power profile