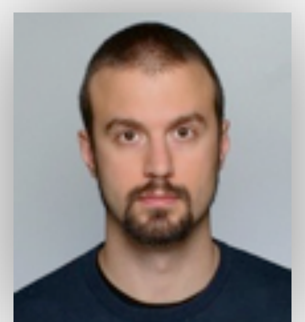
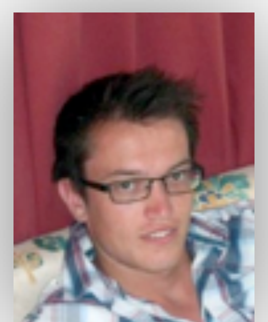
Fast Solvers for Nonlinear Optimal Control and Estimation

Milan Vukov Rien Quirynen Moritz Diehl

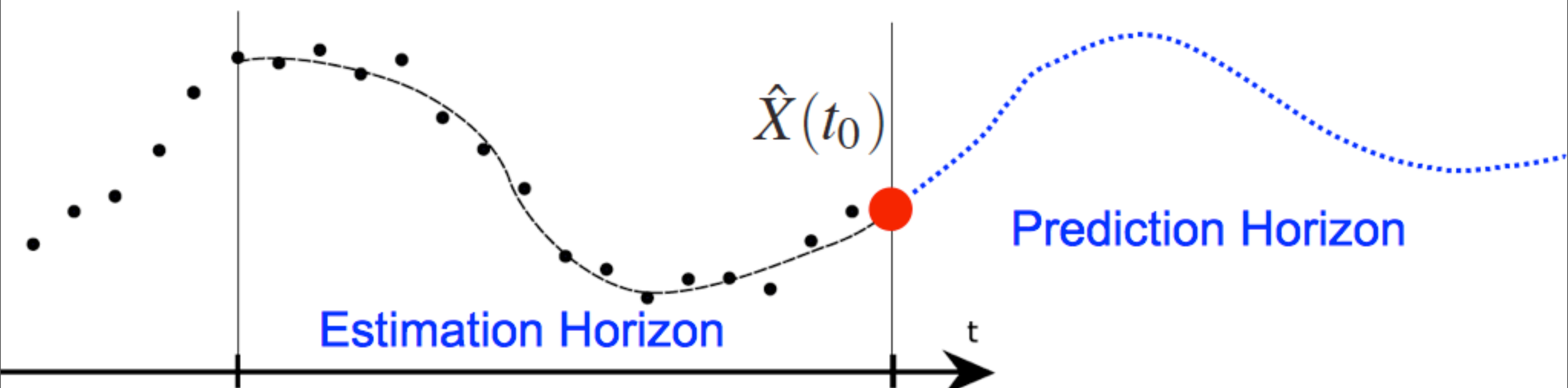




The Context

- Solvers for fast nonlinear MPC & MHE
- Embedded integrators
- Benchmarks & real-world applications
- Demonstration

Predictive Control: First Observe, then Decide

- Estimation: Obtain state by least squares fit of model to measurements
- Control: Obtain control input by solution of optimal control problem



Moving Horizon Estimation

Model Predictive Control

Two consecutive optimization problems!

Optimal Control Problem

$$\underset{x(\cdot),\,u(\cdot)}{\text{minimize}} \quad \int_{t_0}^{t_0+T} \|h(t,x(t),u(t)) - y(t)\|_W^2 \,\mathrm{d}t + \|h_{\mathrm{E}}(x(t_0+T)) - y(t_0+T)\|_{W_{\mathrm{E}}}^2$$

$$\begin{aligned} \text{subject to} \quad x(t_0) &= \bar{x}_0, \\ \dot{x}(t) &= f(t,x(t),u(t),z(t)), & \forall t \in [t_0, t_0+T], \\ 0 &\geq h(x(t),u(t)), & \forall t \in [t_0, t_0+T], \\ 0 &\geq r(x(t_0+T)), \end{aligned}$$

Optimal Control Problem (OCP)

$$
\begin{aligned}
\underset{x(\cdot),\,u(\cdot)}{\text{minimize}} \quad & \int_{t_0}^{t_0+T} \|h(t,x(t),u(t)) - y(t)\|_W^2 \,\mathrm{d}t \\
& + \|h_{\mathrm{E}}(x(t_0+T)) - y(t_0+T)\|_{W_{\mathrm{E}}}^2 \\
\text{subject to} \quad & x(t_0) = \bar{x}_0, \\
& \dot{x}(t) = f(t,x(t),u(t),z(t)), \qquad \forall t \in [t_0, t_0+T], \\
& 0 \geq h(x(t),u(t)), \qquad \forall t \in [t_0, t_0+T], \\
& 0 \geq r(x(t_0+T)),
\end{aligned}
$$

Discretization

- single/multiple shooting
- collocation

Nonlinear Program (NLP)

$$\begin{aligned} \underset{X,U}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=0}^{N-1} \|h_i(x_i,u_i)-y_i\|_W^2 + \|h_N(x_N)-y_N\|_{W_{\mathrm{E}}}^2 \\ \text{subject to} \quad & \color{red}{0 = x_0 - \bar{x}_0,} \\ & 0 = x_{i+1} - F_i(x_i,u_i,z_i), \quad i=0,\ldots,N-1, \\ & 0 \geq h_i(x_i,u_i), \quad i=0,\ldots,N-1, \\ & 0 \geq r(x_N), \end{aligned}$$

Integrator output

Solution methods

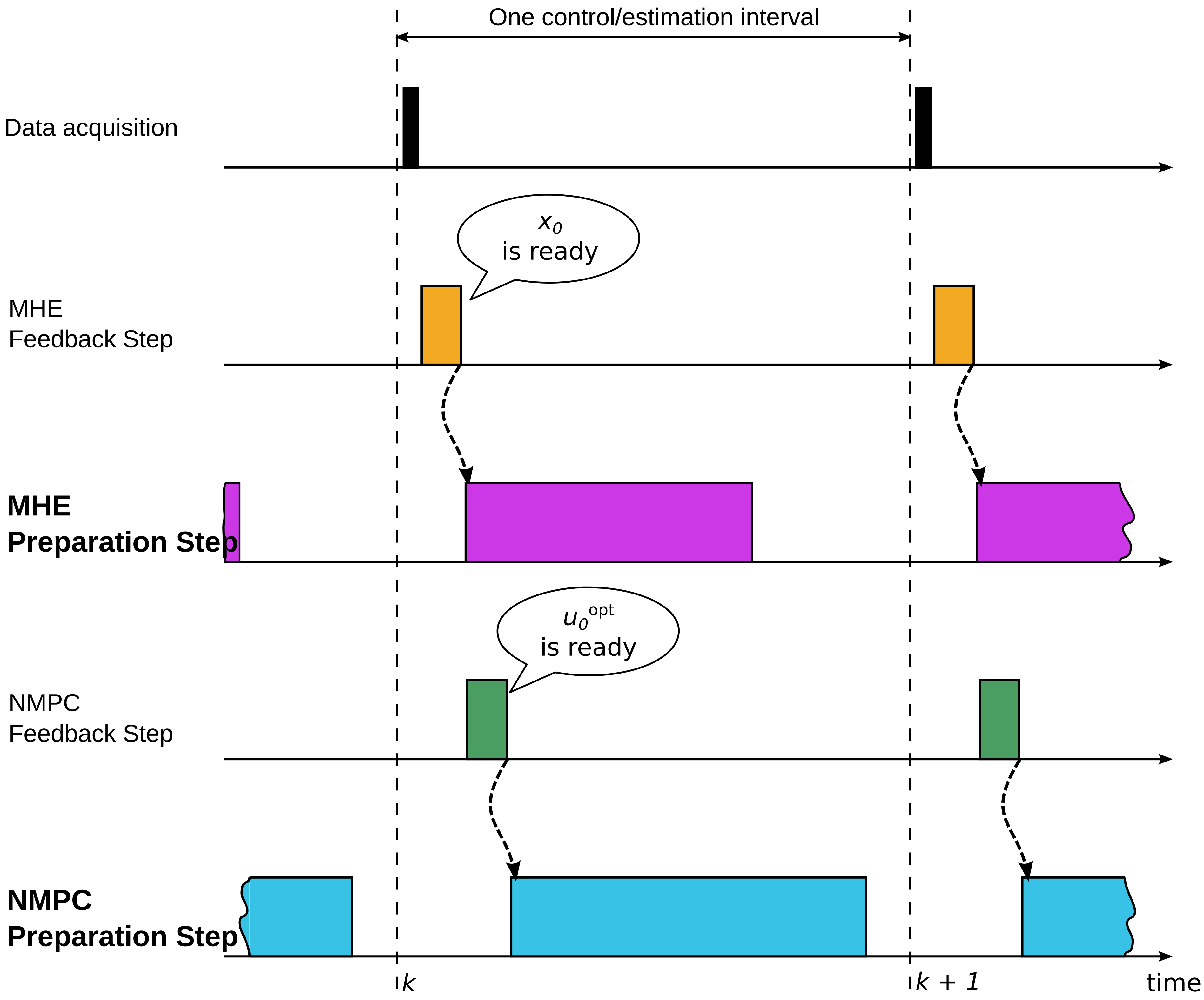
Real-time Iterations *

- Problem discretization - single/multiple shooting **
- Least squares objective - employ Gauss-Newton method
- Perform only one SQP iteration per sampling time
- Optionally *condense* a sparse QP
- Division into preparation and feedback phase
  & Initial Value Embedding

* Diehl2002
** Bock1984

$$
\begin{aligned}
\underset{X,U}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=0}^{N-1} \|h_i(x_i,u_i) - y_i\|_W^2 + \|h_N(x_N) - y_N\|_{W_{\mathrm{E}}}^2 \\
\text{subject to} \quad & 0 \; = \; x_0 - \bar{x}_0, \\
& 0 \; = \; x_{i+1} - F_i(x_i,u_i,z_i), \quad i = 0,\ldots,N-1, \\
& 0 \; \geq \; h_i(x_i,u_i), \qquad\qquad i = 0,\ldots,N-1, \\
& 0 \; \geq \; r(x_N),
\end{aligned}
$$

RTI Scheme



One control/estimation interval
Data acquisition
x_0 is ready
MHE Feedback Step
MHE Preparation Step
u_0^{opt} is ready
NMPC Feedback Step
NMPC Preparation Step
k
$k + 1$
time

The ingredients

- Algorithmic differentiation & function evaluation
- Integrators
- OCP structure exploitation
- QP solvers

Embedded integrators

Runge-Kutta methods

Explicit

$$\dot{x}(t) = f(t, x(t), u(t))$$

- non-stiff system
- explicit ODEs

Implicit

$$0 = f(t, \dot{x}(t), x(t), z(t), u(t))$$

- stiff system
- DAEs of index 1
- continuos output

$$x_{k+1} = \boldsymbol{\Phi_k}(\underbrace{x_k, u_k}_{w_k}) \quad \longrightarrow \quad 0 = \boldsymbol{\Phi_k}(\boldsymbol{\bar{w}_k}) - x_{k+1} + \frac{\boldsymbol{\partial \Phi_k}}{\boldsymbol{\partial w}}(\boldsymbol{\bar{w}_k})(w_k - \bar{w}_k)$$

Exploitation of linear subsystems*

Linear Input System

$$C_1\dot{x}^{[1]} = A_1 x^{[1]} + B_1 u$$

$$\downarrow$$

Nonlinear System

$$0 = f_2(x^{[1]}, x^{[2]}, \dot{x}^{[1]}, \dot{x}^{[2]}, z, u)$$

$$\downarrow$$

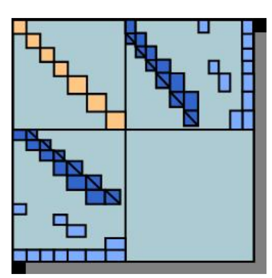
Linear Output System

$$C_3\dot{x}^{[3]} = A_3 x^{[3]} + f_3(x^{[1]}, x^{[2]}, \dot{x}^{[1]}, \dot{x}^{[2]}, z, u)$$

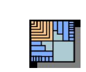
* Quirynen2013

Classical Condensing

$$
\begin{aligned}
\underset{x_0,u_0,\ldots,x_N}{\text{minimize}} \quad & \frac{1}{2}\sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q_k & S_k \\ S_k^T & R_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} g_k^{\mathrm{x}} \\ g_k^{\mathrm{u}} \end{bmatrix}^T \\
& \qquad + \tfrac{1}{2} x_N^T Q_{\mathrm{e}} x_N + x_N^T g_{\mathrm{e}}^{\mathrm{x}} \\
\text{subject to} \quad & x_{k+1} = A_k x_k + B_k u_k + c_k, \quad \text{for} \quad k = 0, \ldots, N-1 \\
& x_k^{\mathrm{lo}} \leq x_k \leq x_k^{\mathrm{up}}, \quad \text{for} \quad k = 0, \ldots, N, \\
& u_k^{\mathrm{lo}} \leq u_k \leq u_k^{\mathrm{up}}, \quad \text{for} \quad k = 0, \ldots, N-1, \\
& b_k^{\mathrm{lo}} \leq C_k x_k + D_k u_k \leq b_k^{\mathrm{up}}, \quad \text{for} \quad k = 0, \ldots, N-1, \\
& b_{\mathrm{e}}^{\mathrm{lo}} \leq C_{\mathrm{e}} x_N \leq b_{\mathrm{e}}^{\mathrm{up}},
\end{aligned}
$$



$$
\begin{aligned}
\underset{u}{\text{minimize}} \quad & \frac{1}{2} u^T H_{\mathrm{C}} u + u^T g_{\mathrm{C}} \\
\text{subject to} \quad & u^{\mathrm{lo}} \leq u \leq u^{\mathrm{up}} \\
& b_{\mathrm{C}}^{\mathrm{lo}} \leq A_{\mathrm{C}} u \leq b_{\mathrm{C}}^{\mathrm{up}}
\end{aligned}
$$



... and employ dense linear algebra QP solver, e.g. qpOASES

Classical Condensing

states → controls

$$\Delta w_2 = d + C\Delta s_0 + E\Delta w_1$$

$$C = \begin{bmatrix} C_0 \\ C_1 \\ \cdots \\ C_{N-1} \end{bmatrix}, \quad E = \begin{bmatrix} E_{0,0} & & & \\ E_{0,1} & E_{1,1} & & \\ \vdots & \vdots & \ddots & \\ E_{0,N-1} & & \cdots & E_{N-1,N-1} \end{bmatrix}$$

$O(N^3)$

$$H_c = E'\overline{Q}E + \overline{R}$$

$$g_c = E'\overline{Q}d + E'g^s + g^u + E'\overline{Q}C\Delta s_0$$

Exploit the structure!*

$$C = \begin{bmatrix} C_0 \\ C_1 \\ \dots \\ C_{N-1} \end{bmatrix}, \quad E = \begin{bmatrix} E_{0,0} & & & \\ E_{0,1} & E_{1,1} & & \\ \vdots & \vdots & \ddots & \\ E_{0,N-1} & & \dots & E_{N-1,N-1} \end{bmatrix}$$

Then cond. Hessian can be computed more efficiently

$$\frac{1}{2}N^3 \quad \longrightarrow \quad \frac{1}{6}N(N+1)(N+2)$$

* Leineweber1999

N^2 Condensing

... From another point of view ...

$$A\,x = B\,u + c \Leftrightarrow$$

$$\begin{bmatrix} I & & & \\ -A_1 & I & & \\ & \ddots & \ddots & \\ & & -A_{N-1} & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} B_0 & & & \\ & B_1 & & \\ & & \ddots & \\ & & & B_{N-1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} + \begin{bmatrix} A_0\,x_0 + c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{bmatrix}$$

$$\ldots \Rightarrow \quad H_{\mathrm{c}} = \overline{R} + B^T A^{-T}(\overline{Q}E) + \ldots$$

$$\ldots \Rightarrow \quad H_c = \overline{R} + B^T A^{-T}(\overline{Q}E) + \ldots$$

$$\underbrace{\begin{bmatrix} I & -A_1^{\mathrm{T}} & & \\ & I & \ddots & \\ & & \ddots & -A_{N-1}^{\mathrm{T}} \\ & & & I \end{bmatrix}}_{A^{\mathrm{T}}} T = \underbrace{\begin{bmatrix} E_{0,0}^Q & & & \\ E_{1,0}^Q & E_{1,1}^Q & & \\ \vdots & \vdots & \ddots & \\ E_{N-1,0}^Q & E_{N-1,1}^Q & \cdots & E_{N-1,N-1}^Q \end{bmatrix}}_{\bar{Q}E}$$

N^2 complexity

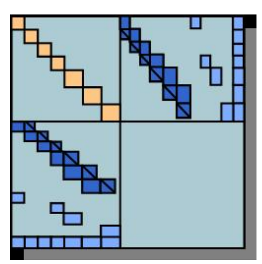
Note: overall complexity is still N^3 (QP solver)

* Frison2012, Andersson2013, Frison2013

And what about long horizons?

$$
\begin{aligned}
\underset{x_0,u_0,\dots,x_N}{\text{minimize}} \quad & \frac{1}{2}\sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q_k & S_k \\ S_k^T & R_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} g_k^{\mathrm{x}} \\ g_k^{\mathrm{u}} \end{bmatrix}^T \\
& + \tfrac{1}{2} x_N^T Q_{\mathrm{e}} x_N + x_N^T g_{\mathrm{e}}^{\mathrm{x}} \\
\text{subject to} \quad & x_{k+1} = A_k x_k + B_k u_k + c_k, \quad \text{for } k = 0,\dots,N-1 \\
& x_k^{\mathrm{lo}} \le x_k \le x_k^{\mathrm{up}}, \quad \text{for } k = 0,\dots,N, \\
& u_k^{\mathrm{lo}} \le u_k \le u_k^{\mathrm{up}}, \quad \text{for } k = 0,\dots,N-1, \\
& b_k^{\mathrm{lo}} \le C_k x_k + D_k u_k \le b_k^{\mathrm{up}}, \quad \text{for } k = 0,\dots,N-1, \\
& b_{\mathrm{e}}^{\mathrm{lo}} \le C_{\mathrm{e}} x_N \le b_{\mathrm{e}}^{\mathrm{up}},
\end{aligned}
$$

$$
\begin{aligned}
\underset{u}{\text{minimize}} \quad & \frac{1}{2} u^T H_{\mathrm{C}} u + u^T g_{\mathrm{C}} \\
\text{subject to} \quad & u^{\mathrm{lo}} \le u \le u^{\mathrm{up}} \\
& b_{\mathrm{C}}^{\mathrm{lo}} \le A_{\mathrm{C}} u \le b_{\mathrm{C}}^{\mathrm{up}}
\end{aligned}
$$



FORCES*

Structure exploiting QCQP solver

Implements primal-dual IP method

Auto-generated C-code

Linear complexity in *N* but cubic in number of states and controls

* Domahidi2012, http://forces.ethz.ch

HPMPC*

- Library of high-performance solvers for MPC
- Standalone library coded in C language
- Interior-point method based solvers
- Customizations for different CPUs (x86, ARM) and instruction sets.

* Frison2013, https://github.com/giaf/hpmpc

A Dual Newton Strategy*

C-code Software Implementation **qpDUNES**

$$\begin{aligned} \min_{z} \quad & \sum_{k=0}^{N} \left(\frac{1}{2} z_k^{\mathrm{T}} H_k z_k + g_k^{\mathrm{T}} z_k \right) \\ \text{s.t.} \quad & E_{k+1} z_{k+1} = C_k z_k + c_k && \forall\, k = 0, \dots, N-1 \\ & \underline{d}_k \le D_k z_k \le \overline{d}_k && \forall\, k = 0, \dots, N. \end{aligned}$$

$$\begin{aligned} \max_{\lambda} \quad \min_{z} \quad & \sum_{k=0}^{N} L_k(z_k, \lambda_k, \lambda_{k+1}) \\ \text{s.t.} \quad & \underline{d}_k \le D_k z_k \le \overline{d}_k \quad \forall k = 0, \dots, N \end{aligned}$$

- Local QPs can be solved in parallel
- Overall complexity is linear in *N*

* Ferreau2012, Frasch2014, https://github.com/jfrasch/qpDUNES

Exact Hessian based NMPC

$$
\begin{aligned}
\underset{x(\cdot),u(\cdot)}{\text{minimize}} \quad & \int_0^T \boldsymbol{l(t, x(t), u(t))}\,\mathrm{d}t \\
\text{subject to} \quad & x(0) = \bar{x}_0 \\
& \dot{x}(t) = f(t, x(t), u(t)) \\
& 0 \geq h(x(t), u(t)) \\
& 0 \geq r(x(0), x(T)) \\
& \forall t \in [0, T]
\end{aligned}
$$

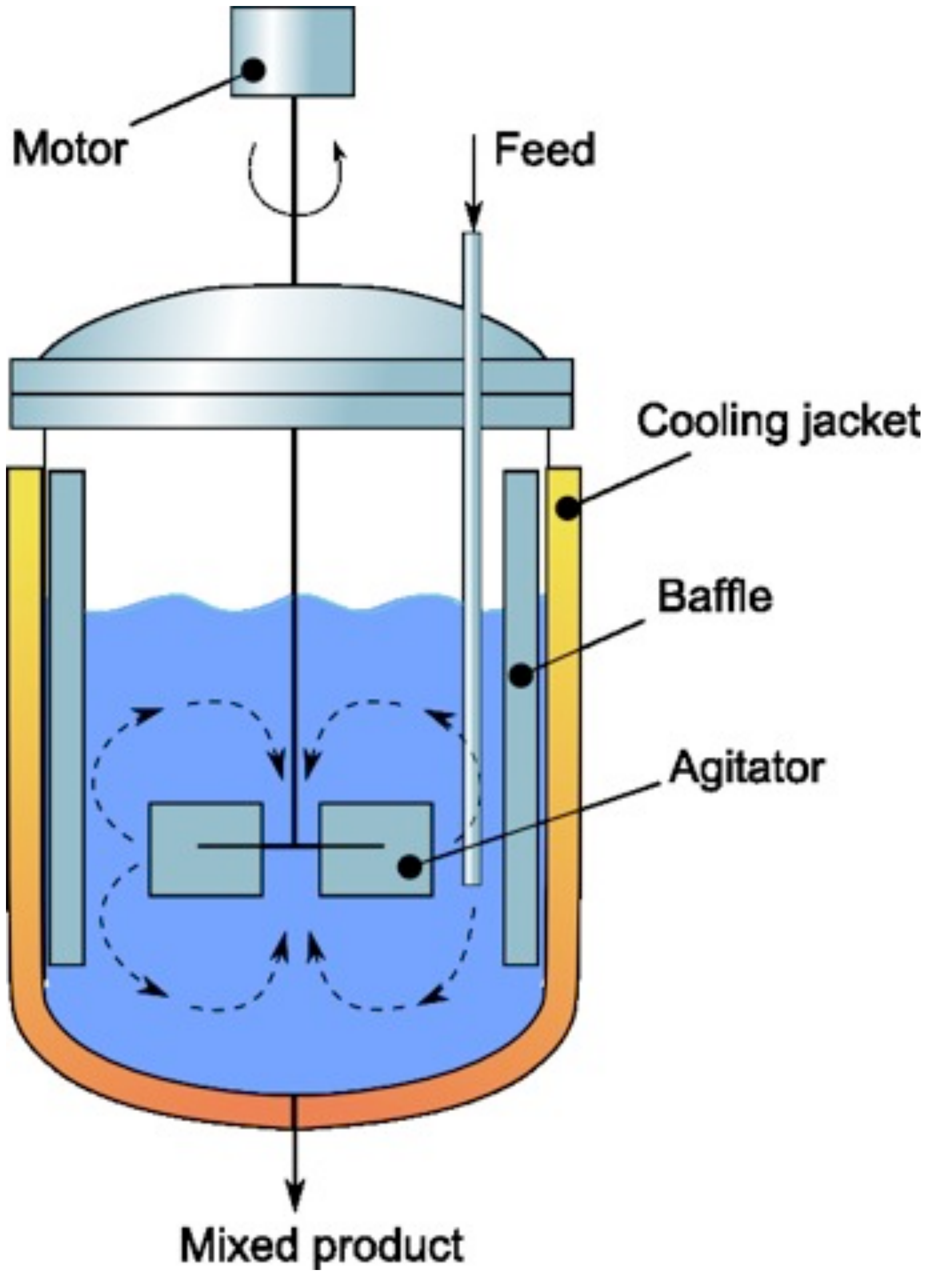
Symmetric Hessian propagation

- Forward-over-adjoint (FOA)
- Symmetric scheme (TSP) *
  - → less computations & memory

* Quirynen2014

Economic MPC bioreactor

- explicit ODE system
- 5 states, 1 input
- maximize average productivity



N = 20	FOA scheme	TSP scheme
integration	138 µs	64 µs
condensing	15 µs	15 µs
regularization	16 µs	16 µs
qpOASES	60 µs	60 µs
RTI step	229 µs	155 µs

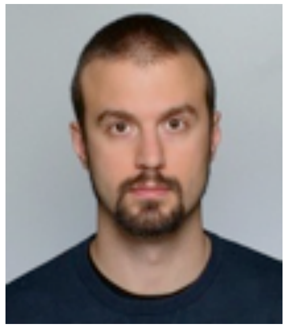
ACADO toolkit *

www.acadotoolkit.org

- Open source package (LGPL)
- Depends only on the STL C++ library
- Multi-platform: Linux, OS X, Windows
- MATLAB & Simulink interfaces

- Optimal control of dynamic systems
- State and parameter estimation
- Feedback control based on MPC/MHE
- Fast implementations for RT execution: ACADO Code Generation tool



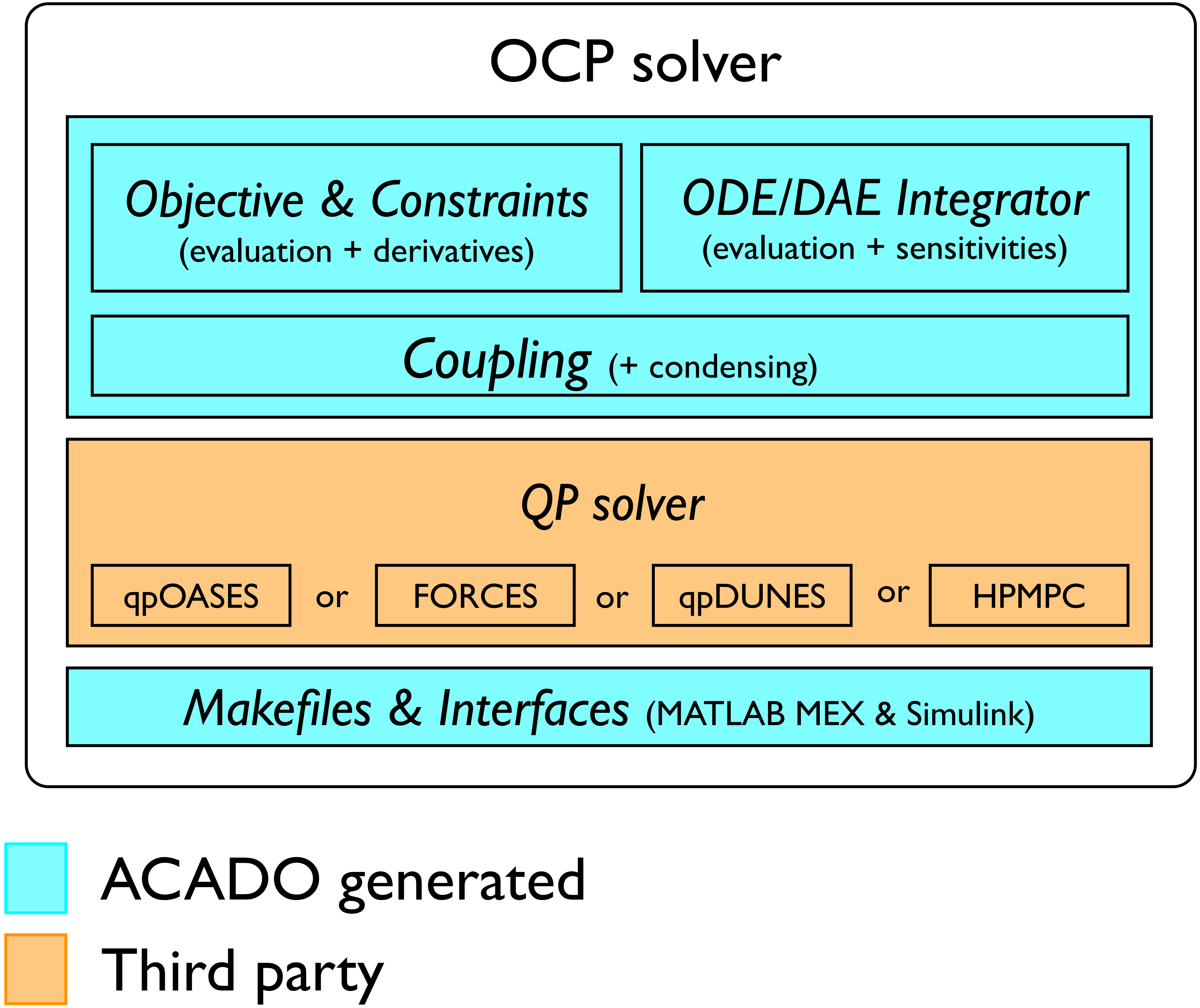



* Houska2009

ACADO Code Generation Tool *

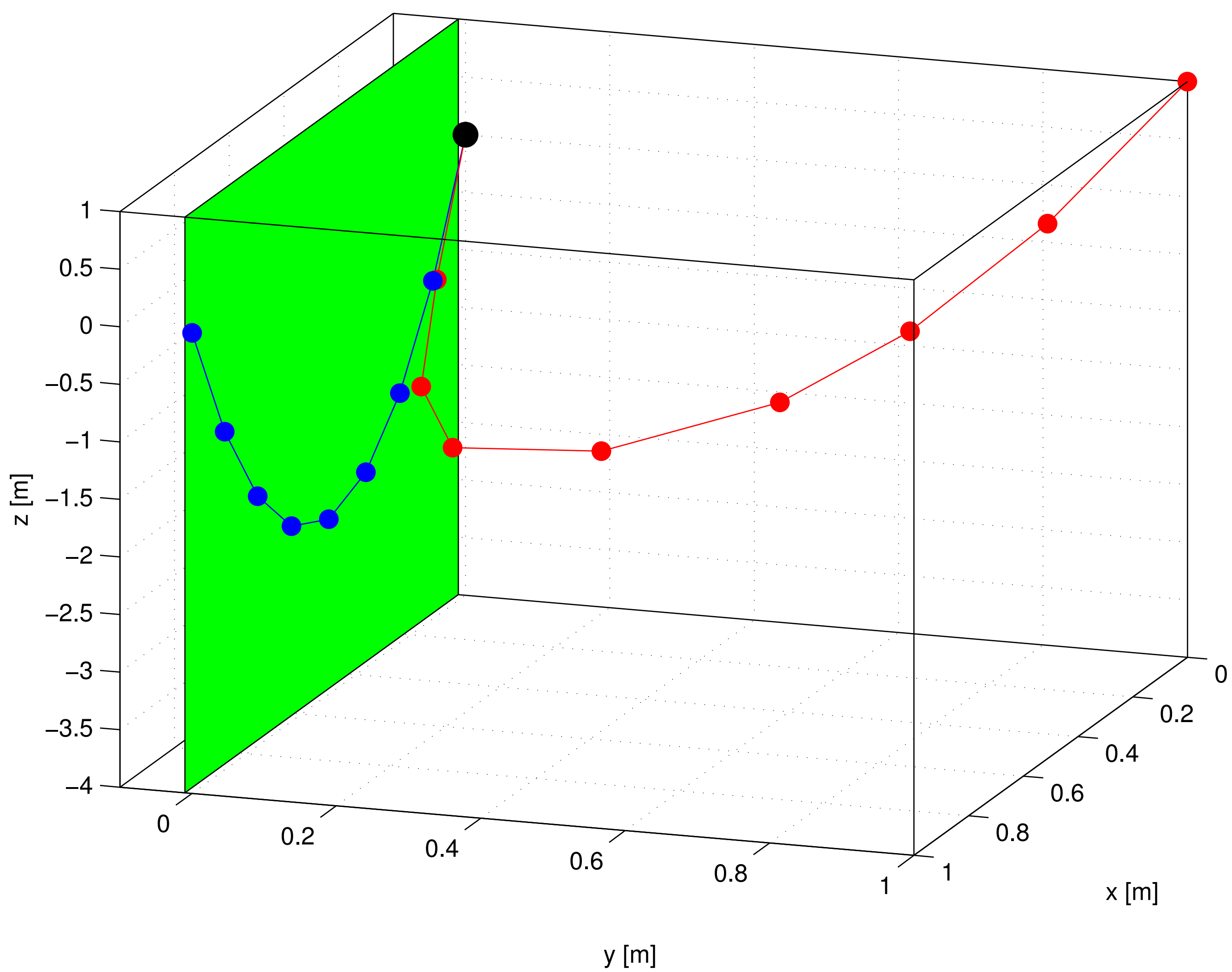
- Optimize the number of evaluations of the right-hand-side of ODE/DAE and its derivatives.
- Use tailored fixed-step Runge-Kutta integrators
- Avoid dynamic memory allocation
- Minimize branching in the exported code
- Export optimized linear algebra routines
- Interfaces to MATLAB & Simulink
- OpenMP support for multiple shooting

* Houska2011, Ferreau2012, Quirynen2012, Vukov2012, Quirynen2013, Vukov2013


OCP solver
Objective & Constraints
(evaluation + derivatives)
ODE/DAE Integrator
(evaluation + sensitivities)
Coupling (+ condensing)
QP solver
qpOASES
or
FORCES
or
qpDUNES
or
HPMPC
Makefiles & Interfaces (MATLAB MEX & Simulink)
ACADO generated
Third party

Benchmarks

- A chain of masses problem



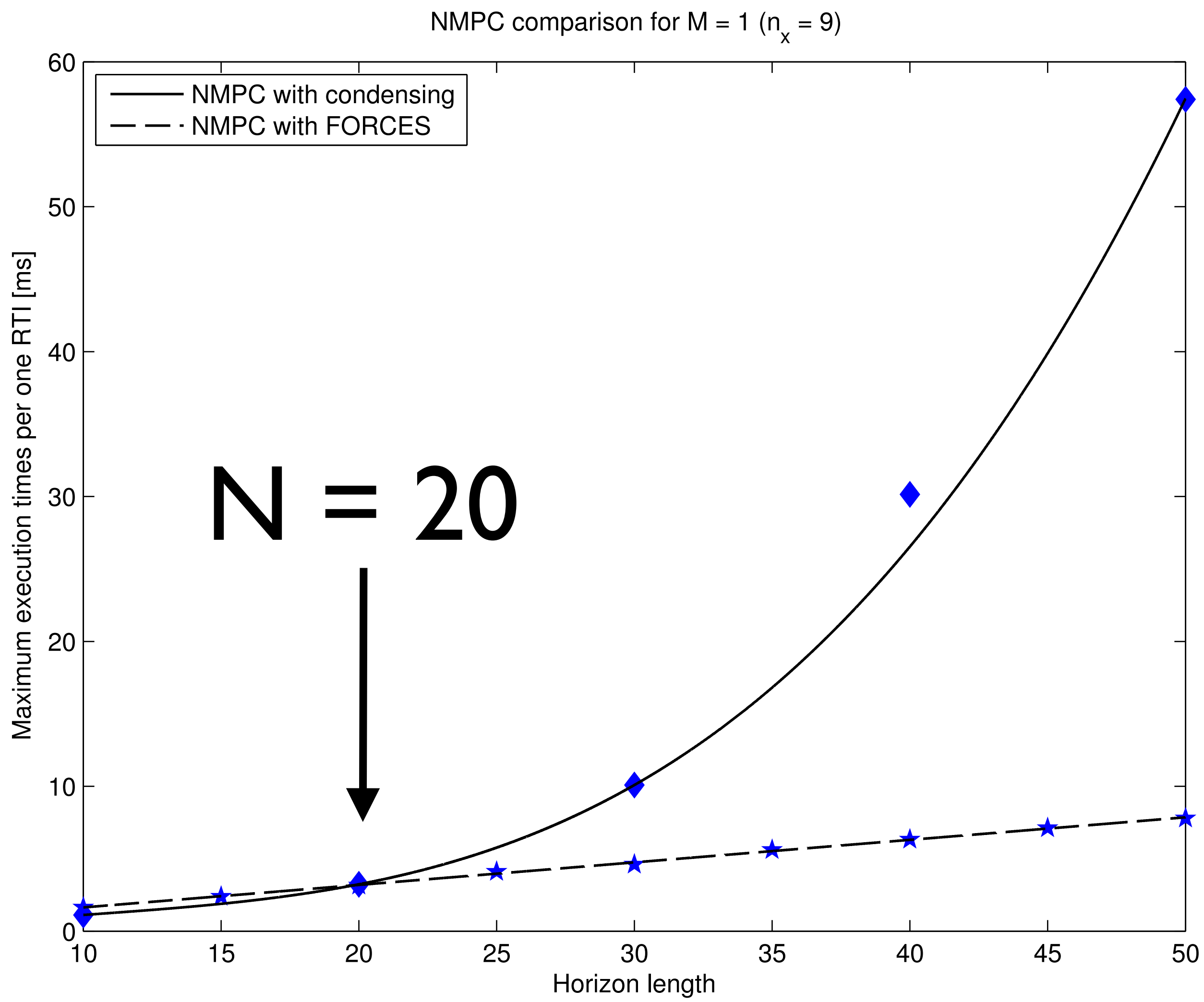
M masses

3(2M + 1) states

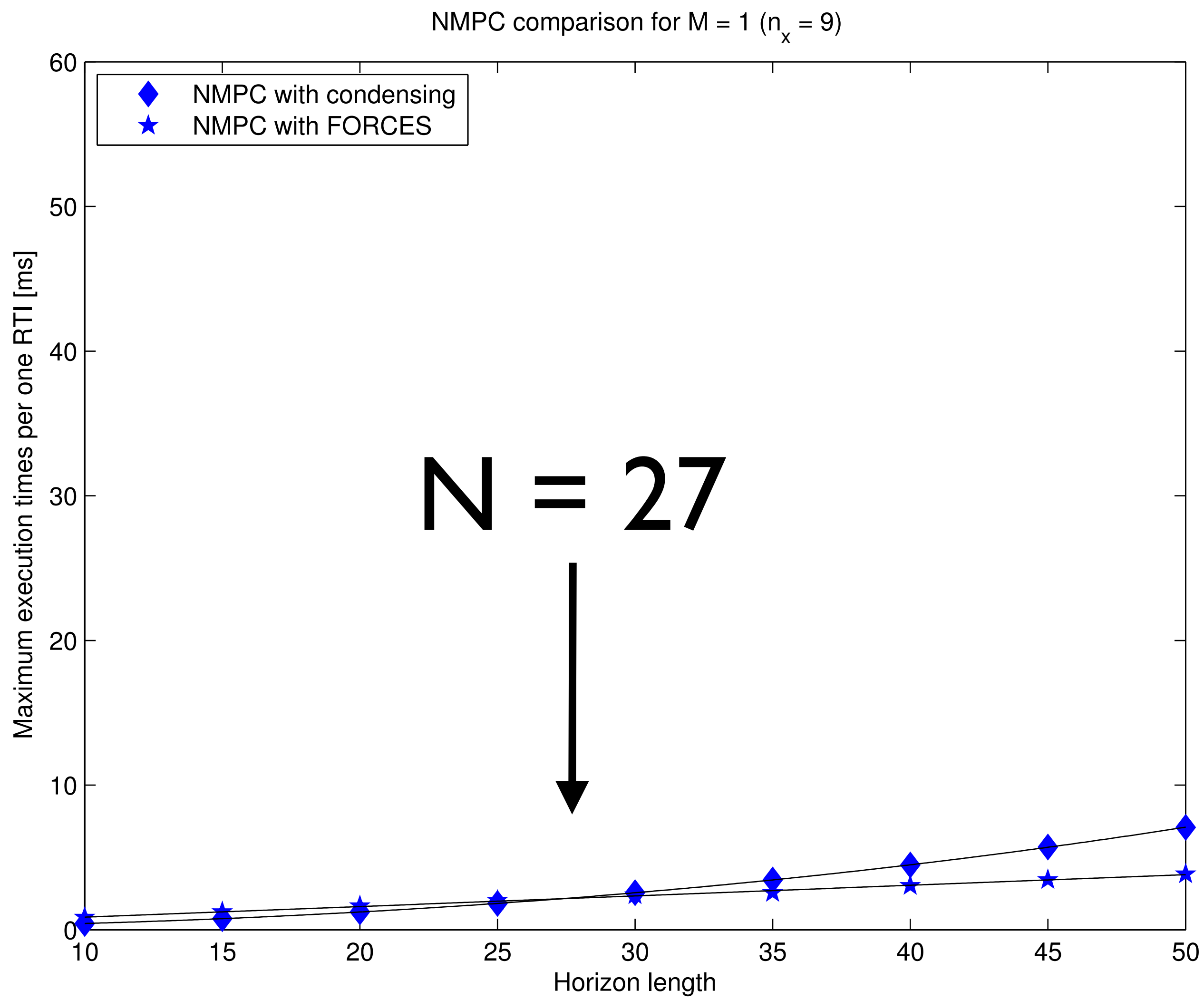
3 controls

*Wirsching2006

$M = 1, N = 10 .. 50; n_x = 9, n_u = 3$



Old

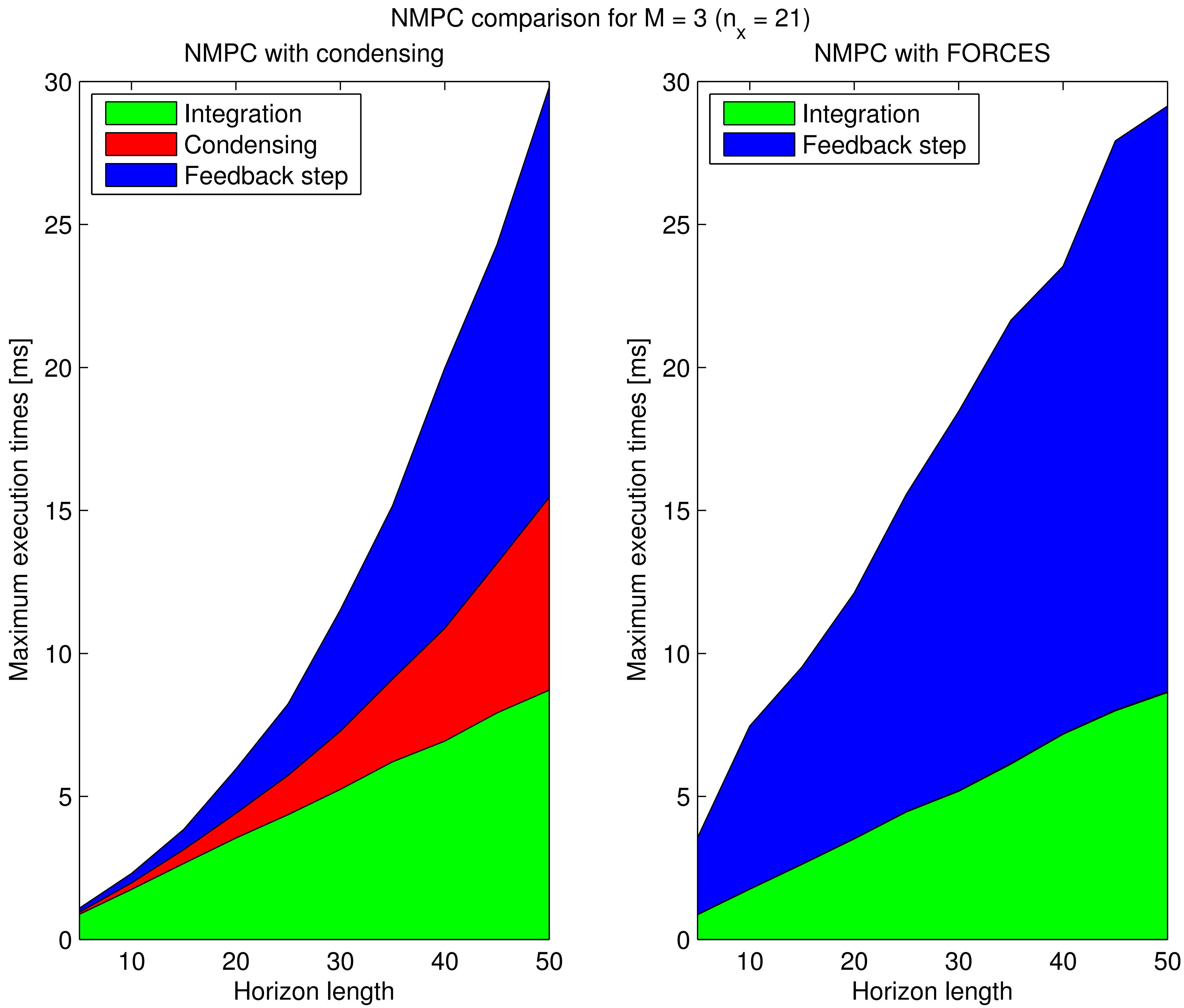


New

Improvements: x6 for condensing & x2 for FORCES

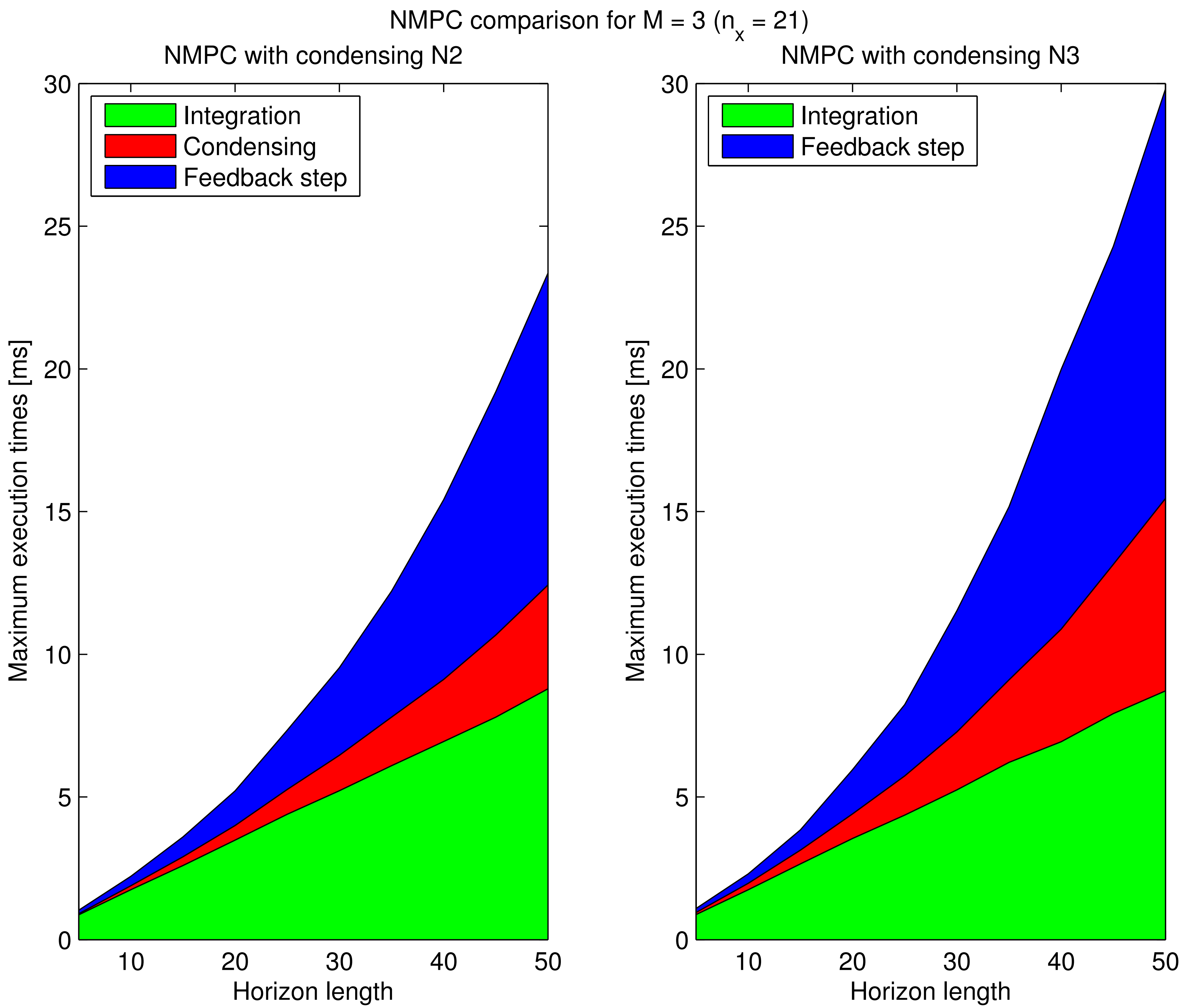
N^3 condensed vs sparse NMPC

$n_x = 21, n_u = 3$

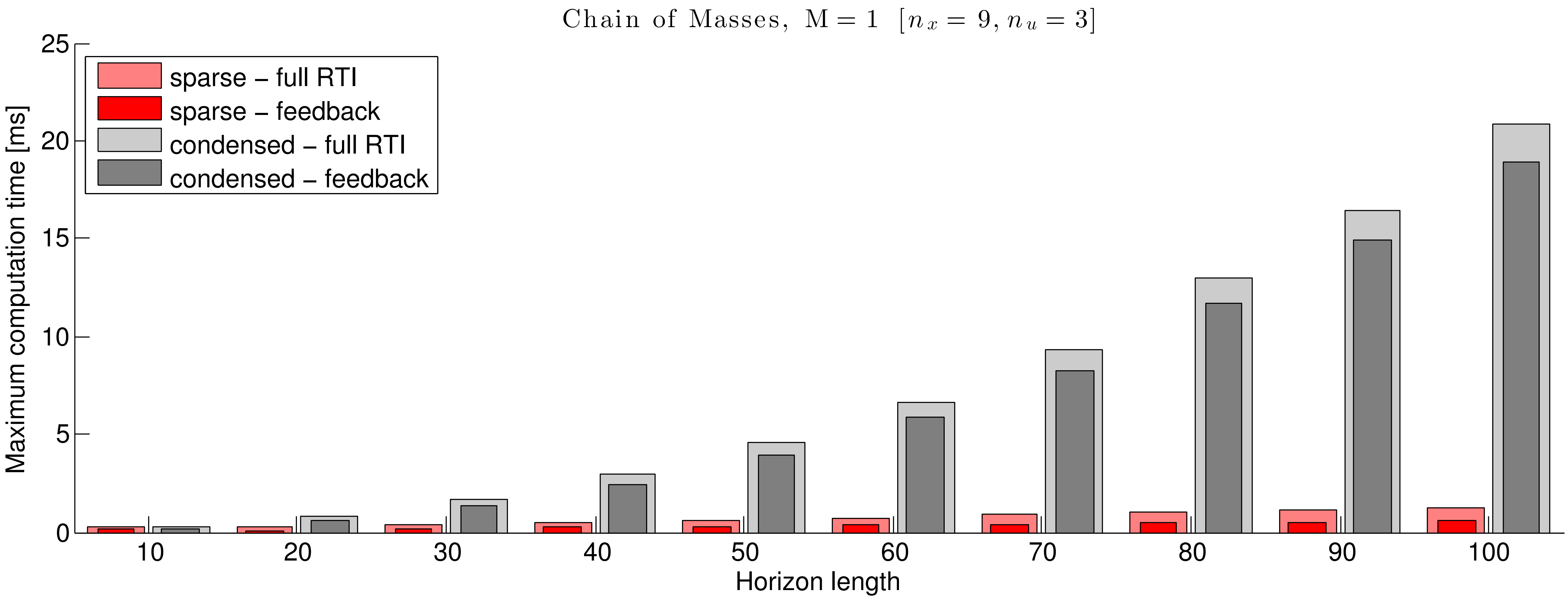


N^2 Condensed vs N^3 Cond. NMPC

$n_x = 21, n_u = 3$



qpDUNES vs qpOASES
based NMPC



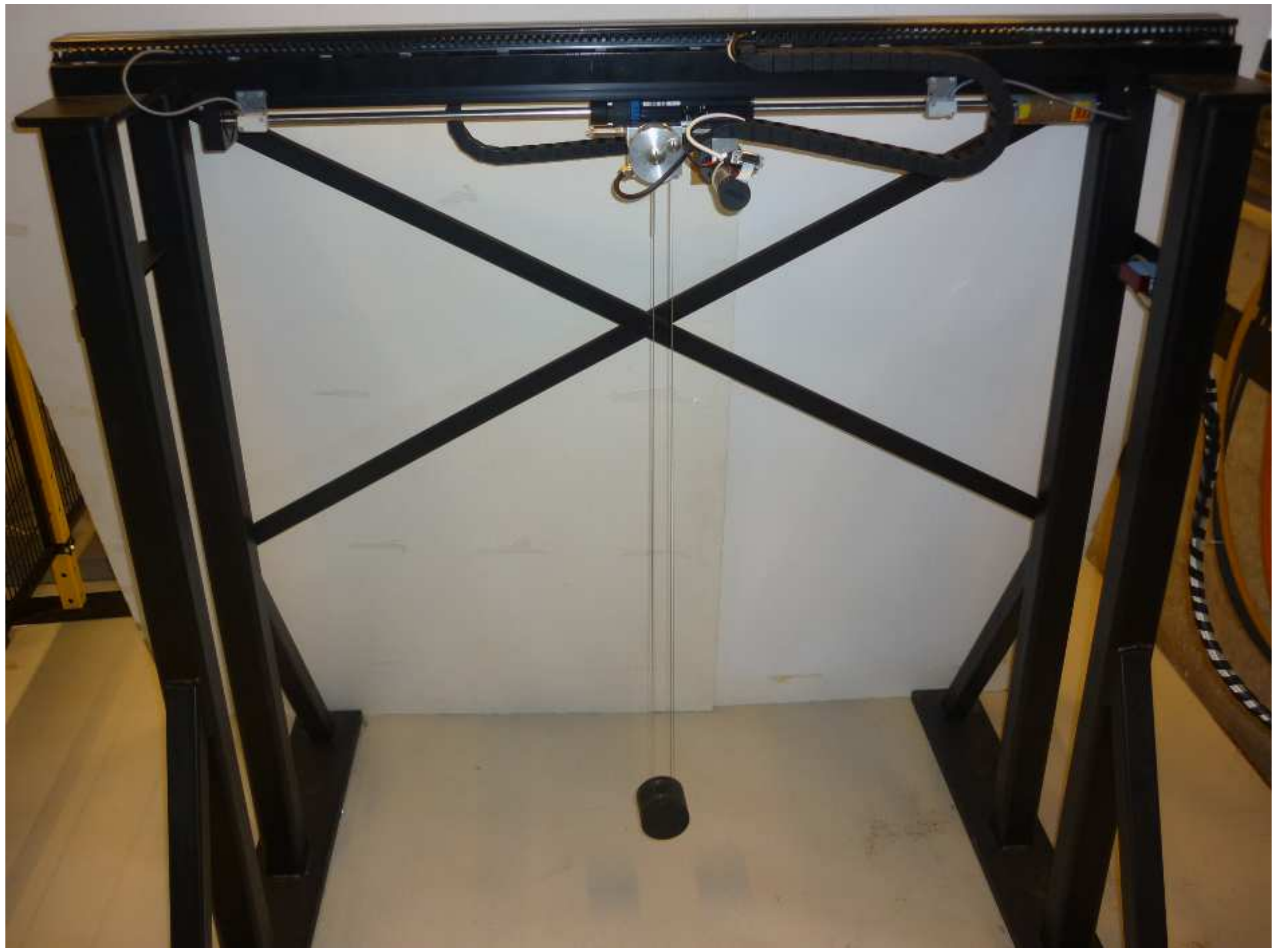
Real-world apps

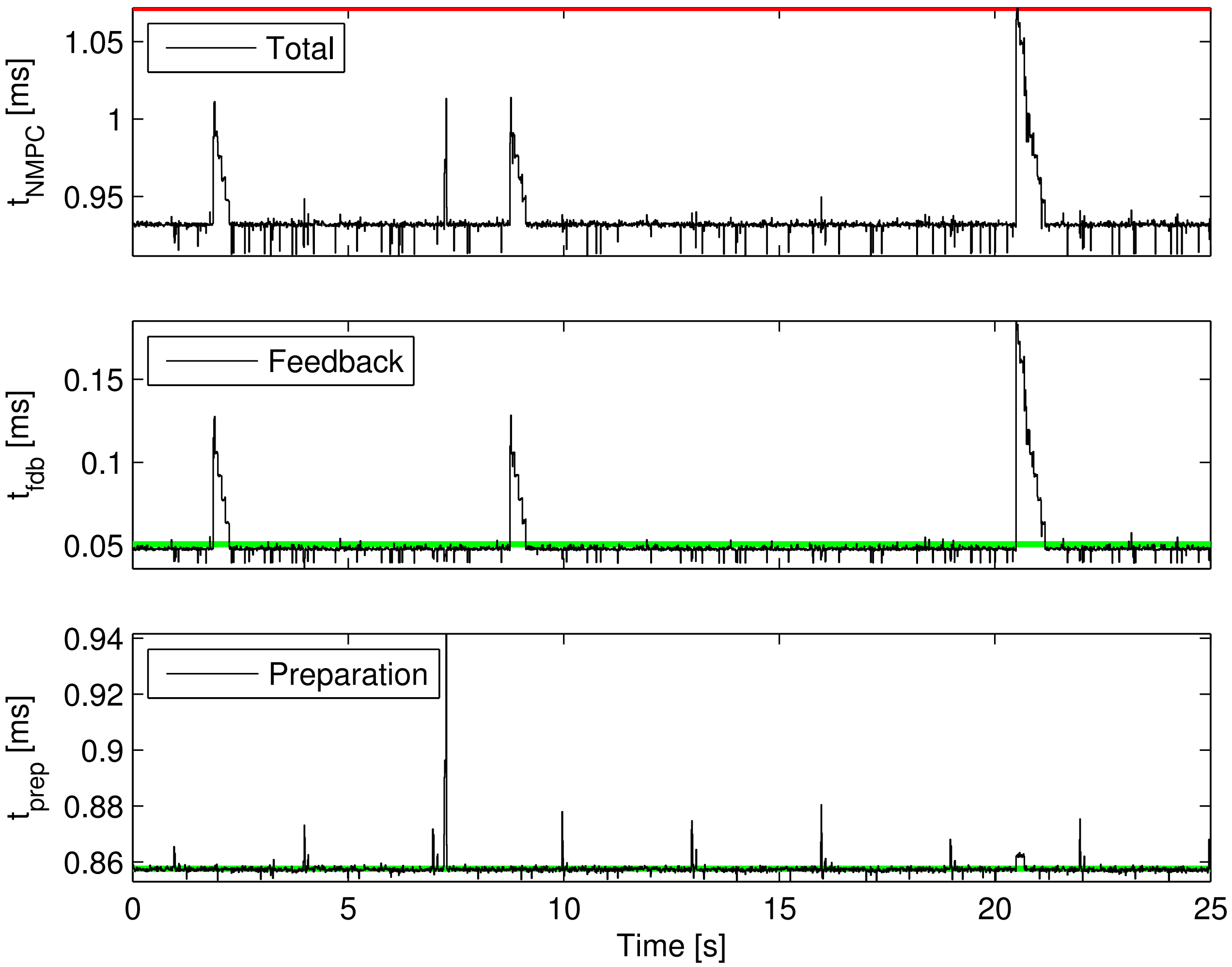
The Overhead Crane

8 states, 2 inputs, Ts = 10 ms

total max < 1.1 ms

feedback max < 0.2 ms





*Vukov2012

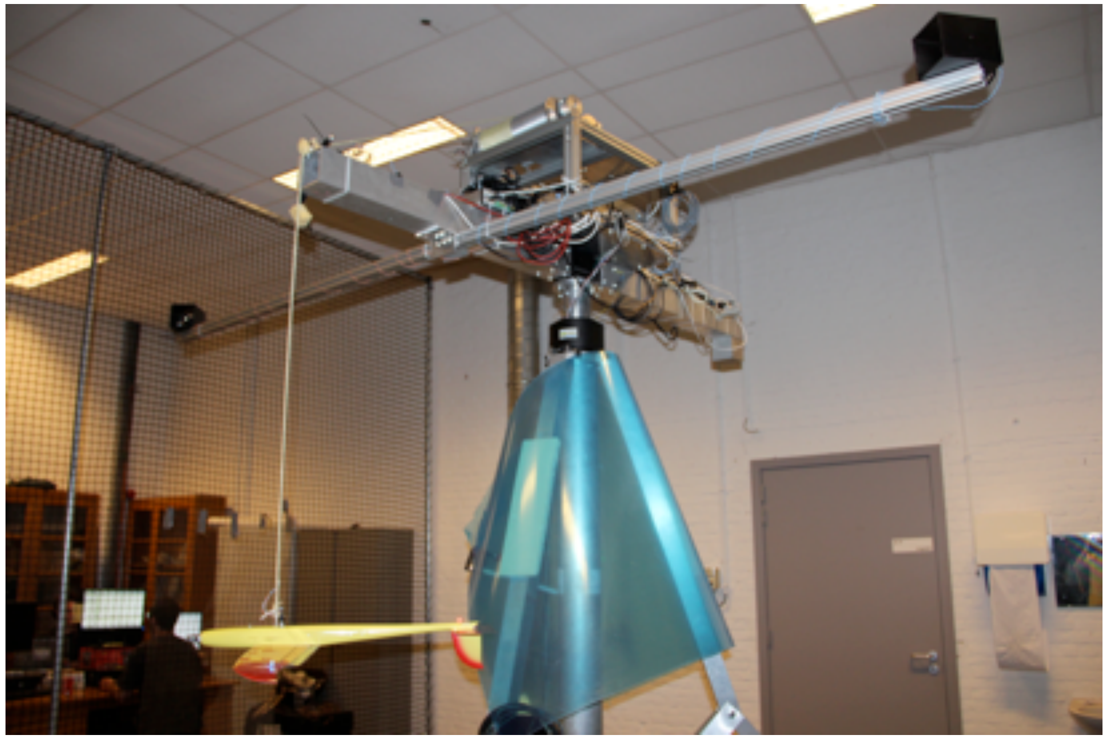

Our vision: replace tons of steel...

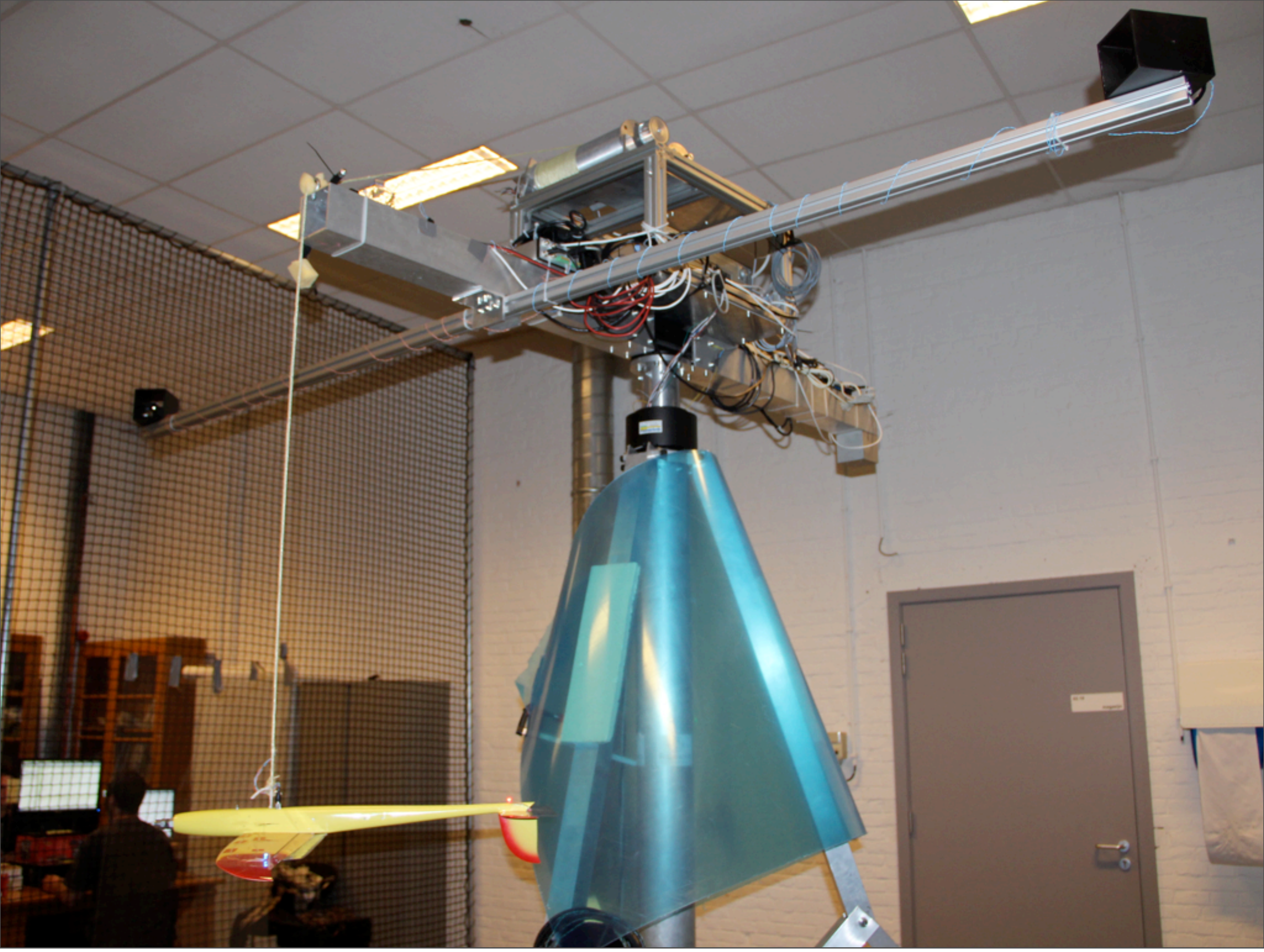

... with a tethered aircraft and smart control

Estimation & Control for kites

ERC HIGHWIND

An experimental test set-up for launch/recovery of an airborne wind energy (AWE) system, @ KU Leuven.





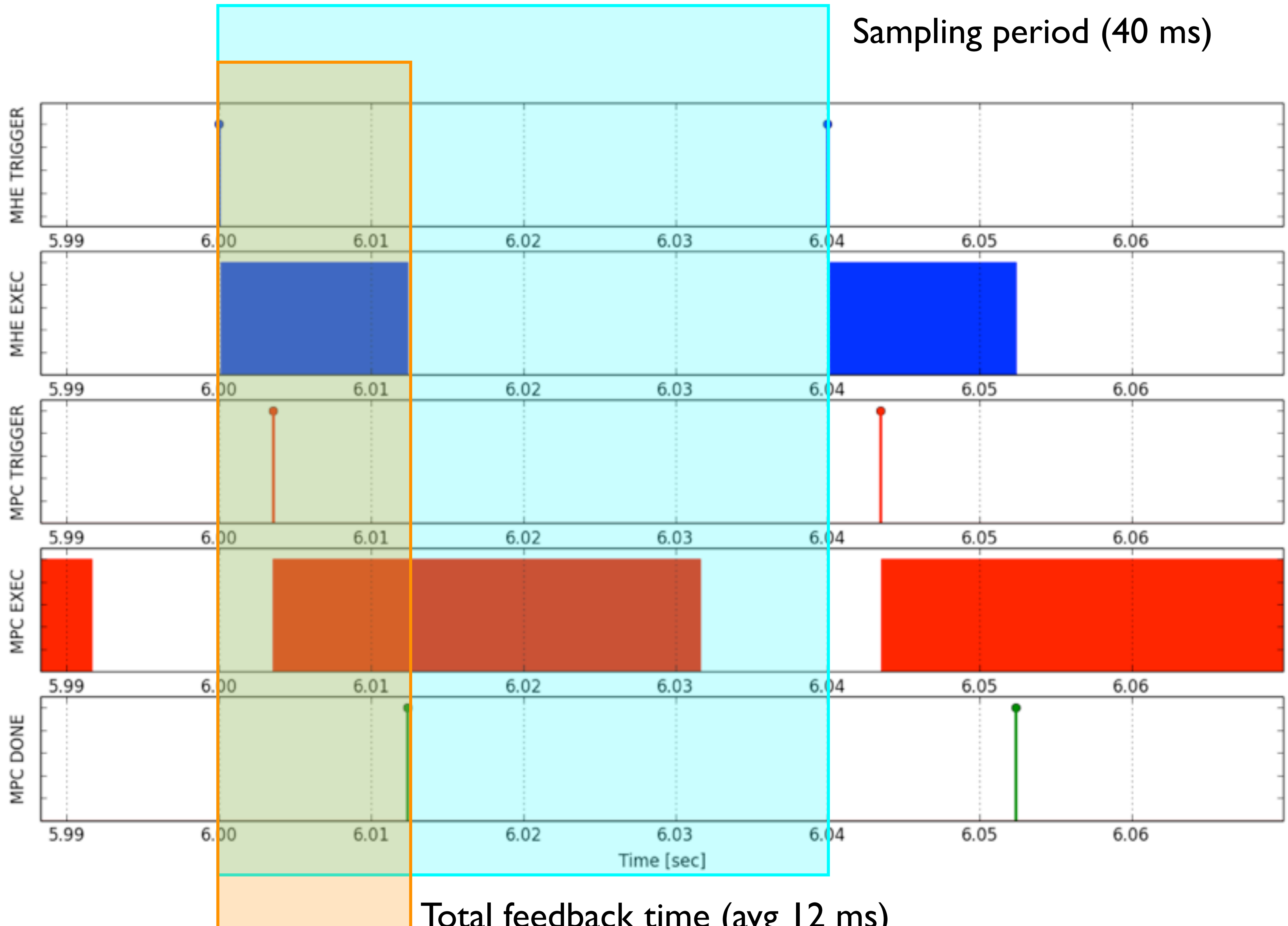
MHE & NMPC for kites*

- Nonlinear dynamics: a DAE with **27** differential states, **1** algebraic state and **4** controls*
- Nonlinear measurement (MHE) and reference (NMPC) functions
- Sensor fusion: IMU measurements @ 500 Hz, encoder measurements @ 1 kHz, line angle sensor @ 1 kHz. All averaged.
- MHE & NMPC update frequency: 25 Hz
- Max. feedback exec. time for MHE: **4 ms (N = 15)****
- Max. feedback exec. time for NMPC: **< 12 ms (N =50)**

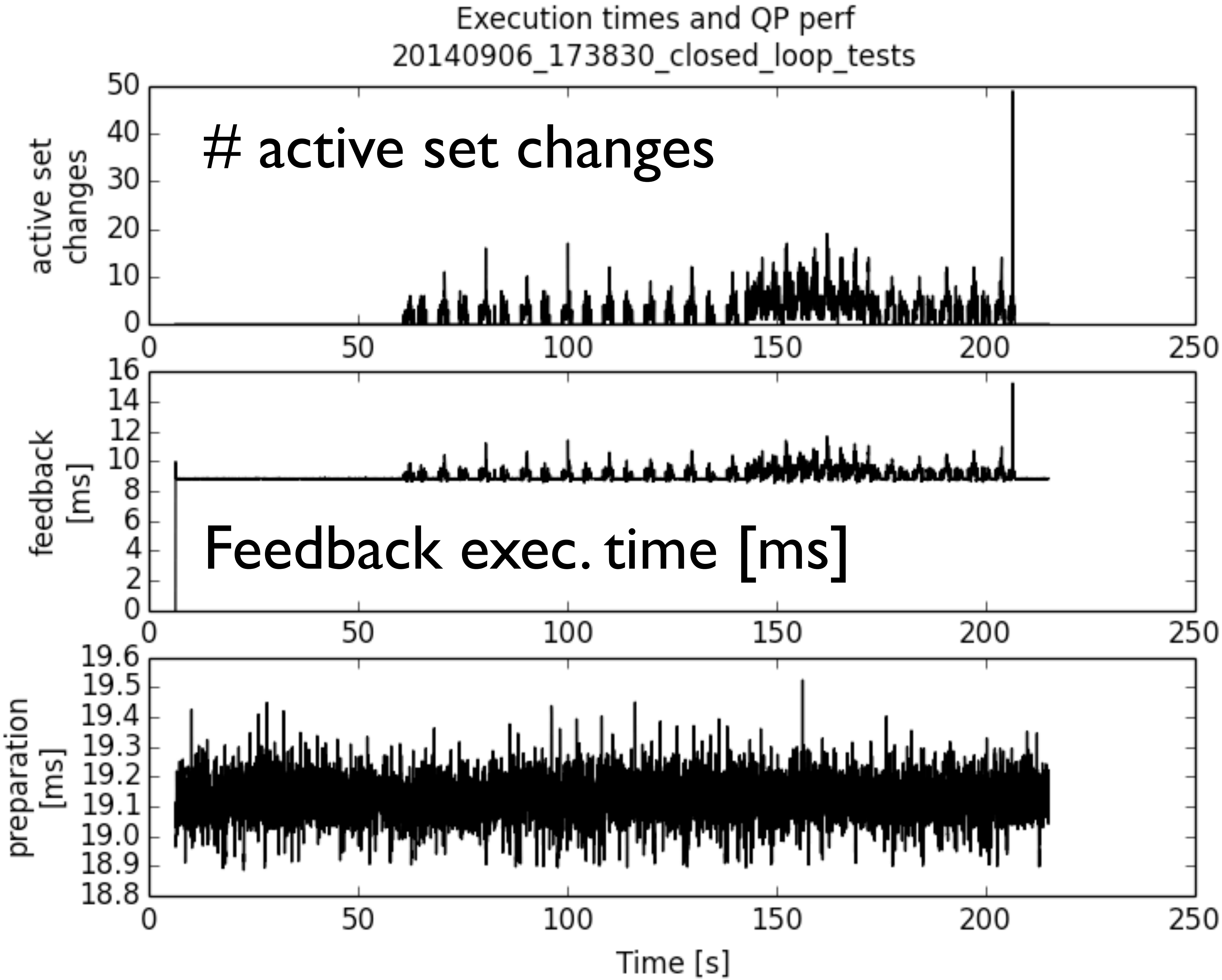
* Gros2012, Gros2013, Geebelen, Vukov et all 2013, Zanon2013

** MHE incorporates a simple disturbance model: 6 more states and 6 more controls

Timing info (experiments)



Timing info (experiments)



Other applications

- KUL: *Friction estimation for nano-positioning xy-tables*
- KUL, cooperation with CNH and New Holland:
  MHE and NMPC for agricultural machines
- KUL & Flanders' Mechatronics Technology Centre (FMTC):
  Control of mobile robots
- ETHZ & KUL: *MHE for induction motors*
- University of Linz, Austria:
  MHE and NMPC for diesel engine air system control
- ABB, Switzerland:
  Anti-surge control for centrifugal compressors & MPC for torque control in power el. applications
- U. Magdeburg: path following NMPC for Kuka LWR robot (Ts = 1Khz!)

Acknowledgements

EMBOCON

ERC HIGHWIND

Eurostars project SMART, Hans Joachim Ferreau, Boris Houska, Joel Andersson, Janick Frasch, Alex Domahidi, Gianluca Frison, Sebastien Gros, Mario Zanon

Demonstration of the ACADO toolkit

Thank you very much for your attention!

Questions?