# The QP Solvers in the ACADO Code Generation Tool

Milan Vukov    Moritz Diehl

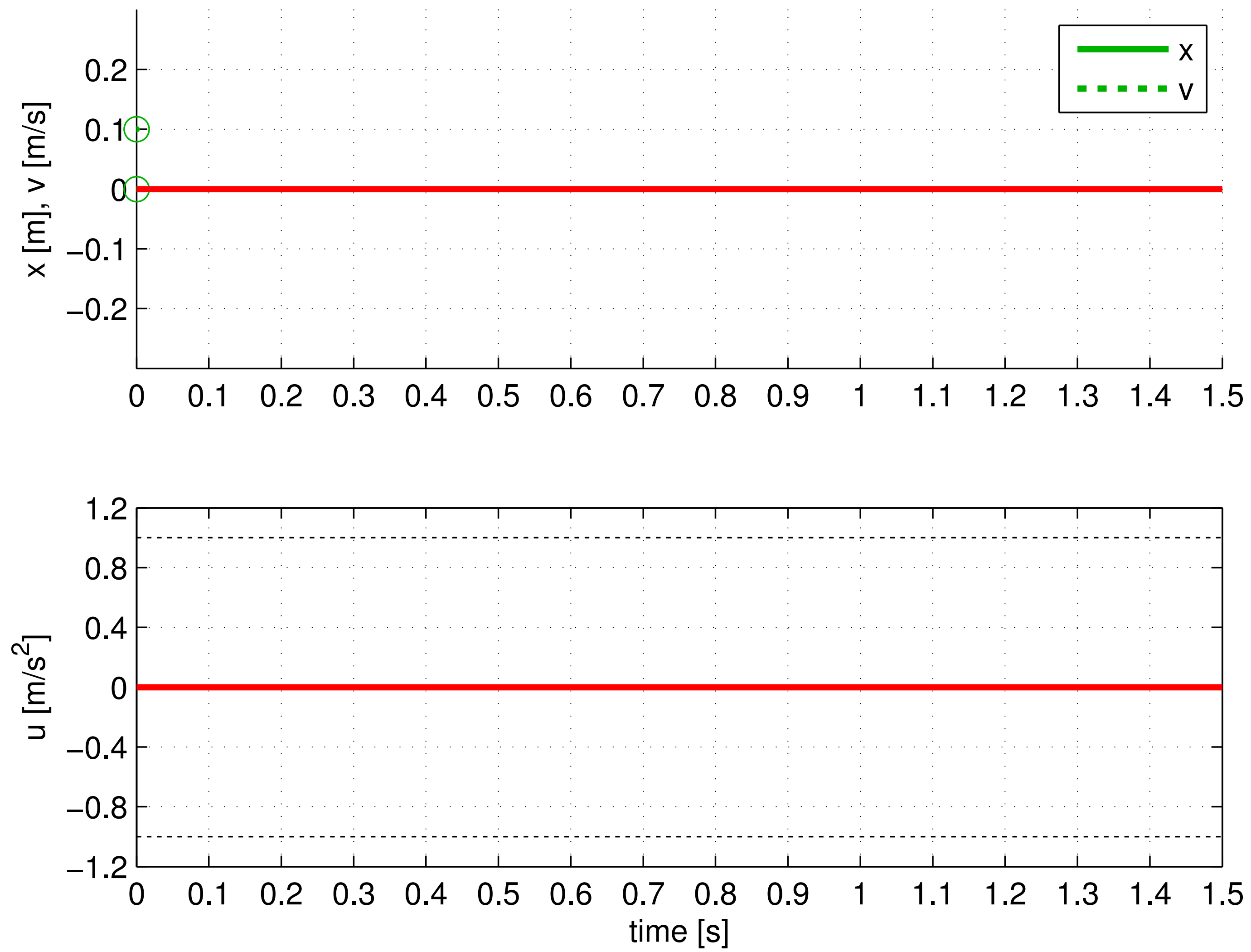
KU LEUVEN

UNI FREIBURG

# The Context

- Nonlinear Model Predictive Control (NMPC)
- Nonlinear Moving Horizon Estimation (NMHE)

# Nonlinear Model Predictive Control



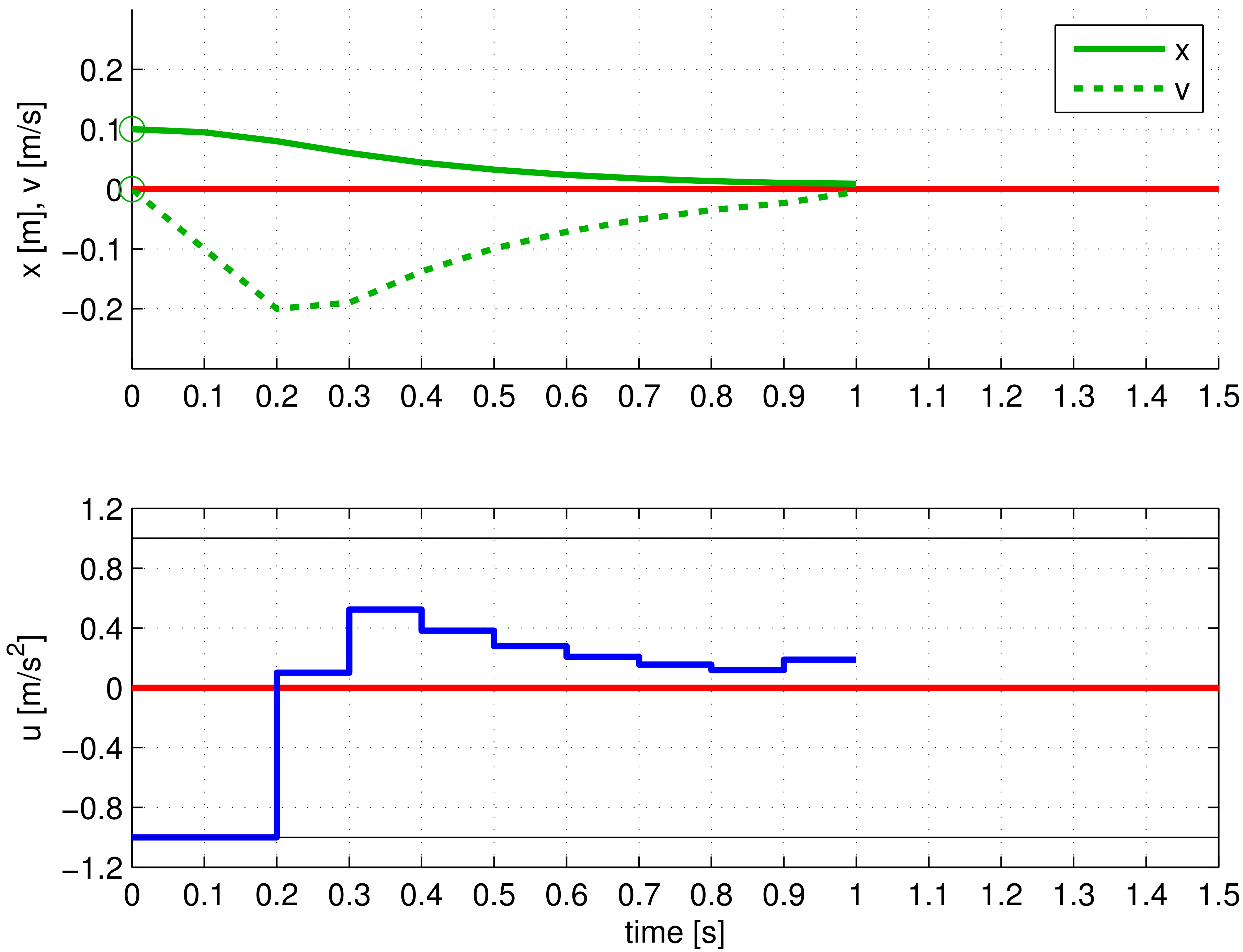

$$\min_{u,s} \quad \|s_P - s_{ref}\|^2_{Q_P} + \sum_{k=0}^{P-1} \|s_k - s_{ref}\|^2_Q + \|u_k - u_{ref}\|^2_R \quad \rightarrow$$ **deviation** from the reference

$$\text{s.t.} \quad s_{k+1} = f(s_k, u_k), \qquad k = 0, \ldots, P-1, \quad \rightarrow$$ **model** of the system evolution

$$h(s_k, u_k) \leq 0, \qquad k = 0, \ldots, P-1, \quad \rightarrow$$ **constraints**

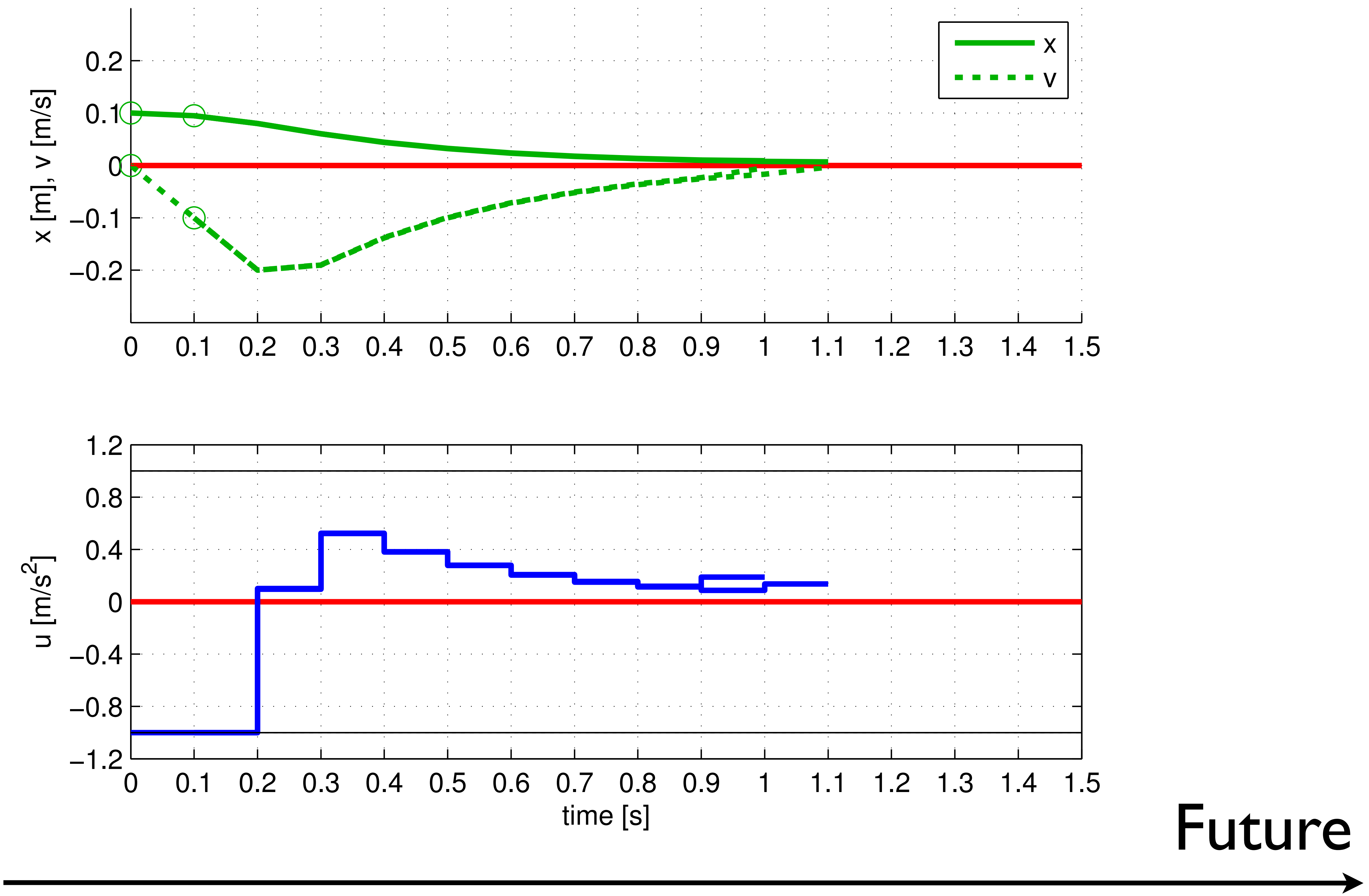
$$s_0 = \hat{x}_0 \quad \rightarrow$$ **current state** of the system

# Nonlinear Model Predictive Control

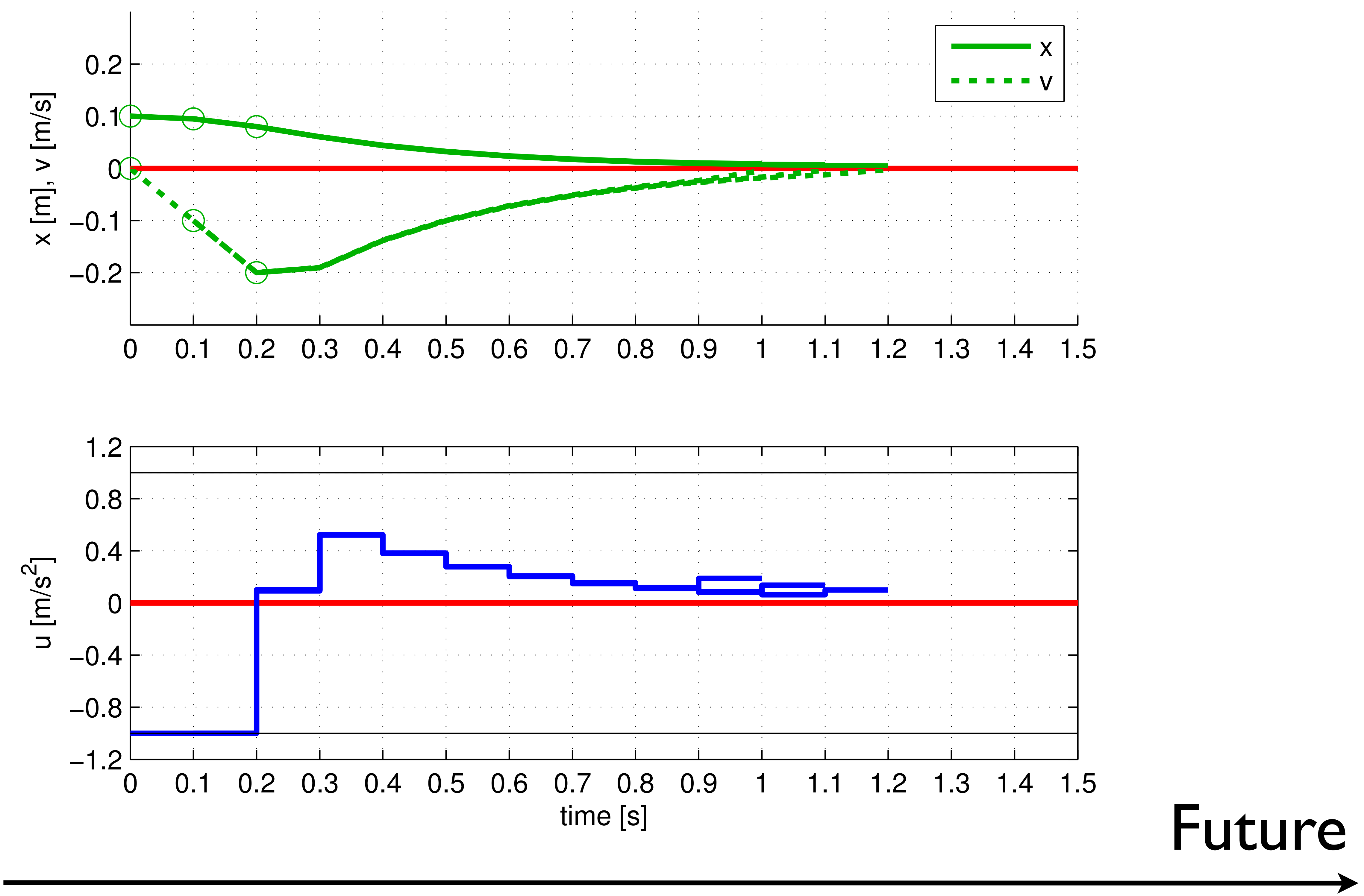



Future

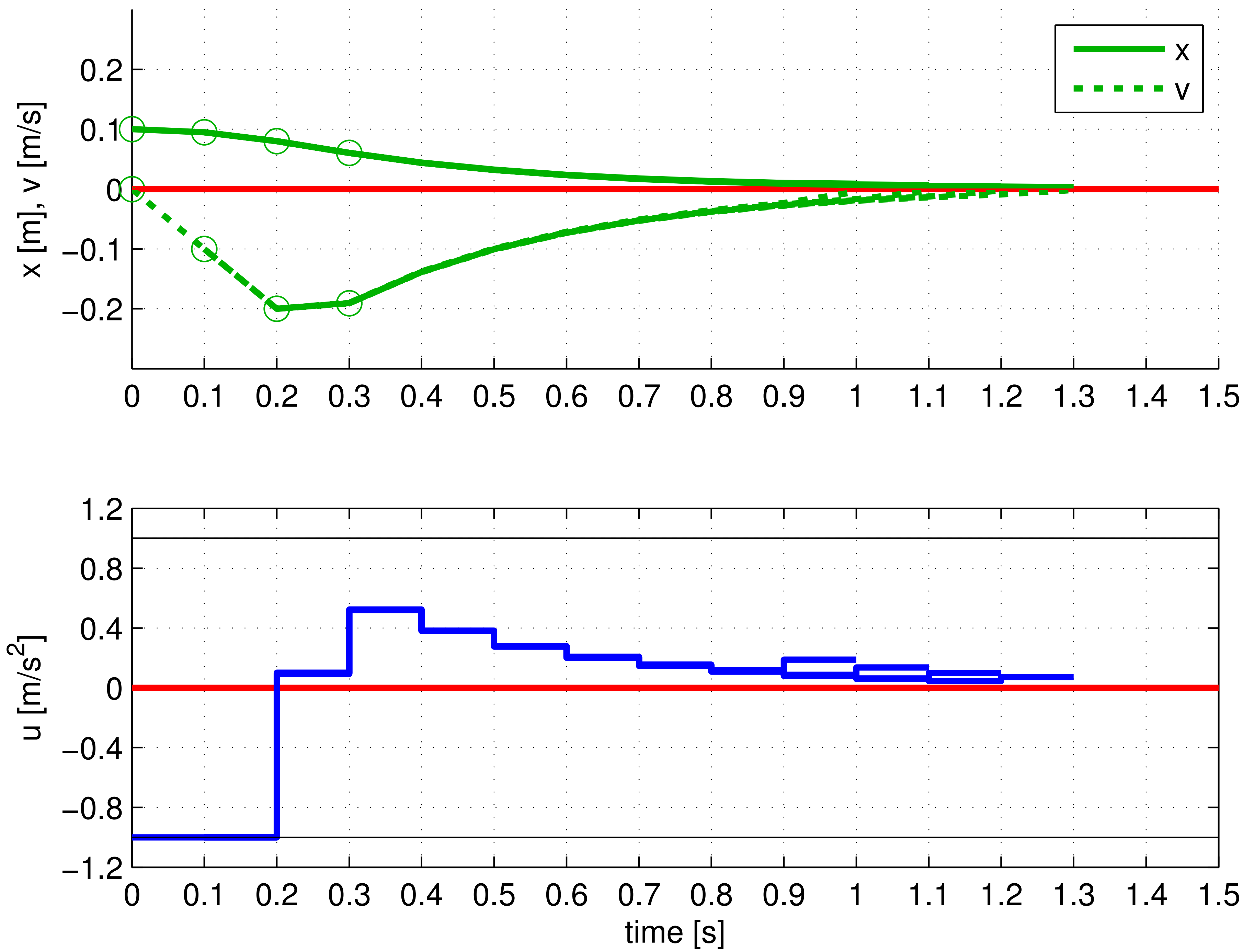
# Nonlinear Model Predictive Control



# Nonlinear Model Predictive Control



# Nonlinear Model Predictive Control




Future

# Optimal Control Problem

$$\min_{\substack{x_0,\dots,x_N \\ u_0,\dots,u_{N-1}}} \quad \sum_{k=0}^{N-1} ||h(x_k,u_k) - \tilde{y}_k||_{S_k}^2 + ||h_N(x_N) - \tilde{y}_N||_{S_N}^2$$

s.t.

$$x_0 = \hat{x}_0$$

$$x_{k+1} = F(x_k, u_k, z_k) \qquad \text{for } k = 0,\dots,N-1$$

$$x_k^{\text{lo}} \le x_k \le x_k^{\text{up}} \qquad \text{for } k = 0,\dots,N$$

$$u_k^{\text{lo}} \le u_k \le u_k^{\text{up}} \qquad \text{for } k = 0,\dots,N-1$$

$$r_k^{\text{lo}} \le r_k(x_k, u_k) \le r_k^{\text{up}} \qquad \text{for } k = 0,\dots,N-1$$

$$r_N^{\text{lo}} \le r_N(x_n) \le r_N^{\text{up}}$$

# Optimal Control Problem

$$\min_{\substack{x_0,\ldots,x_N \\ u_0,\ldots,u_{N-1}}} \quad \sum_{k=0}^{N-1} ||h(x_k,u_k) - \tilde{y}_k||_{S_k}^2 + ||h_N(x_N) - \tilde{y}_N||_{S_N}^2$$

s.t.

$$x_0 = \hat{x}_0$$

$$x_{k+1} = F(x_k, u_k, z_k) \quad \text{for } k = 0,\ldots,N-1$$

$$x_k^{\text{lo}} \le x_k \le x_k^{\text{up}} \quad \text{for } k = 0,\ldots,N$$

$$u_k^{\text{lo}} \le u_k \le u_k^{\text{up}} \quad \text{for } k = 0,\ldots,N-1$$

$$r_k^{\text{lo}} \le r_k(x_k,u_k) \le r_k^{\text{up}} \quad \text{for } k = 0,\ldots,N-1$$

$$r_N^{\text{lo}} \le r_N(x_n) \le r_N^{\text{up}}$$

# Optimal Control Problem

$$\min_{\substack{x_0,\ldots,x_N \\ u_0,\ldots,u_{N-1}}} \quad \sum_{k=0}^{N-1} ||h(x_k,u_k) - \tilde{y}_k||_{S_k}^2 + ||h_N(x_N) - \tilde{y}_N||_{S_N}^2$$

$$\text{s.t.} \quad \color{red}{x_0 = \hat{x}_0}$$

$$x_{k+1} = F(x_k, u_k, z_k) \qquad \text{for } k = 0,\ldots,N-1$$

$$x_k^{\text{lo}} \le x_k \le x_k^{\text{up}} \qquad \text{for } k = 0,\ldots,N$$

$$u_k^{\text{lo}} \le u_k \le u_k^{\text{up}} \qquad \text{for } k = 0,\ldots,N-1$$

$$r_k^{\text{lo}} \le r_k(x_k, u_k) \le r_k^{\text{up}} \qquad \text{for } k = 0,\ldots,N-1$$

$$r_N^{\text{lo}} \le r_N(x_n) \le r_N^{\text{up}}$$

# Optimal Control Problem

$$\min_{\substack{x_0,\ldots,x_N \\ u_0,\ldots,u_{N-1}}} \sum_{k=0}^{N-1} ||h(x_k,u_k) - \tilde{y}_k||_{S_k}^2 + ||h_N(x_N) - \tilde{y}_N||_{S_N}^2$$

s.t.

$$x_0 = \hat{x}_0$$

$$x_{k+1} = F(x_k, u_k, z_k) \quad \text{for } k = 0,\ldots,N-1$$

$$x_k^{\text{lo}} \leq x_k \leq x_k^{\text{up}} \quad \text{for } k = 0,\ldots,N$$

$$u_k^{\text{lo}} \leq u_k \leq u_k^{\text{up}} \quad \text{for } k = 0,\ldots,N-1$$

$$r_k^{\text{lo}} \leq r_k(x_k,u_k) \leq r_k^{\text{up}} \quad \text{for } k = 0,\ldots,N-1$$

$$r_N^{\text{lo}} \leq r_N(x_n) \leq r_N^{\text{up}}$$

# Optimal Control Problem

$$\min_{\substack{x_0,\ldots,x_N \\ u_0,\ldots,u_{N-1}}} \quad \sum_{k=0}^{N-1} ||h(x_k,u_k) - \tilde{y}_k||_{S_k}^2 + ||h_N(x_N) - \tilde{y}_N||_{S_N}^2$$

s.t.

$$x_0 = \hat{x}_0$$

$$x_{k+1} = F(x_k, u_k, z_k) \qquad \text{for } k = 0, \ldots, N-1$$

$$x_k^{\text{lo}} \le x_k \le x_k^{\text{up}} \qquad \text{for } k = 0, \ldots, N$$

$$u_k^{\text{lo}} \le u_k \le u_k^{\text{up}} \qquad \text{for } k = 0, \ldots, N-1$$

$$r_k^{\text{lo}} \le r_k(x_k, u_k) \le r_k^{\text{up}} \qquad \text{for } k = 0, \ldots, N-1$$

$$r_N^{\text{lo}} \le r_N(x_n) \le r_N^{\text{up}}$$

# Optimal Control Problem

$$\min_{\substack{x_0,\ldots,x_N\\u_0,\ldots,u_{N-1}}} \quad \sum_{k=0}^{N-1} ||h(x_k,u_k) - \tilde{y}_k||_{S_k}^2 + ||h_N(x_N) - \tilde{y}_N||_{S_N}^2$$

s.t.

$$x_0 = \hat{x}_0$$

$$x_{k+1} = F(x_k, u_k, z_k) \quad \text{for } k = 0,\ldots,N-1$$

$$x_k^{\text{lo}} \le x_k \le x_k^{\text{up}} \quad \text{for } k = 0,\ldots,N$$

$$u_k^{\text{lo}} \le u_k \le u_k^{\text{up}} \quad \text{for } k = 0,\ldots,N-1$$

$$r_k^{\text{lo}} \le r_k(x_k,u_k) \le r_k^{\text{up}} \quad \text{for } k = 0,\ldots,N-1$$

$$r_N^{\text{lo}} \le r_N(x_n) \le r_N^{\text{up}}$$

# Solution methods

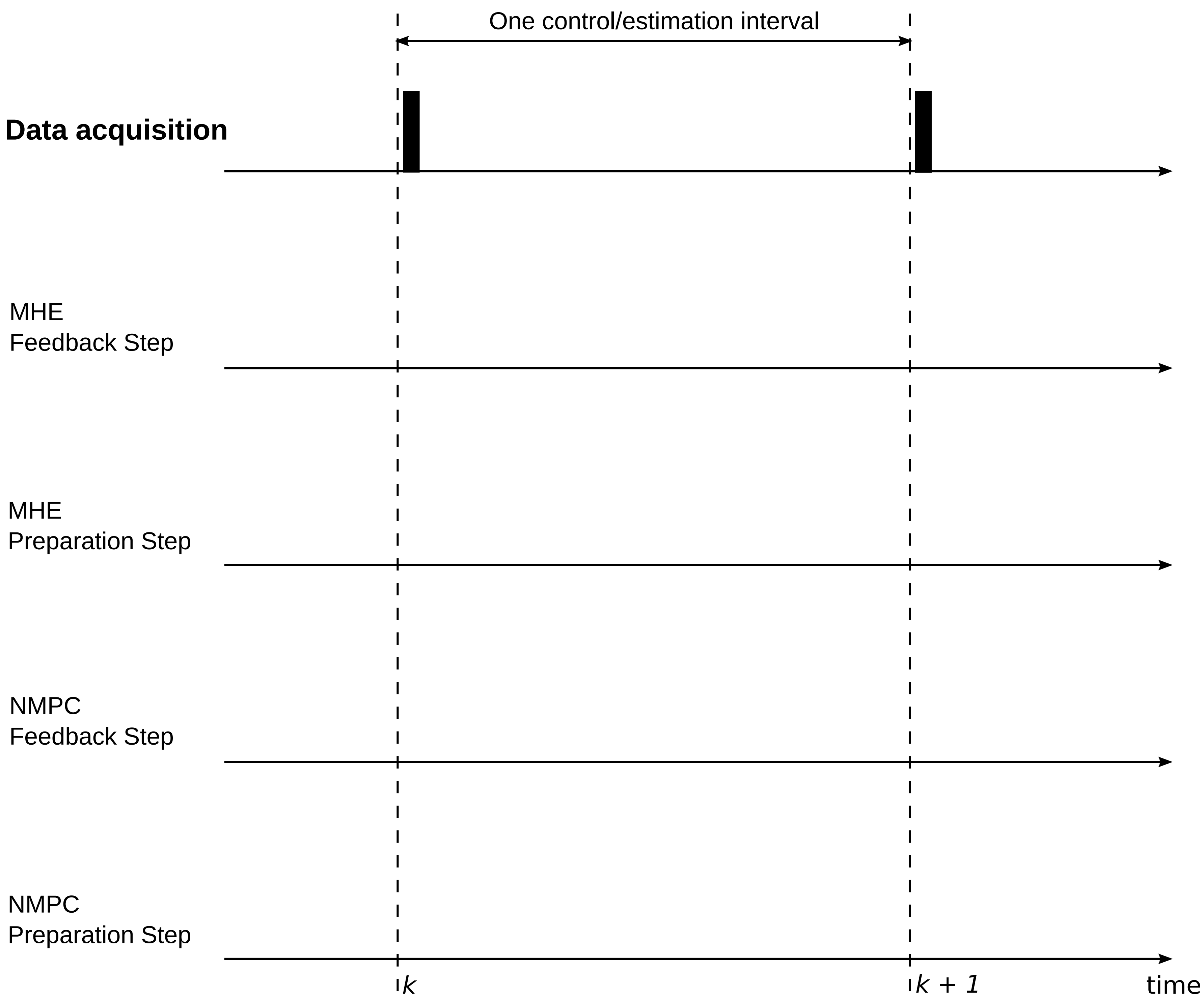
## Real-time Iterations [Diehl 2002]

- Problem discretization - single/multiple shooting [Bock 1984]
- 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

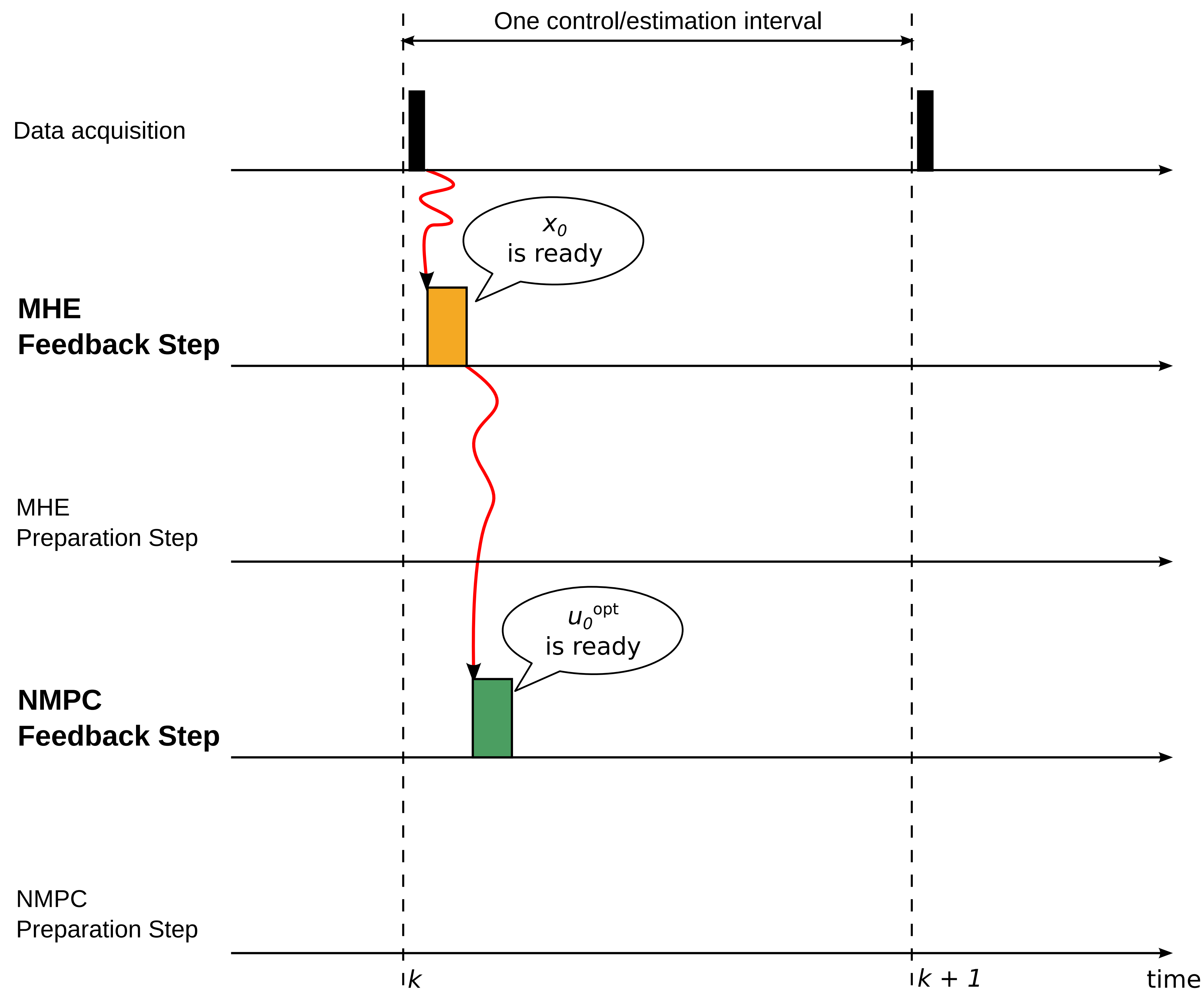
$$\min_{\substack{x_0,\ldots,x_N \\ u_0,\ldots,u_{N-1}}} \quad ||x_0 - x_{\mathrm{AC}}||^2_{S_{\mathrm{AC}}} + \sum_{k=0}^{N-1} ||h(x_k,u_k) - \tilde{y}_k||^2_{S_k} + ||h_N(x_N) - \tilde{y}_N||^2_{S_N}$$

$$\text{s.t.} \quad \begin{aligned}
& x_0 = \hat{x}_0 \\
& x_{k+1} = F(x_k,u_k,z_k) && \text{for } k = 0,\ldots,N-1 \\
& x_k^{\mathrm{lo}} \le x_k \le x_k^{\mathrm{up}} && \text{for } k = 0,\ldots,N \\
& u_k^{\mathrm{lo}} \le u_k \le u_k^{\mathrm{up}} && \text{for } k = 0,\ldots,N-1 \\
& r_k^{\mathrm{lo}} \le r_k(x_k,u_k) \le r_k^{\mathrm{up}} && \text{for } k = 0,\ldots,N-1 \\
& r_N^{\mathrm{lo}} \le r_N(x_n) \le r_N^{\mathrm{up}}
\end{aligned}$$

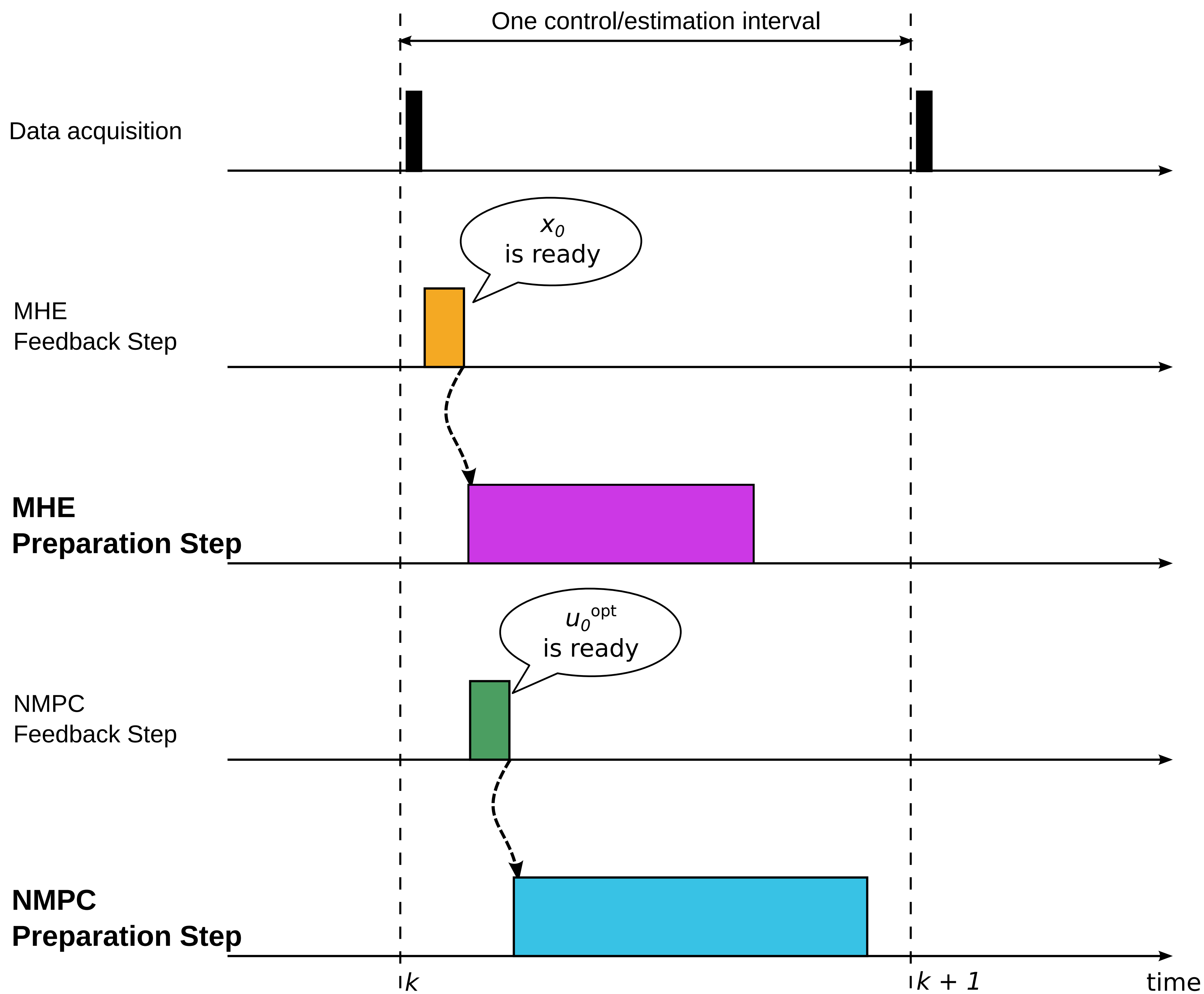
# RTI Scheme 101(I)



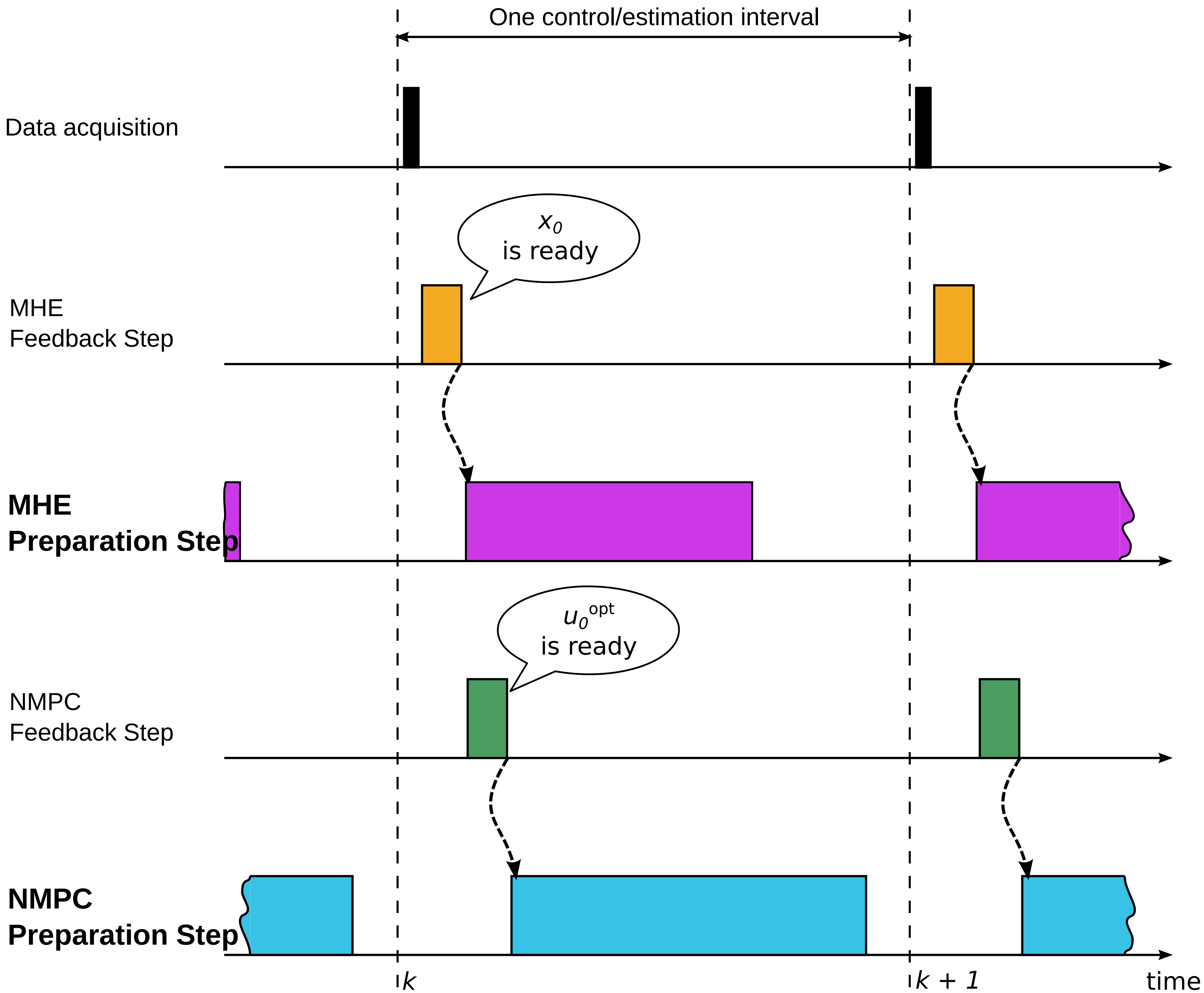
# RTI Scheme 101(2)



# RTI Scheme 101(3)



# RTI Scheme 101(5)

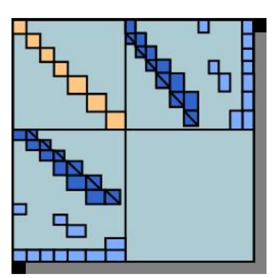


# Classical Condensing

$$\underset{x_0,u_0,\ldots,x_N}{\text{minimize}} \quad \frac{1}{2}\sum_{k=0}^{N-1}\begin{bmatrix}x_k\\u_k\end{bmatrix}^T\begin{bmatrix}Q_k & S_k\\S_k^T & R_k\end{bmatrix}\begin{bmatrix}x_k\\u_k\end{bmatrix}+\begin{bmatrix}x_k\\u_k\end{bmatrix}^T\begin{bmatrix}g_k^{\mathrm{x}}\\g_k^{\mathrm{u}}\end{bmatrix}^T$$

$$+\tfrac{1}{2}x_N^T Q_{\mathrm{e}} x_N + x_N^T g_{\mathrm{e}}^{\mathrm{x}}$$

$$\text{subject to} \quad \begin{aligned}
&x_{k+1} = A_k x_k + B_k u_k + c_k, && \text{for} \quad k = 0,\ldots,N-1\\
&x_k^{\mathrm{lo}} \le x_k \le x_k^{\mathrm{up}}, && \text{for} \quad k = 0,\ldots,N,\\
&u_k^{\mathrm{lo}} \le u_k \le u_k^{\mathrm{up}}, && \text{for} \quad k = 0,\ldots,N-1,\\
&b_k^{\mathrm{lo}} \le C_k x_k + D_k u_k \le b_k^{\mathrm{up}}, && \text{for} \quad k = 0,\ldots,N-1,\\
&b_{\mathrm{e}}^{\mathrm{lo}} \le C_{\mathrm{e}} x_N \le b_{\mathrm{e}}^{\mathrm{up}},
\end{aligned}$$



$$\underset{u}{\text{minimize}} \quad \frac{1}{2}u^T H_{\mathrm{C}} u + u^T g_{\mathrm{C}}$$

$$\text{subject to} \quad \begin{aligned} &u^{\mathrm{lo}} \le u \le u^{\mathrm{up}}\\ &b_{\mathrm{C}}^{\mathrm{lo}} \le A_{\mathrm{C}} u \le b_{\mathrm{C}}^{\mathrm{up}}\end{aligned}$$

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

# Classical Condensing

states → $\Delta w_2 = d + C\Delta s_0 + E\Delta w_1$ ← controls

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

# Classical Condensing

states → $\Delta w_2 = d + C\Delta s_0 + E\Delta w_1$ ← controls

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

$O(N^3)$ → $E'\overline{Q}E$

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

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

# Exploit the structure! [Leineweber1999]

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

Then $E'\overline{Q}E$ can be computed more efficiently

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

# Can we do better?

# ... From another point of view ...

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

# ... From another point of view ...

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

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

# ... From another point of view ...

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

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

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

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

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

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

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

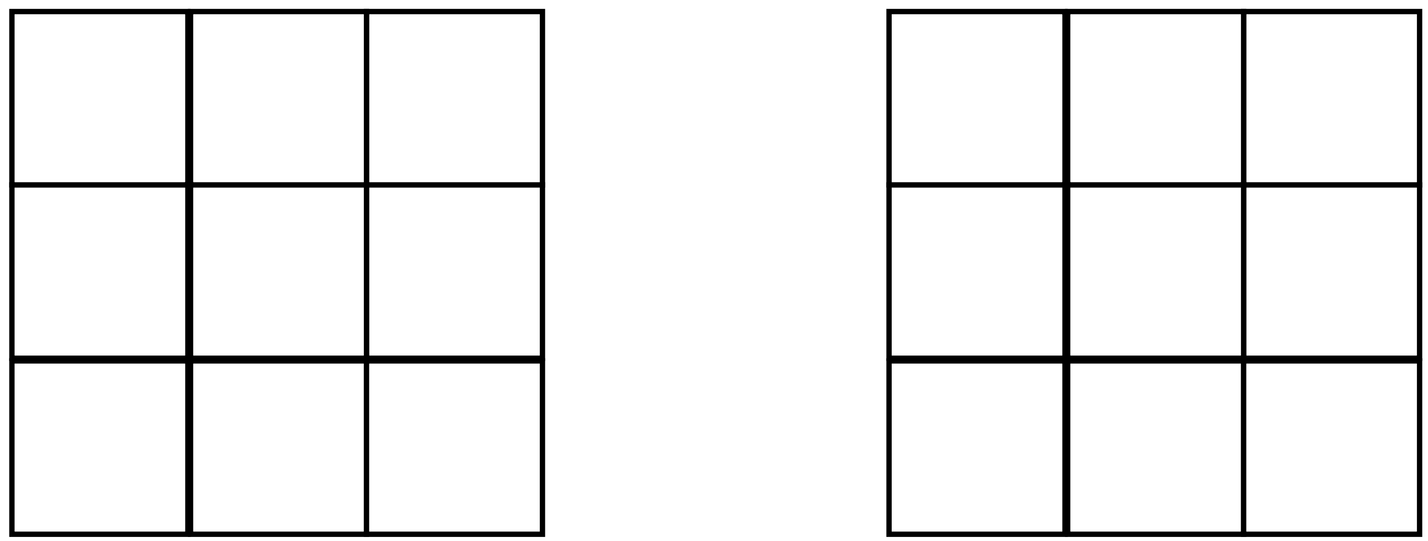
$$\underbrace{\begin{bmatrix} I & -A_1^{\mathrm{T}} & & \\ & I & \ddots & \\ & & \ddots & -A_{N-1}^{\mathrm{T}} \\ & & & I \end{bmatrix}}_{A^{\mathrm{T}}} T = \underbrace{\begin{bmatrix} E_{0,0}^Q & & & \\ E_{1,0}^Q & E_{1,1}^Q & & \\ \vdots & \vdots & \ddots & \\ E_{N-1,0}^Q & E_{N-1,1}^Q & \cdots &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

**$N^2$ complexity**

* Frison2012, Andersson2013, Frison2013

# Building of the Condensed Hessian

***E*** ***H***

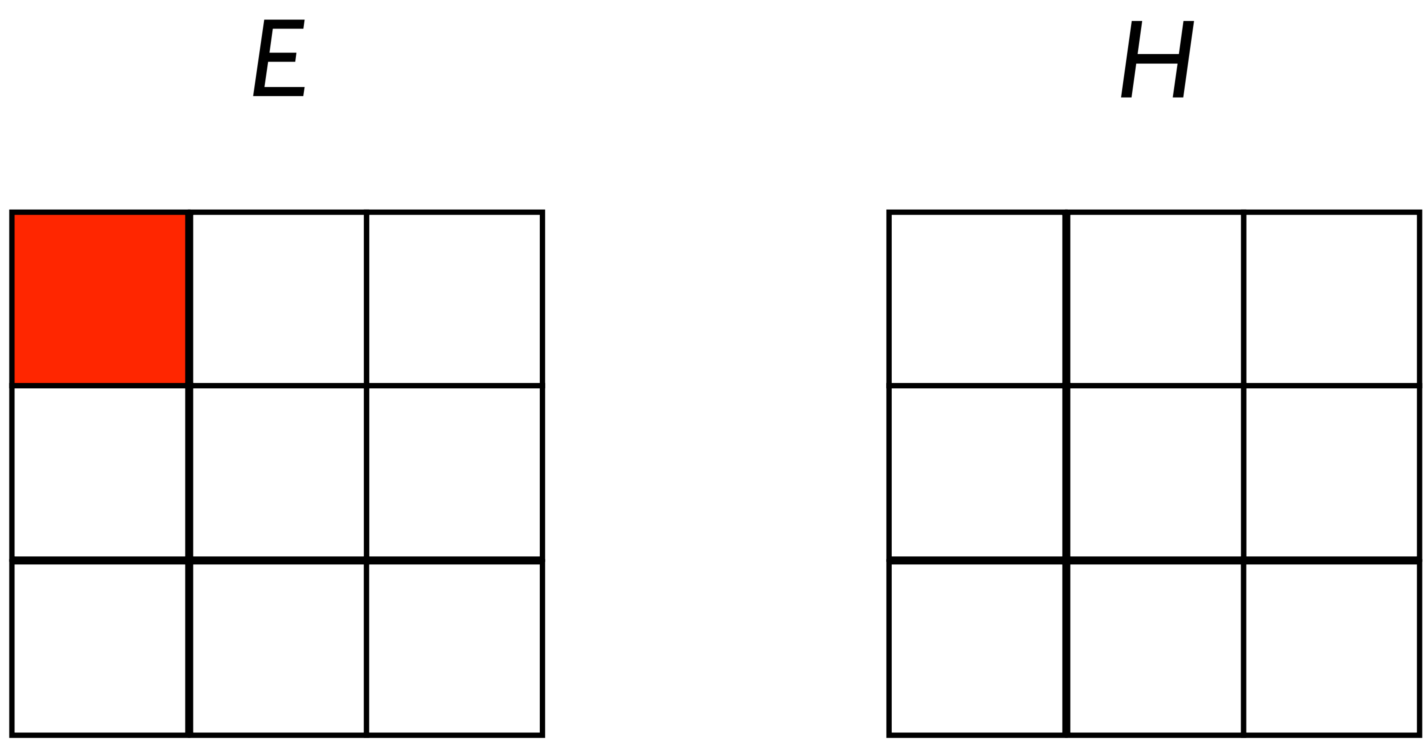


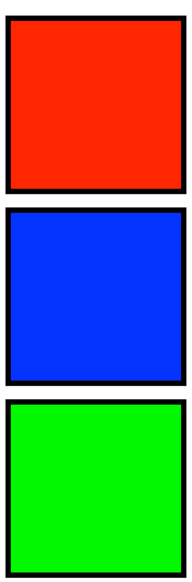
($N$ = 3)

**E**: sensitivity propagation matrix

**H**: condensed Hessian matrix

# Building of the Condensed Hessian



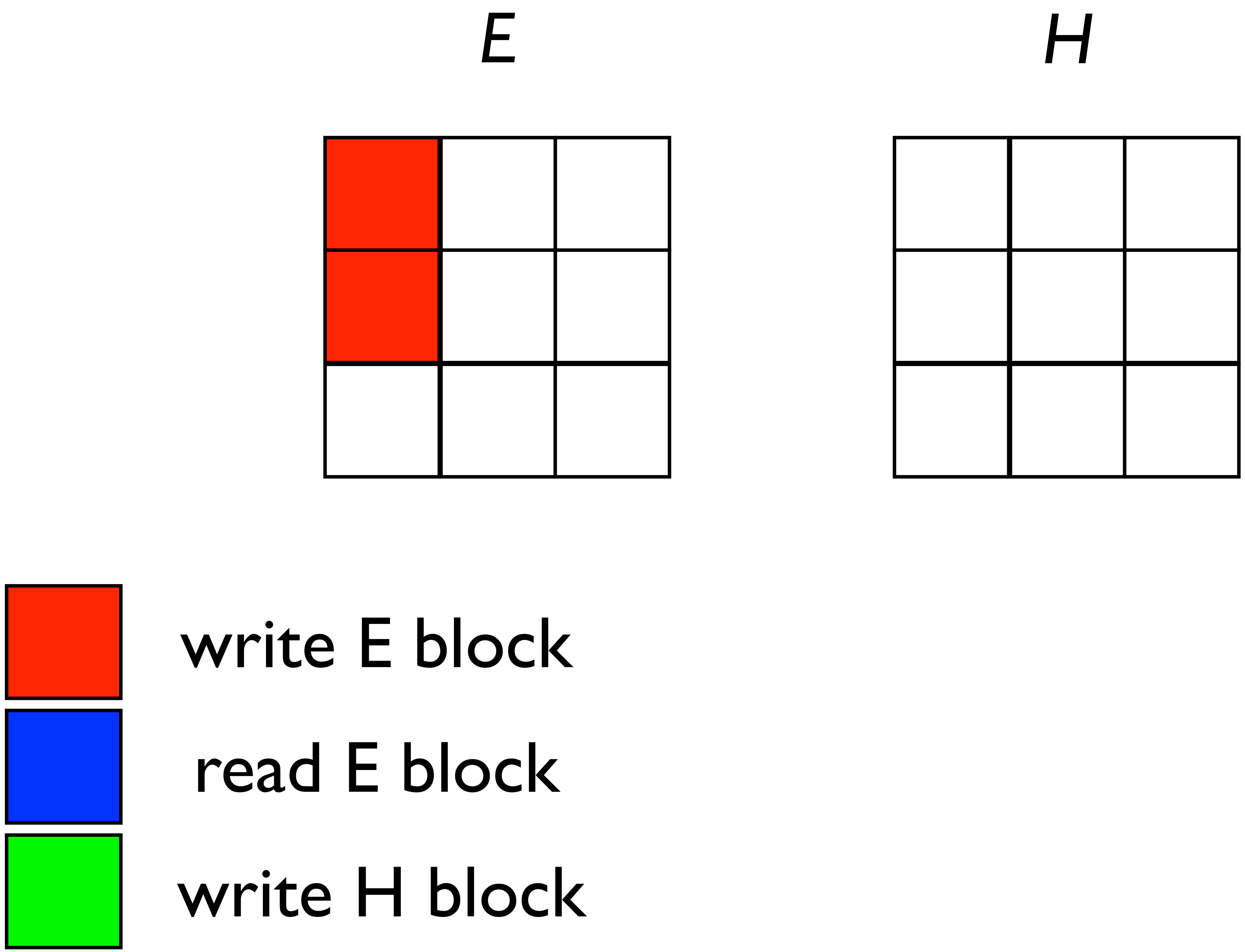



write E block

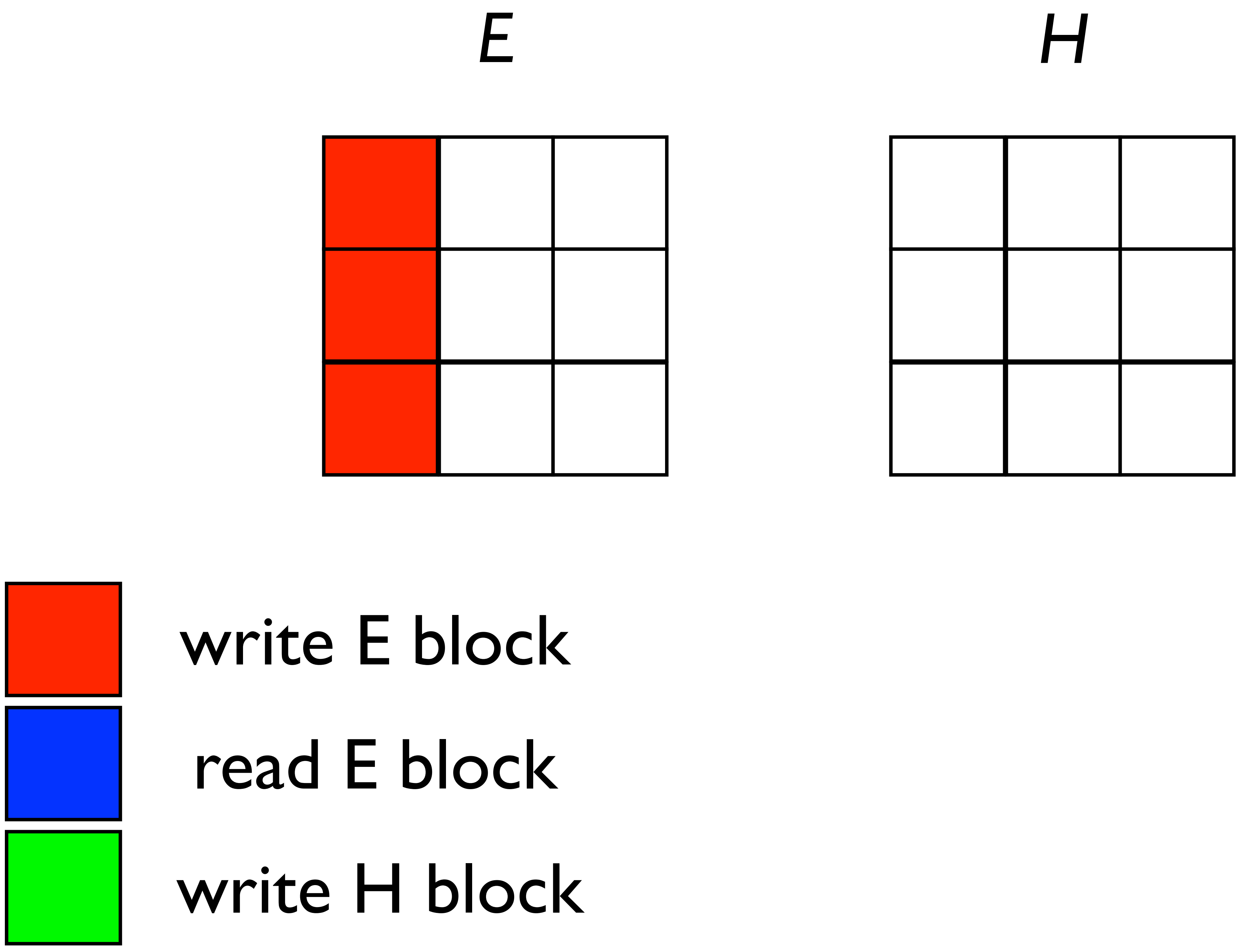
read E block

write H block

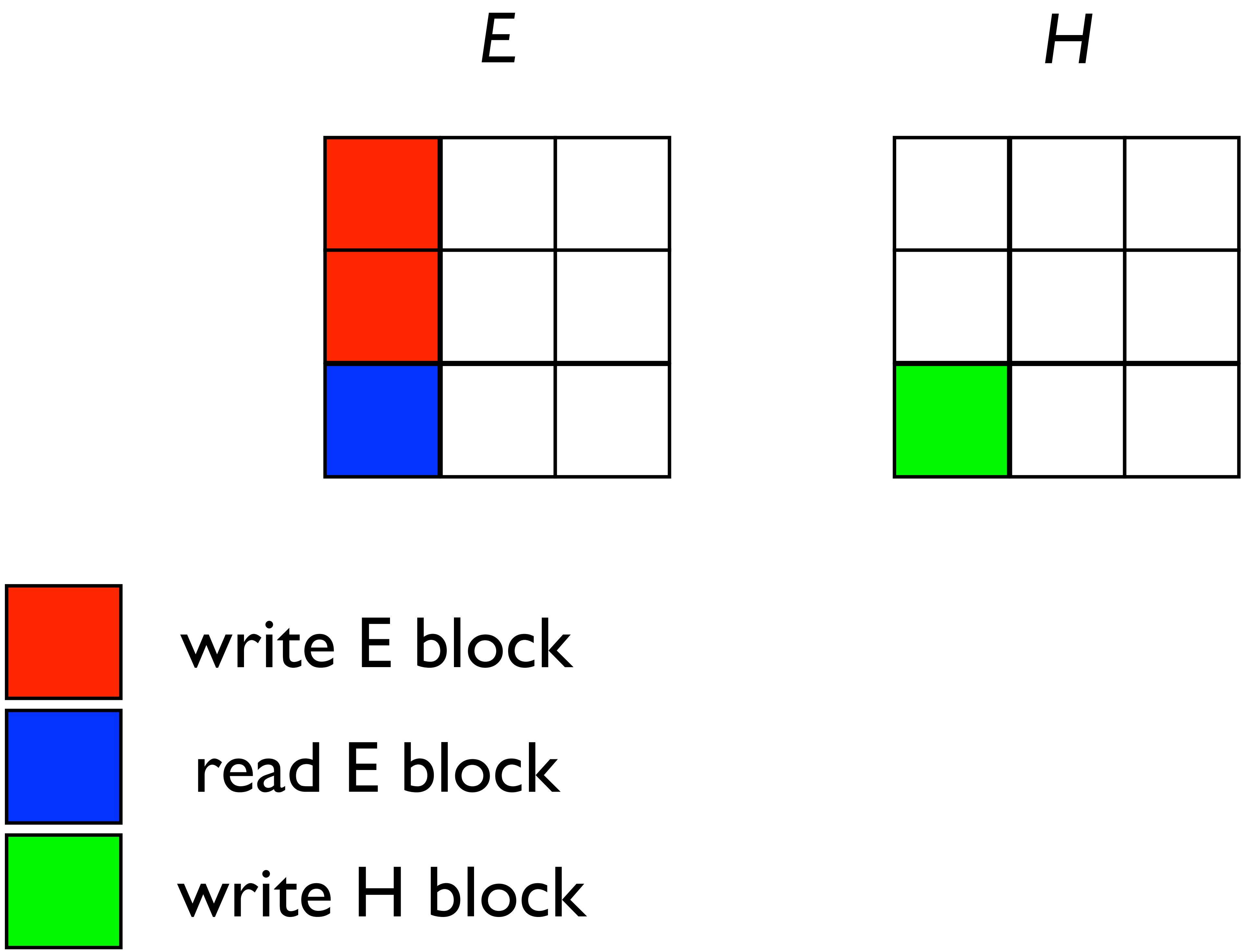
# Building of the Condensed Hessian



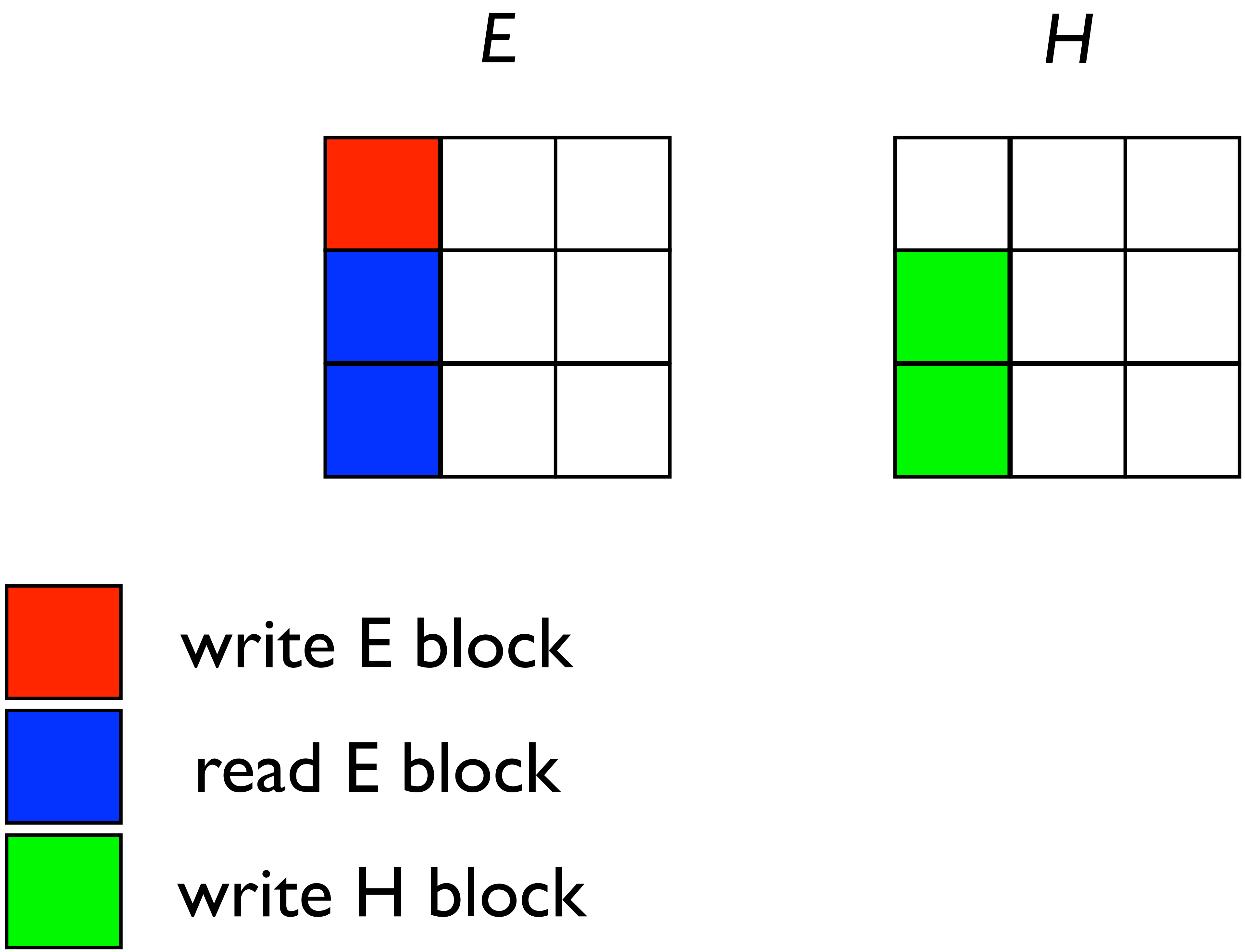
# Building of the Condensed Hessian



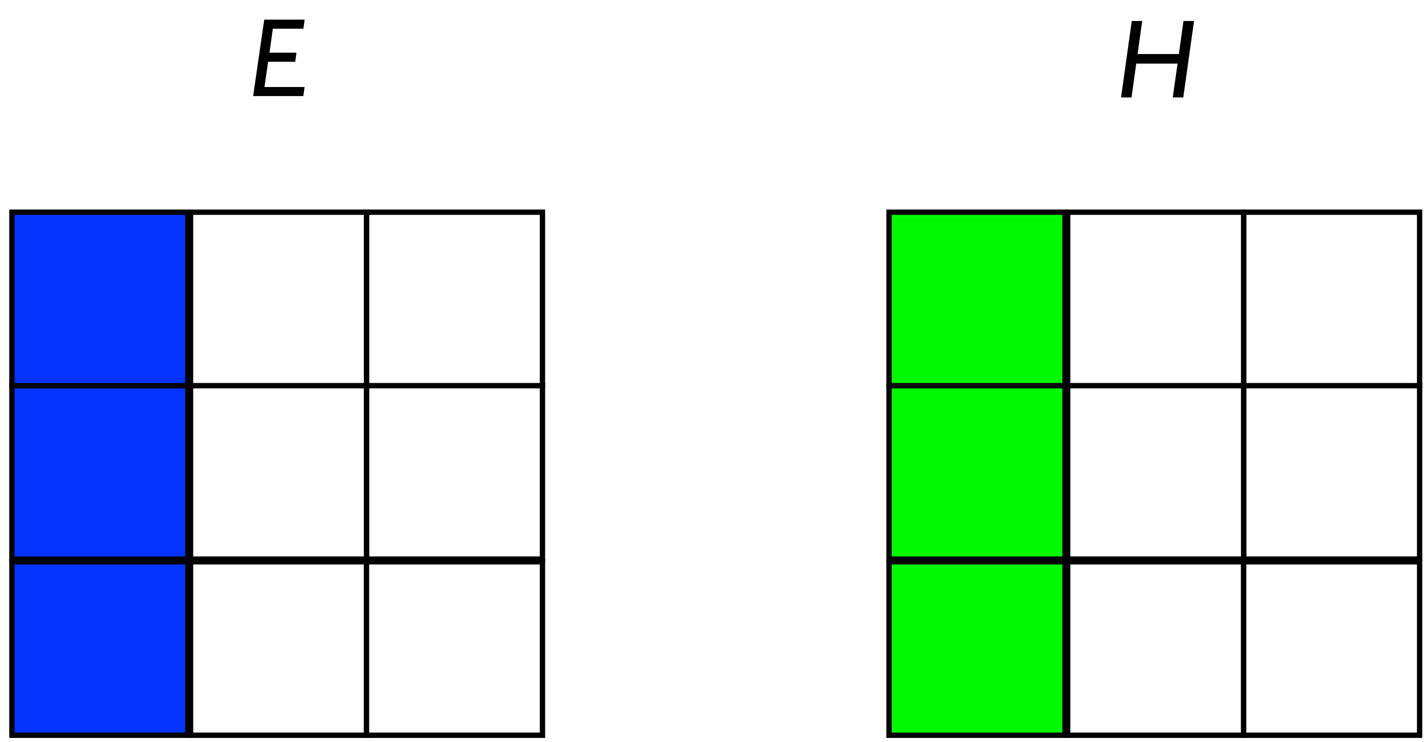
# Building of the Condensed Hessian

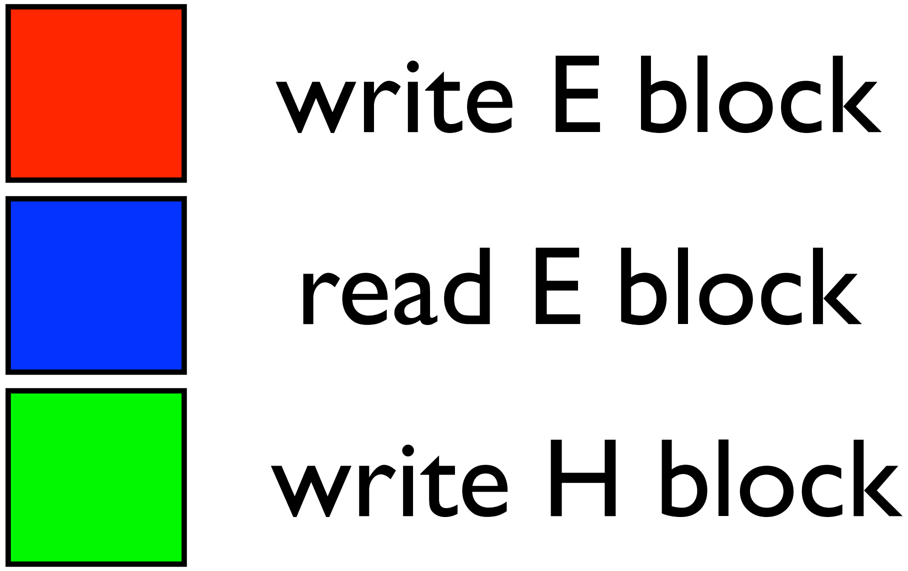


# Building of the Condensed Hessian

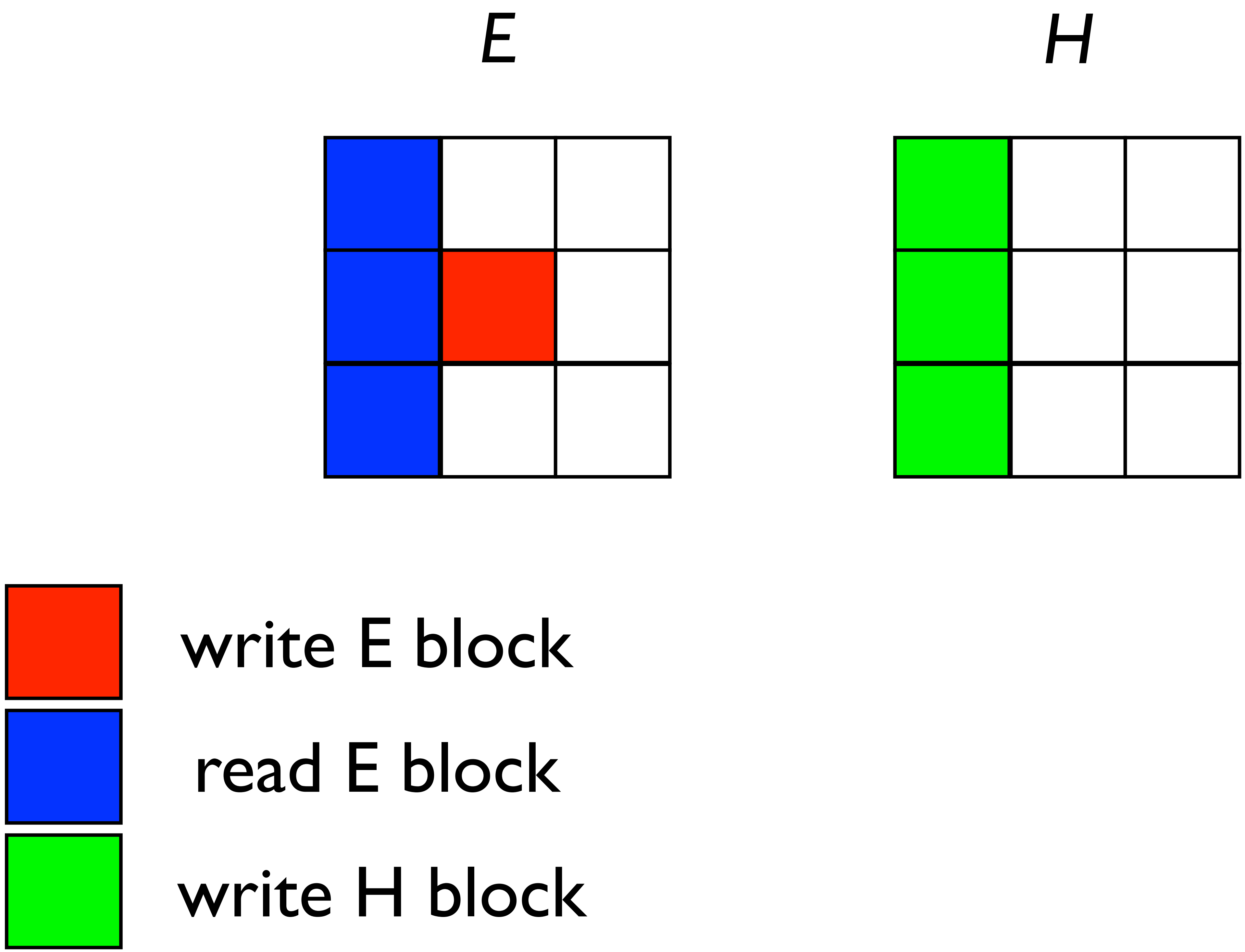


# Building of the Condensed Hessian

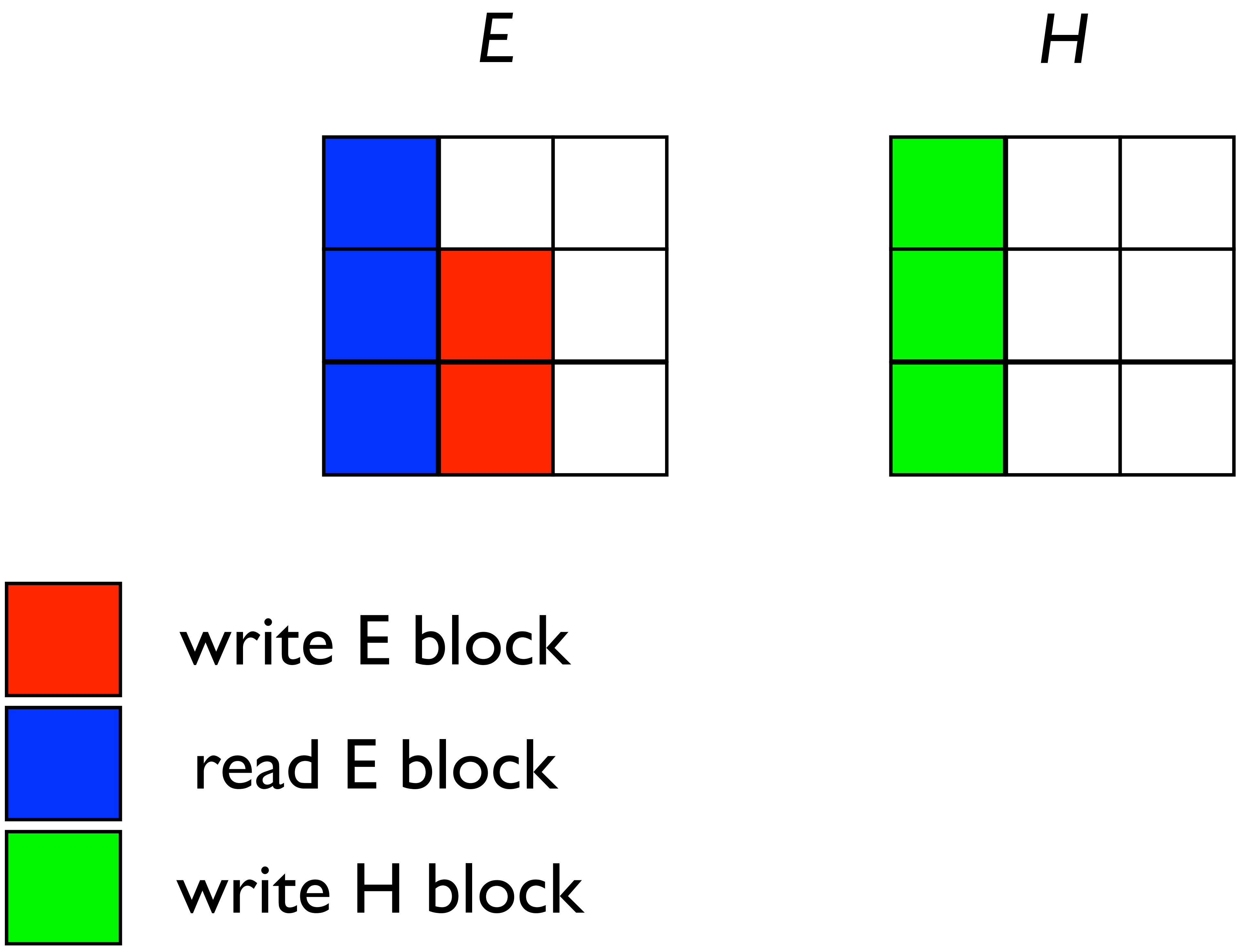


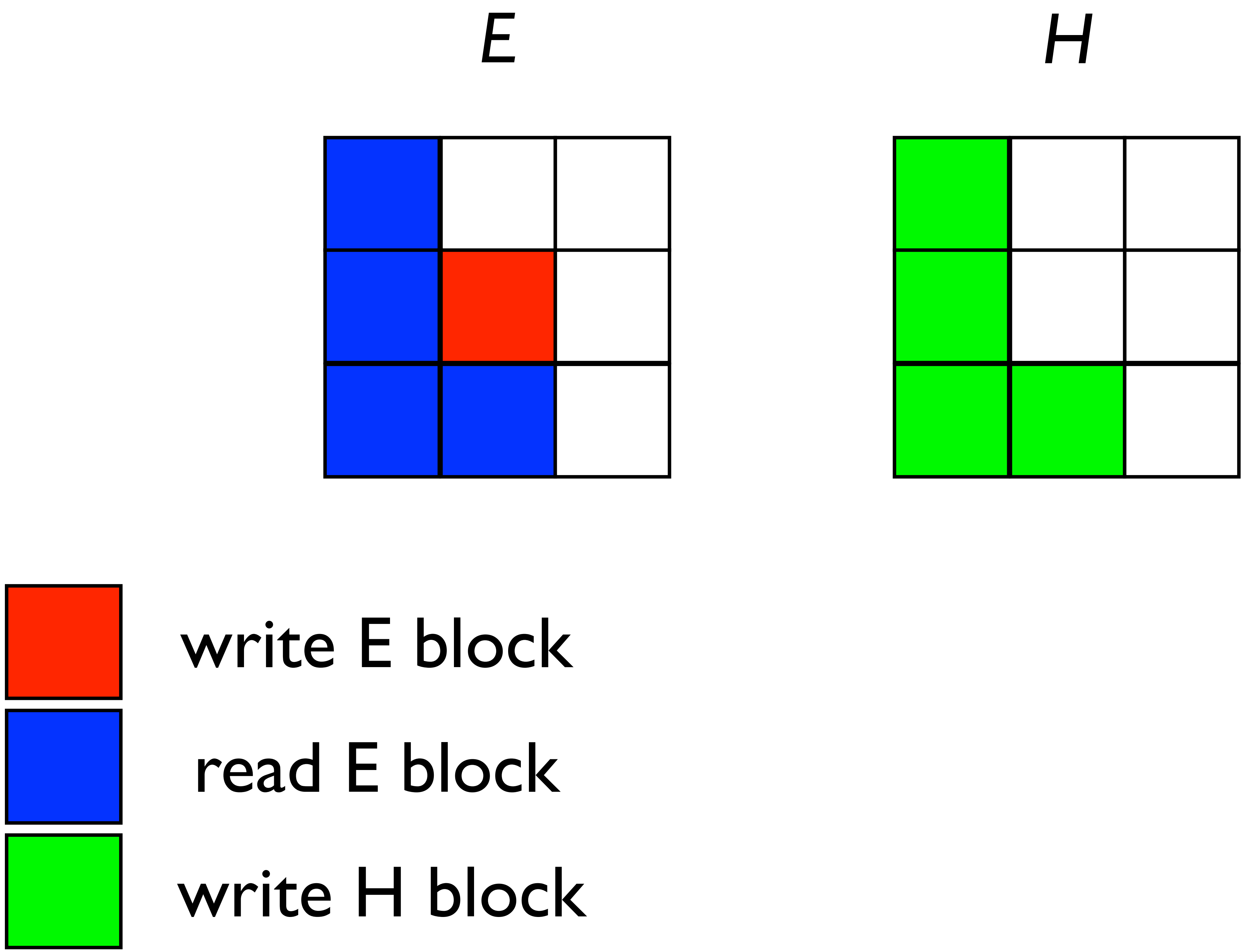
# Building of the Condensed Hessian



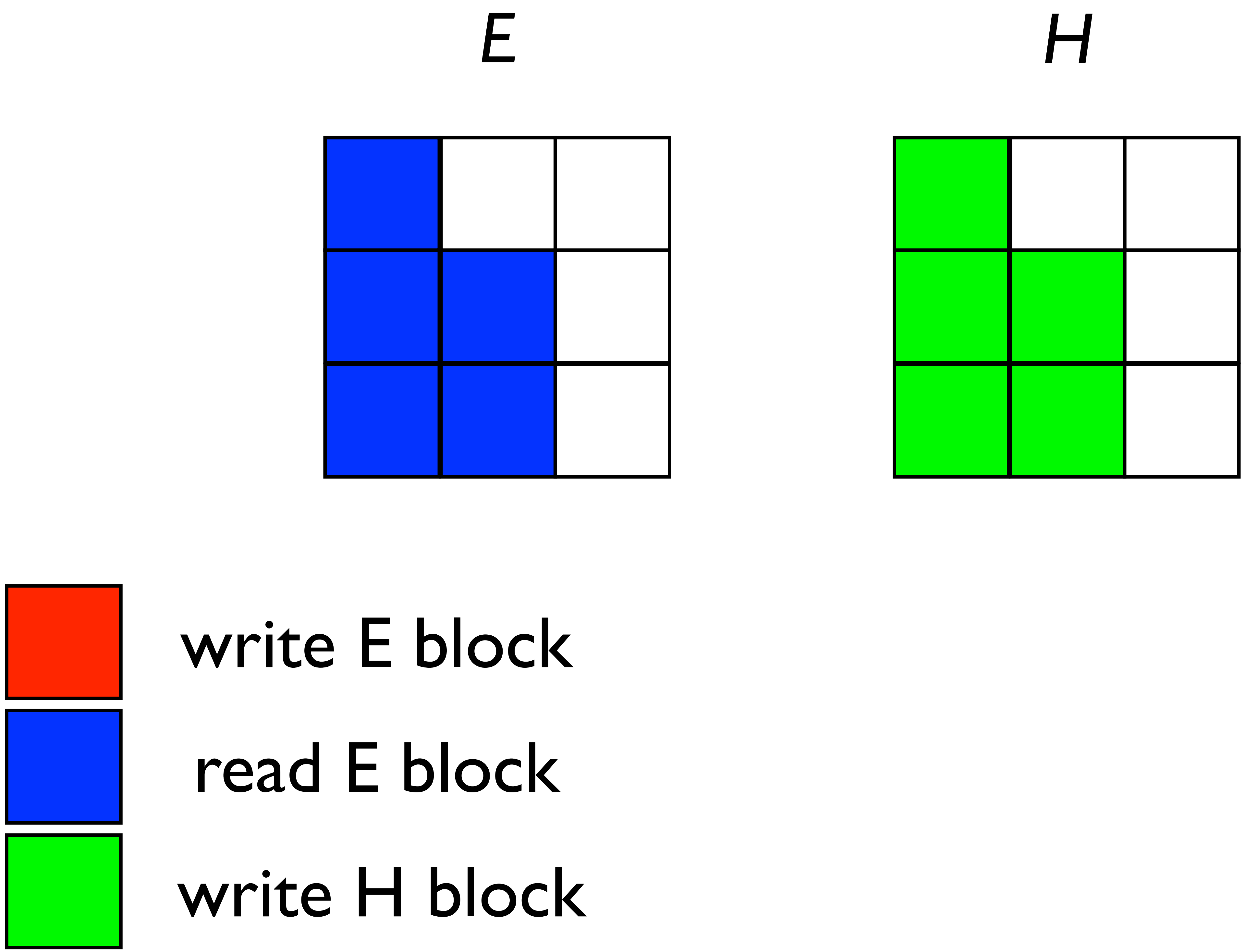
# Building of the Condensed Hessian



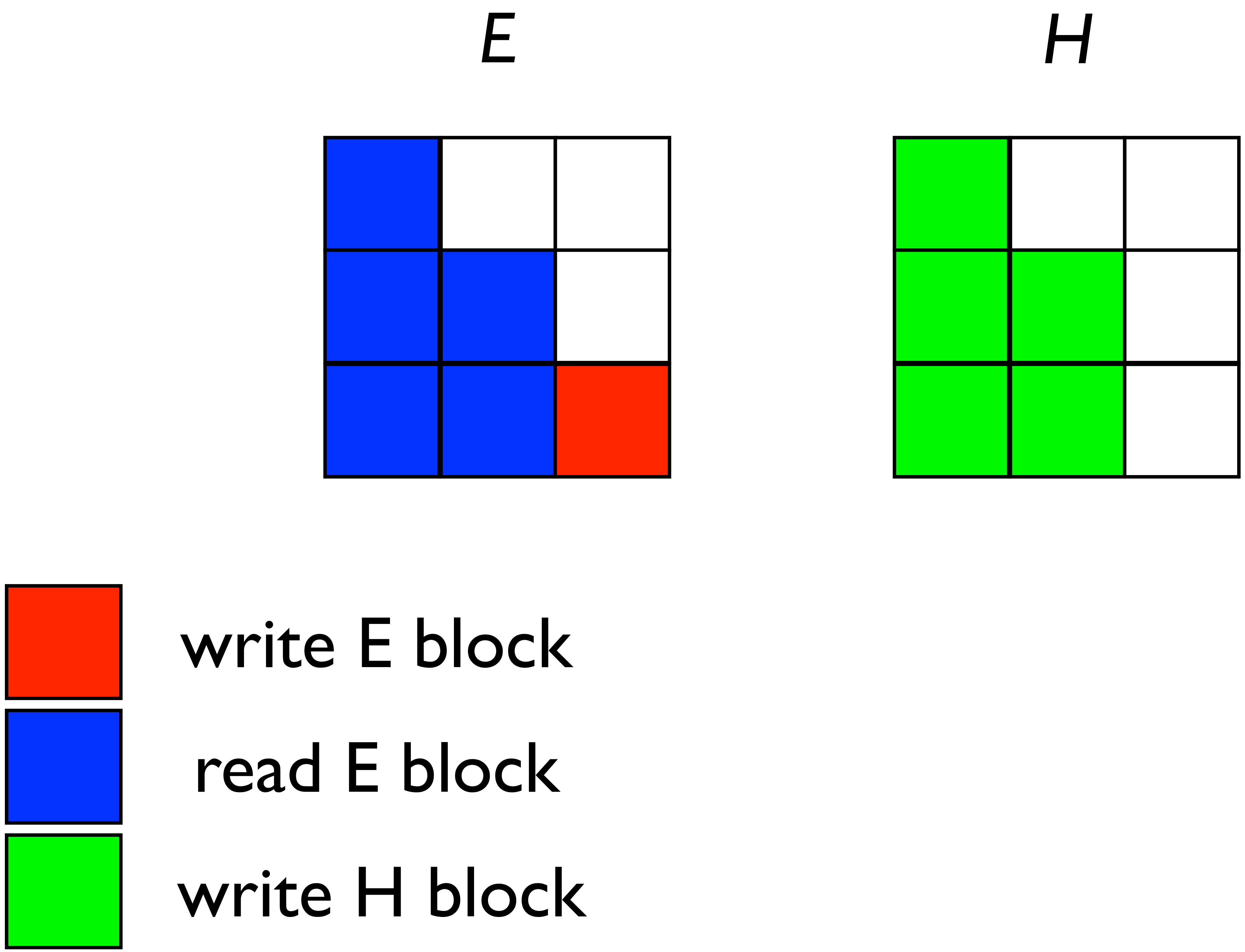
# Building of the Condensed Hessian



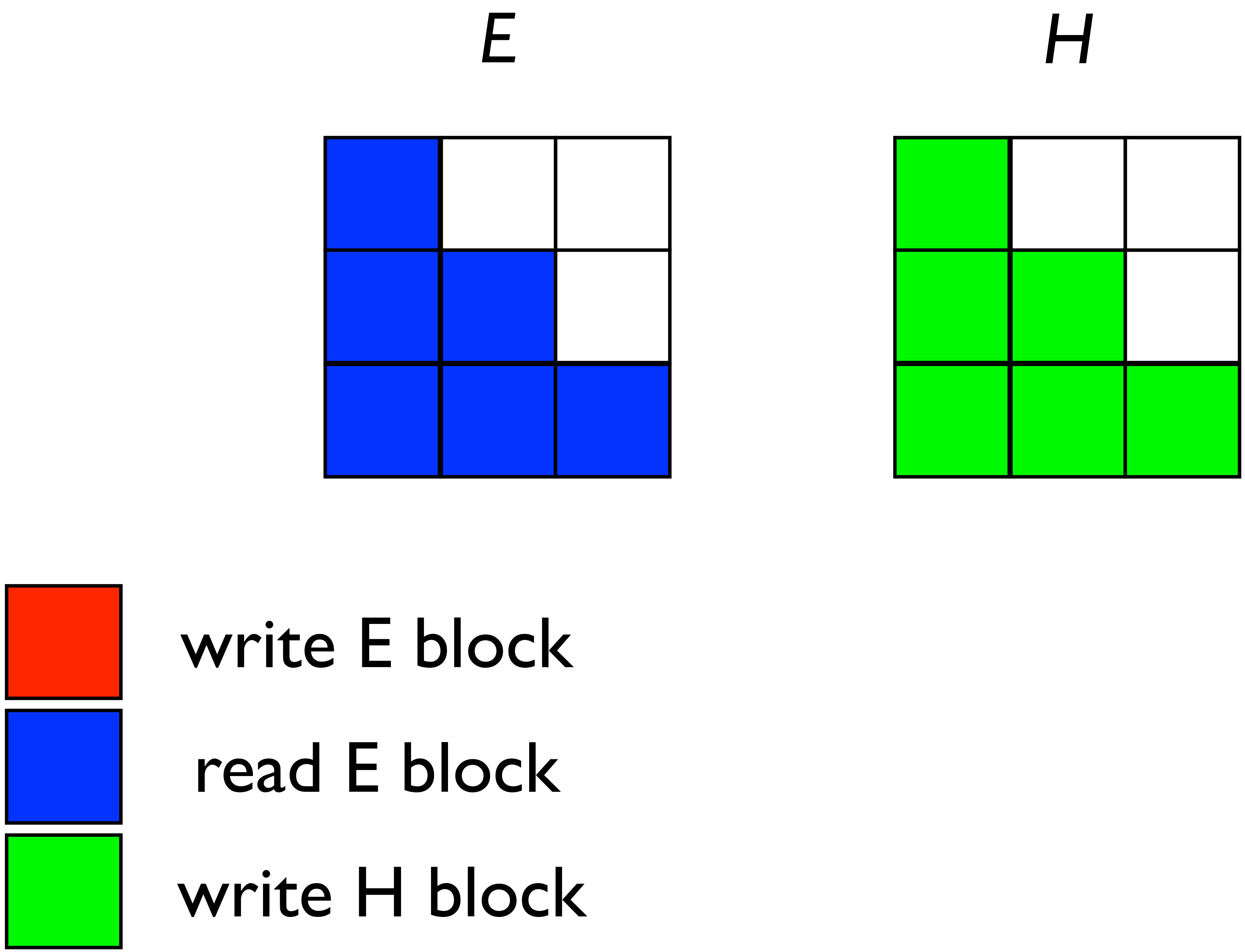
# Building of the Condensed Hessian



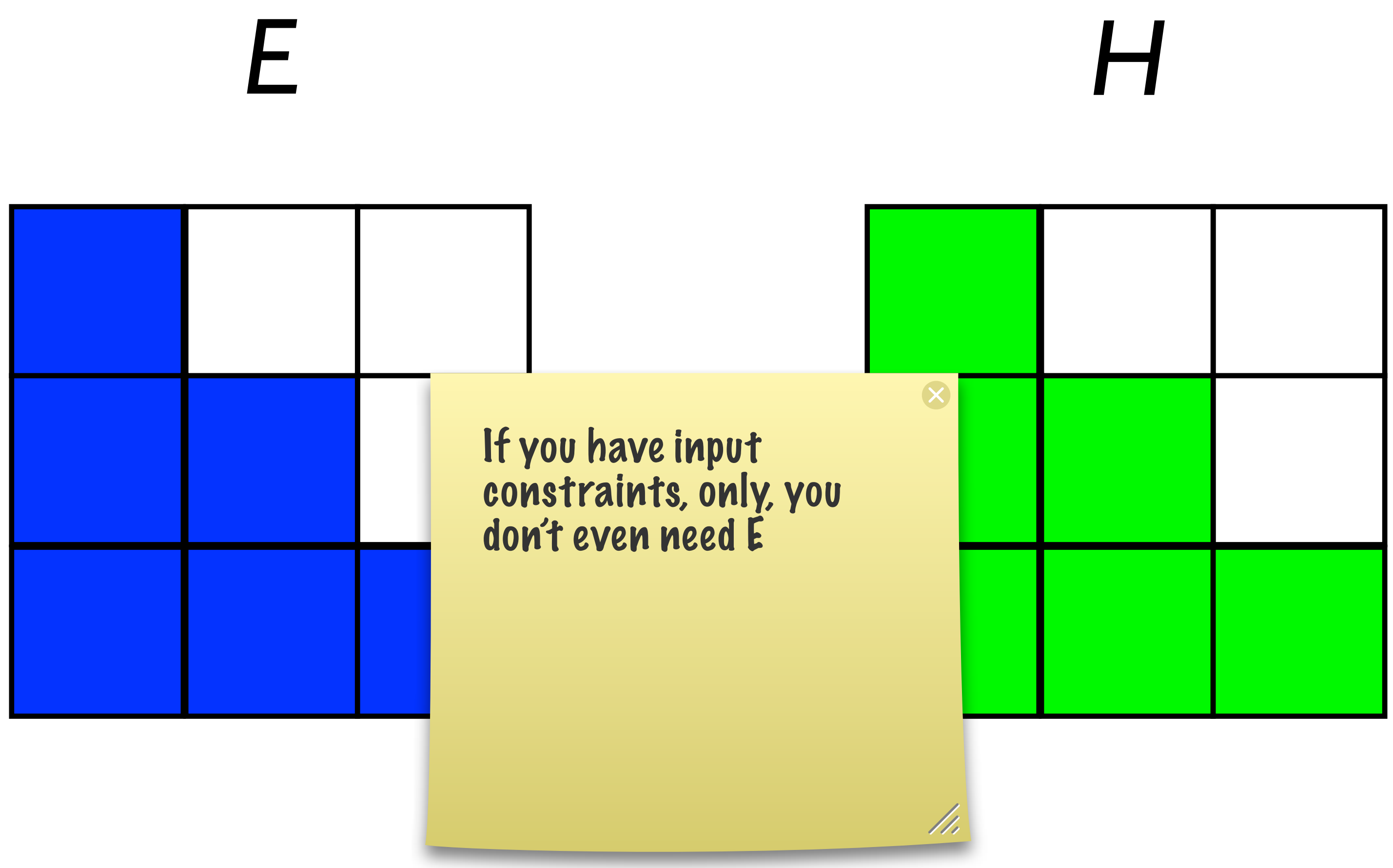
# Building of the Condensed Hessian



# Building of the Condensed Hessian



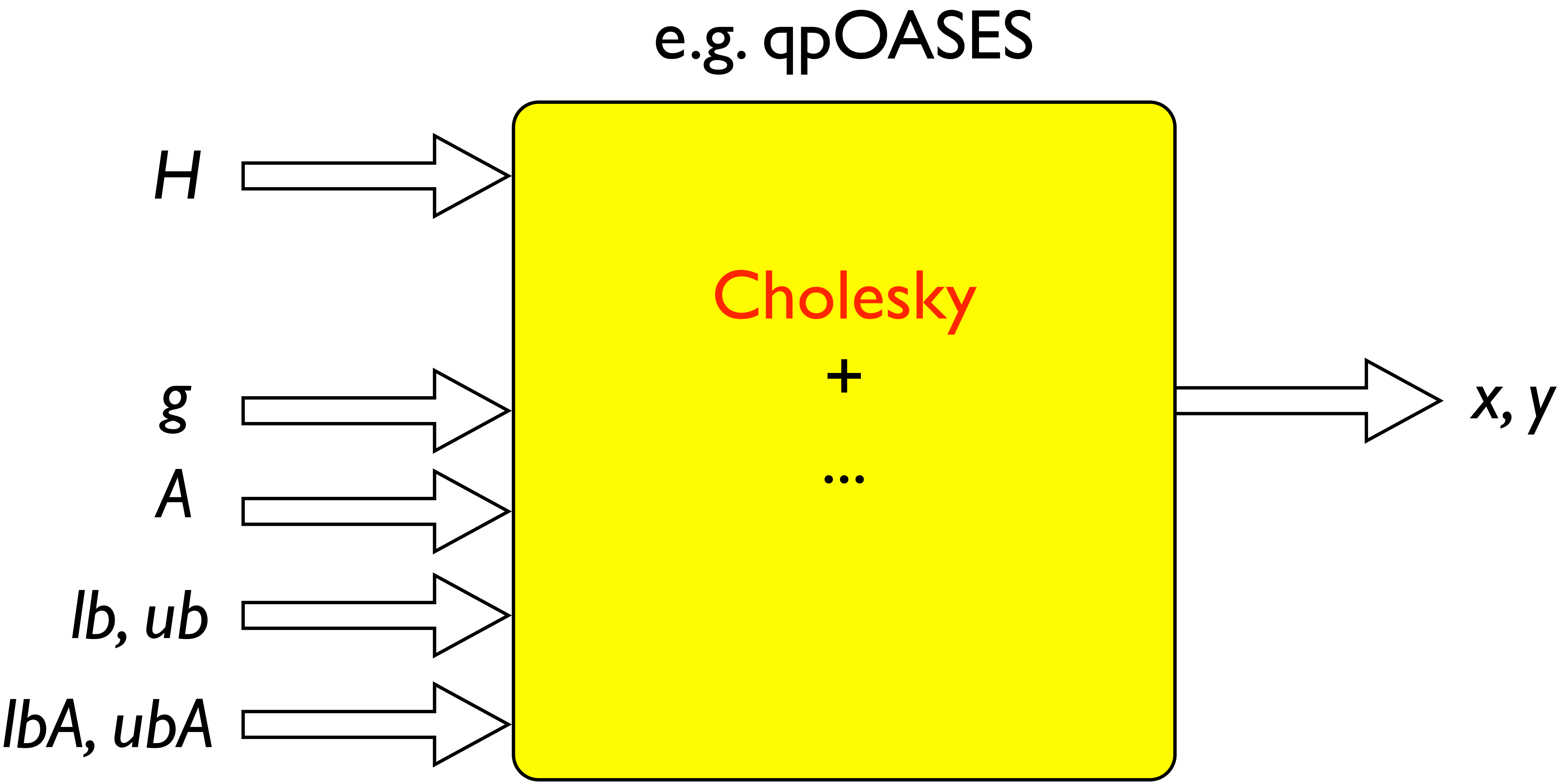
# Building of the Condensed Hessian



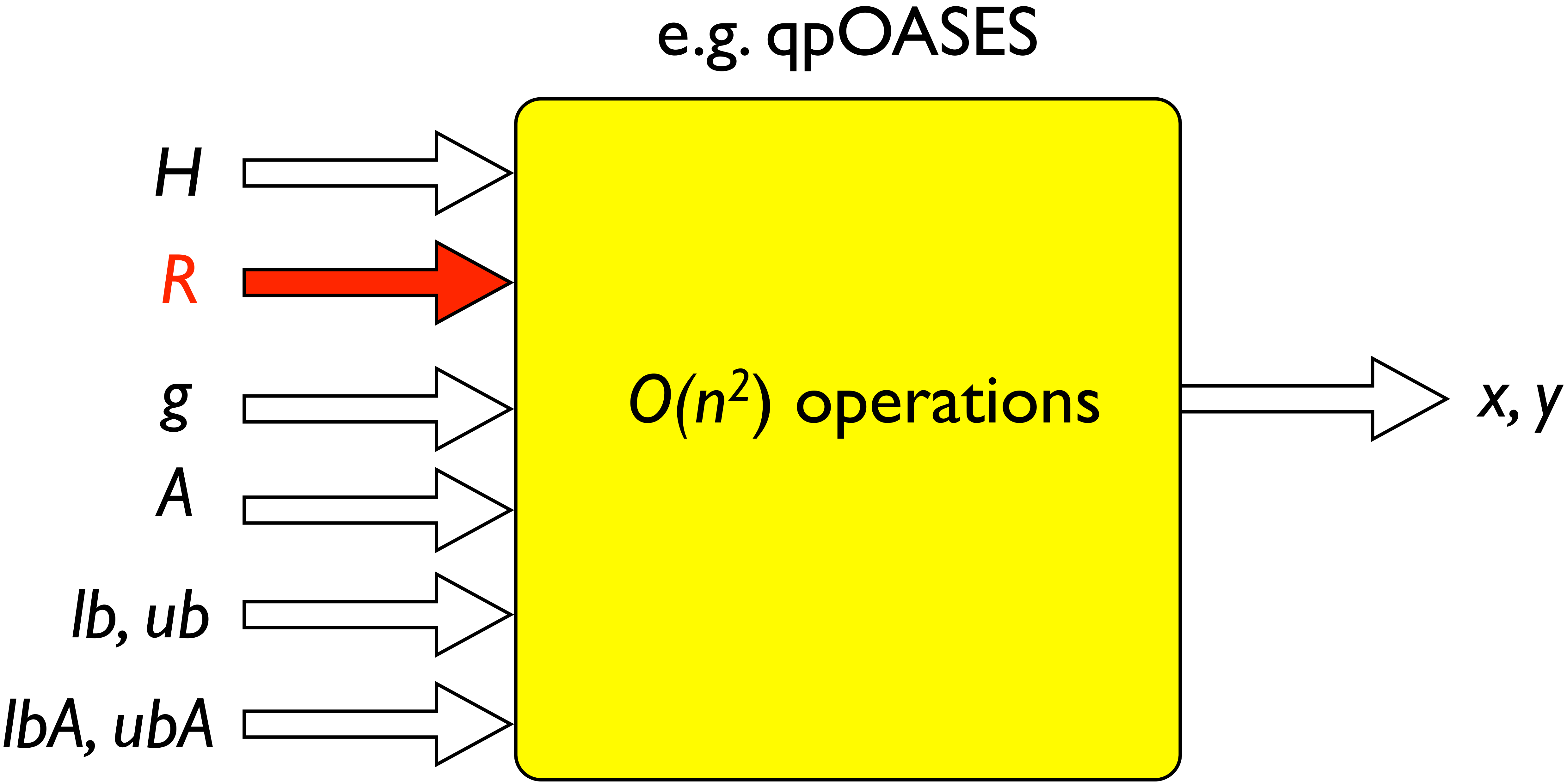

**Each column (row) of *H* can be built independently**

# Even better condensing?

# An interface to a QP solver



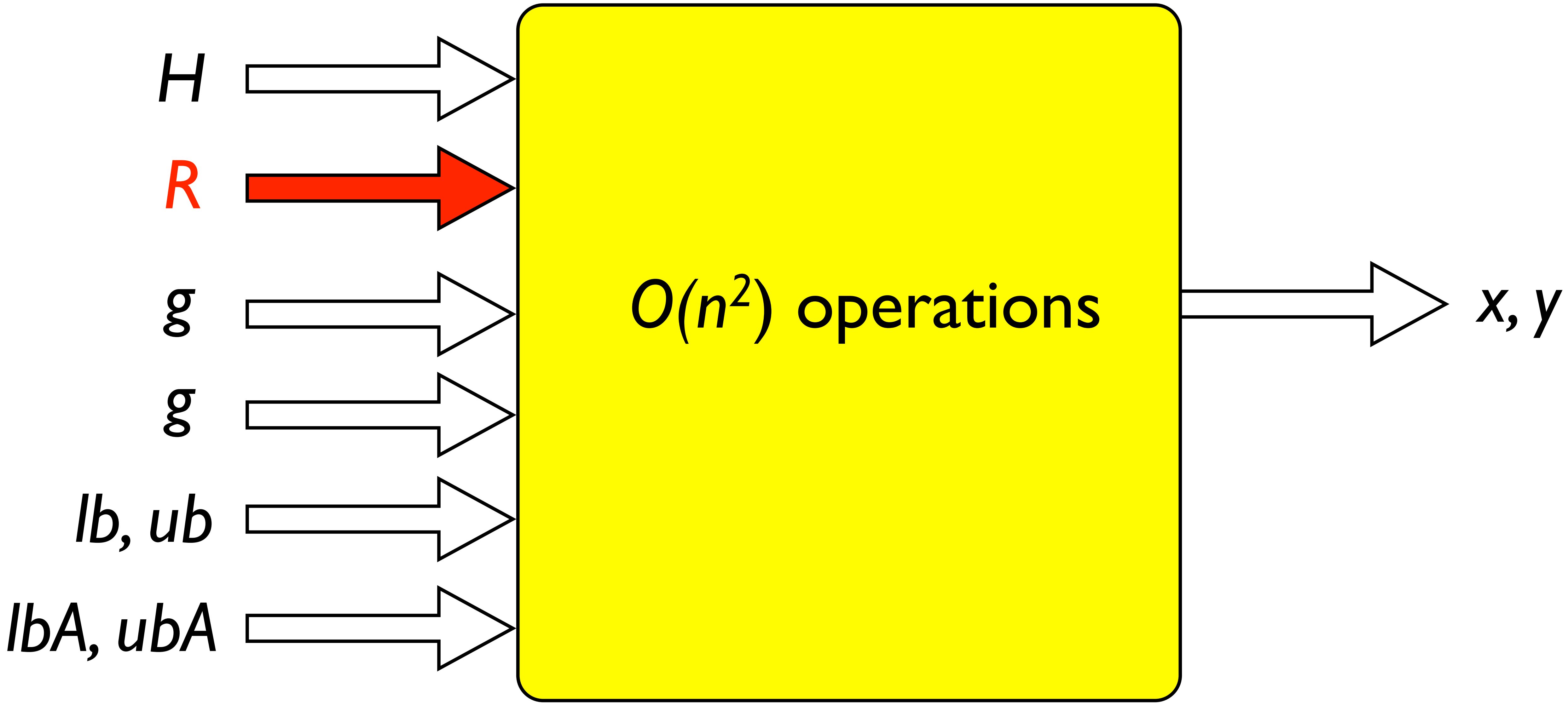
# An interface to a QP solver



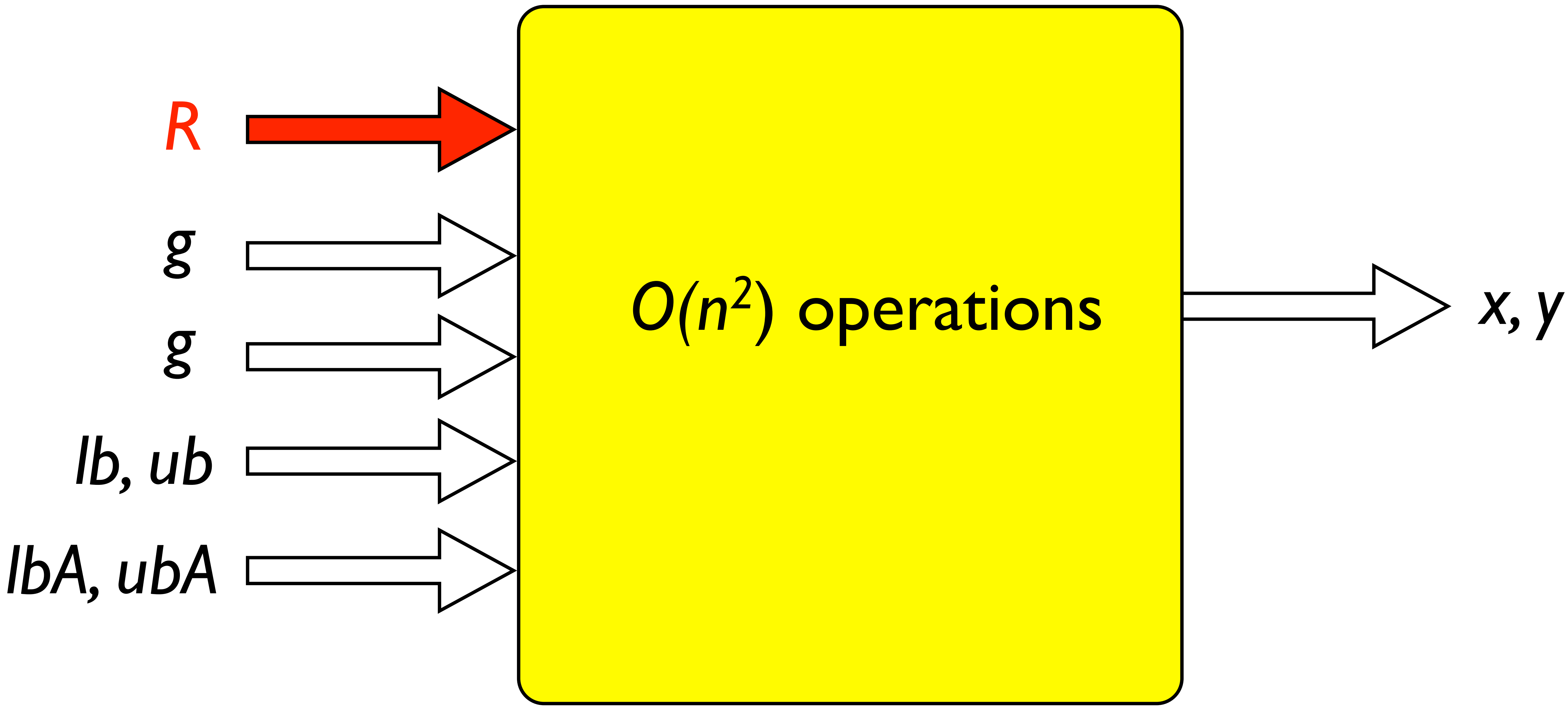
# $N^2$ factorization*

- Exchange $O(N^3 n_u^3)$ with $O(N^2 n_x^2 n_u)$ complexity for factorization of the Hessian
- Preliminary benchmarks show that it is **not** so smart to form both *H* and *R* for $n_x >> n_u$

* Frison2013



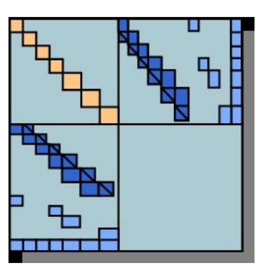
H
R
g
g
lb, ub
lbA, ubA
O(n2) operations
x, y


R
g
g
lb, ub
lbA, ubA
O(n2) operations
x, y

# And what about long horizons?

$$\underset{x_0,u_0,\ldots,x_N}{\text{minimize}} \quad \frac{1}{2}\sum_{k=0}^{N-1}\begin{bmatrix}x_k\\u_k\end{bmatrix}^T\begin{bmatrix}Q_k & S_k\\S_k^T & R_k\end{bmatrix}\begin{bmatrix}x_k\\u_k\end{bmatrix}+\begin{bmatrix}x_k\\u_k\end{bmatrix}^T\begin{bmatrix}g_k^{\mathrm{x}}\\g_k^{\mathrm{u}}\end{bmatrix}^T + \tfrac{1}{2}x_N^T Q_{\mathrm{e}} x_N + x_N^T g_{\mathrm{e}}^{\mathrm{x}}$$

$$\text{subject to} \quad \begin{array}{ll} x_{k+1} = A_k x_k + B_k u_k + c_k, & \text{for } k=0,\ldots,N-1 \\ x_k^{\mathrm{lo}} \le x_k \le x_k^{\mathrm{up}}, & \text{for } k=0,\ldots,N, \\ u_k^{\mathrm{lo}} \le u_k \le u_k^{\mathrm{up}}, & \text{for } k=0,\ldots,N-1, \\ b_k^{\mathrm{lo}} \le C_k x_k + D_k u_k \le b_k^{\mathrm{up}}, & \text{for } k=0,\ldots,N-1, \\ b_{\mathrm{e}}^{\mathrm{lo}} \le C_{\mathrm{e}} x_N \le b_{\mathrm{e}}^{\mathrm{up}}, & \end{array}$$



$$\underset{u}{\text{minimize}} \quad \frac{1}{2}u^T H_{\mathrm{C}} u + u^T g_{\mathrm{C}}$$

$$\text{subject to} \quad \begin{array}{l} u^{\mathrm{lo}} \le u \le u^{\mathrm{up}} \\ b_{\mathrm{C}}^{\mathrm{lo}} \le A_{\mathrm{C}} u \le b_{\mathrm{C}}^{\mathrm{up}} \end{array}$$

# FORCES

http://forces.ethz.ch

Structure exploiting QCQP solver

Implements primal-dual IP method

Auto-generated C-code

* Domahidi2012

# A Dual Newton Strategy*

C-code Software Implementation **qpDUNES**

$$\min_{z} \quad \sum_{k=0}^{N} \left( \frac{1}{2} z_k^{\mathrm{T}} H_k z_k + g_k^{\mathrm{T}} z_k \right)$$

$$\text{s.t.} \quad E_{k+1} z_{k+1} = C_k z_k + c_k \qquad \forall\, k = 0, \ldots, N-1$$

$$\underline{d}_k \leq D_k z_k \leq \overline{d}_k \qquad \forall\, k = 0, \ldots, N.$$

* Ferreau2012, Frasch2014

$$\mathcal{L}(z,\lambda) = \sum_{k=0}^{N} \left( \frac{1}{2} z_k^{\mathrm{T}} H_k z_k + g_k^{\mathrm{T}} z_k + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} -E_k \\ C_k \end{bmatrix} z_k + \lambda_{k+1}^{\mathrm{T}} c_k \right)$$

$$=: \sum_{k=0}^{N} L_k(z_k, \lambda_k, \lambda_{k+1}),$$

$$
\begin{aligned}
\mathcal{L}(z,\lambda) &= \sum_{k=0}^{N} \left( \frac{1}{2} z_k^{\mathrm{T}} H_k z_k + g_k^{\mathrm{T}} z_k + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} -E_k \\ C_k \end{bmatrix} z_k + \lambda_{k+1}^{\mathrm{T}} c_k \right) \\
&=: \sum_{k=0}^{N} L_k(z_k, \lambda_k, \lambda_{k+1}),
\end{aligned}
$$

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

# ACADO toolkit [Houska 2009]

# **www.acadotoolkit.org**

- Open source package (LGPL)
- Depends only on the standard 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

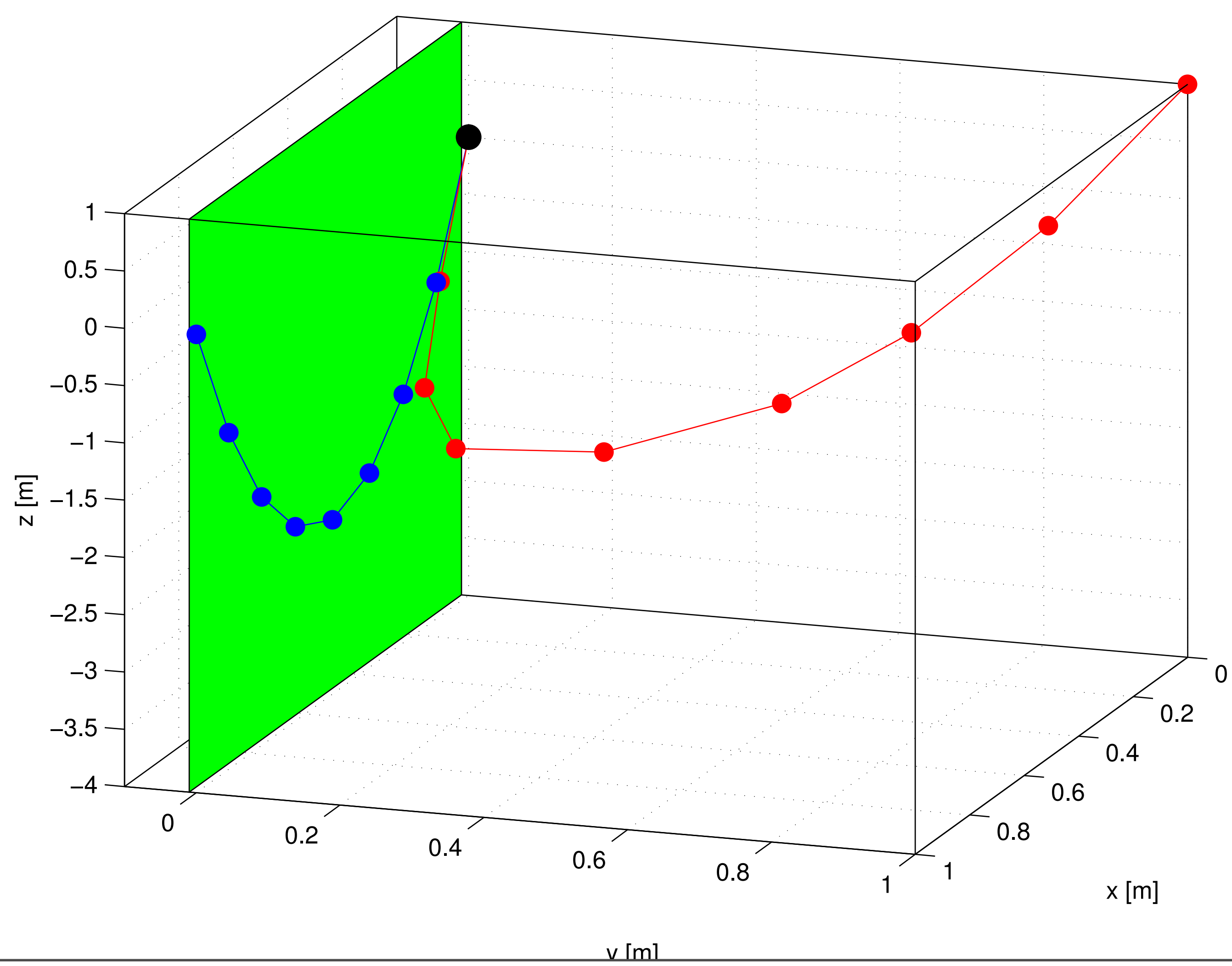
# ACADO Code Generation Tool *

- 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

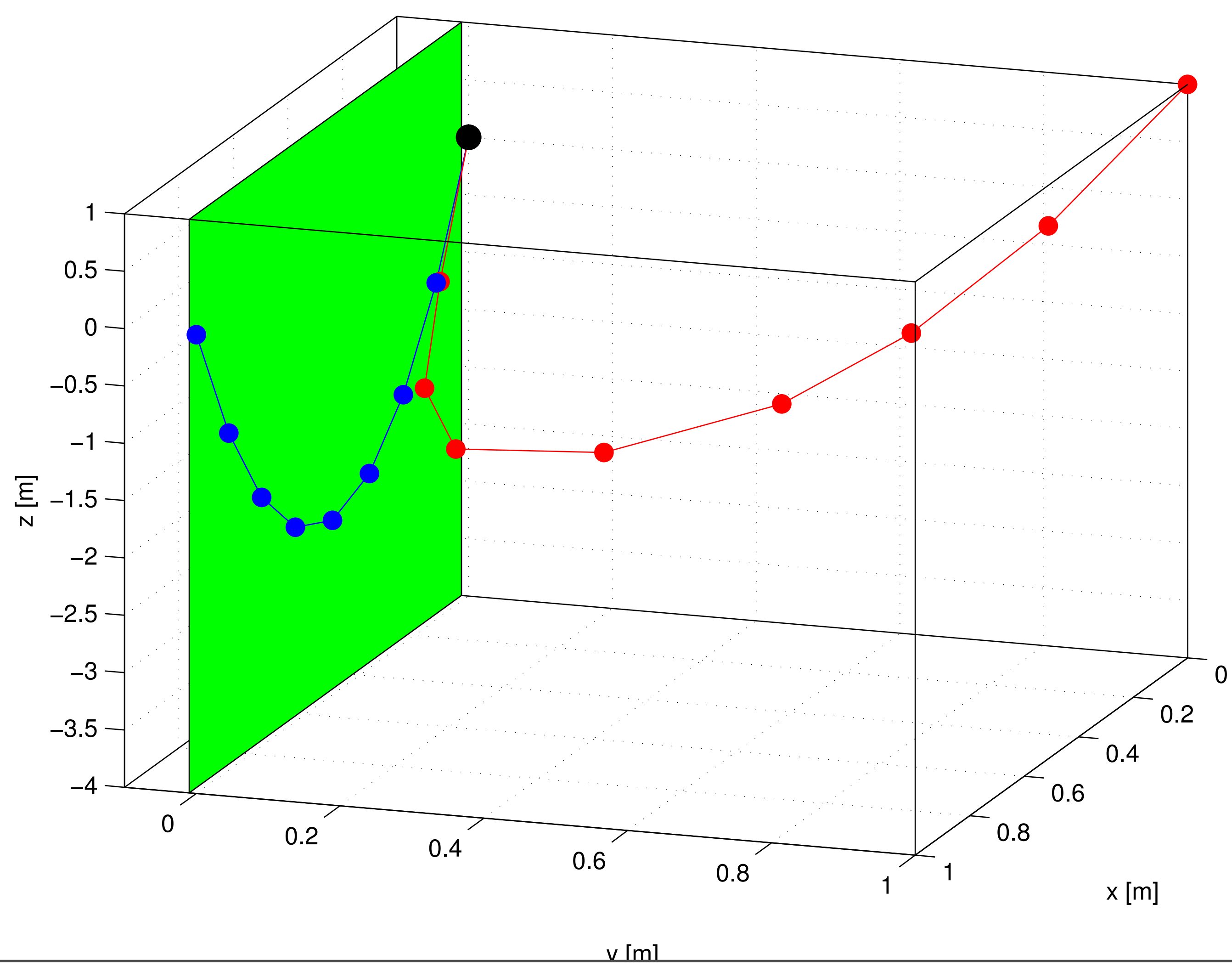
# Results

- Benchmark problem is a chain-mass problem [Wirsching 2006]



# Results

- Benchmark problem is a chain-mass problem [Wirsching 2006]

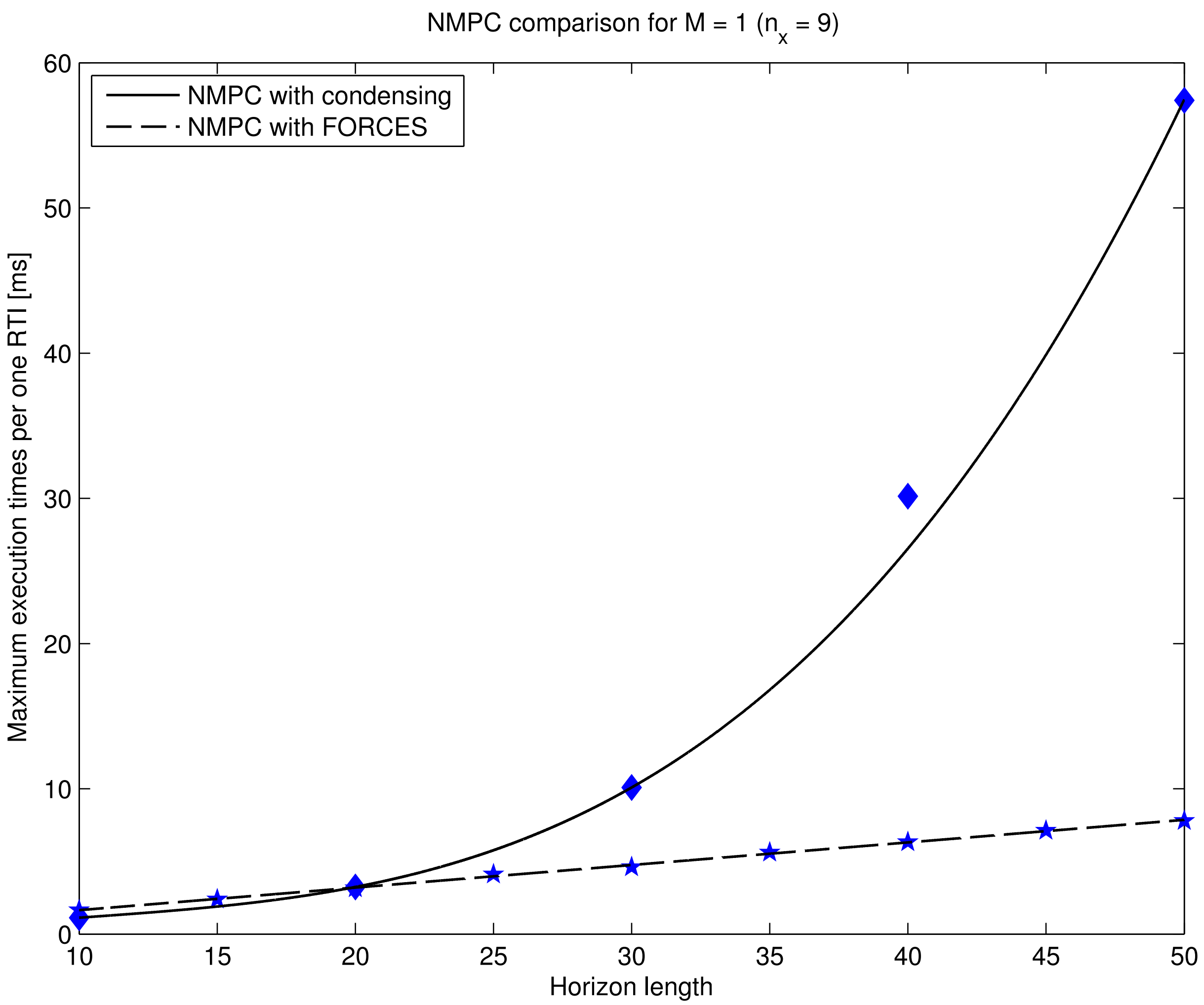



*M* masses

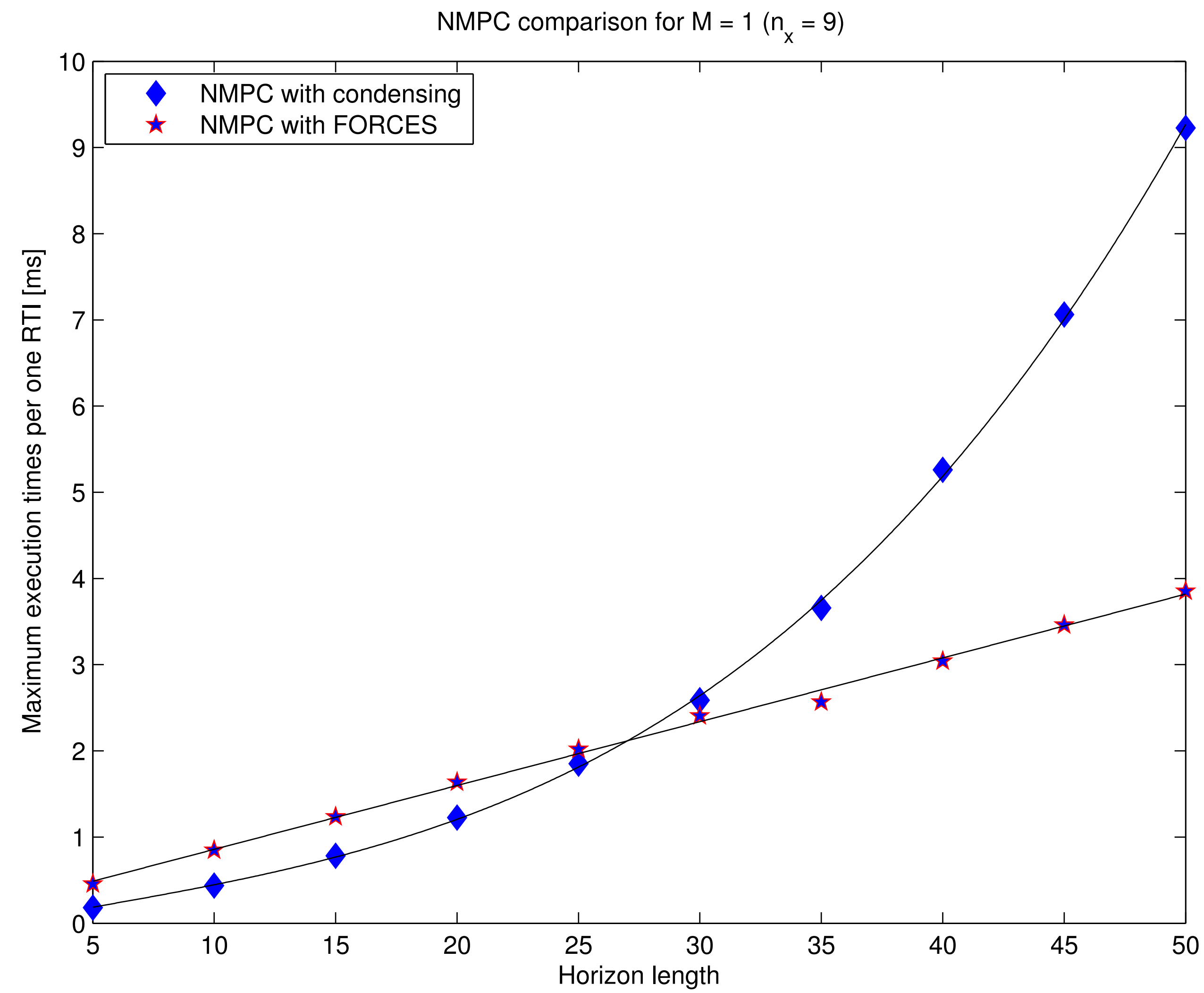
*3(2M + 1)* states

3 controls

# $M = 1, N = 5..\ 50; n_x = 9, n_u = 3$

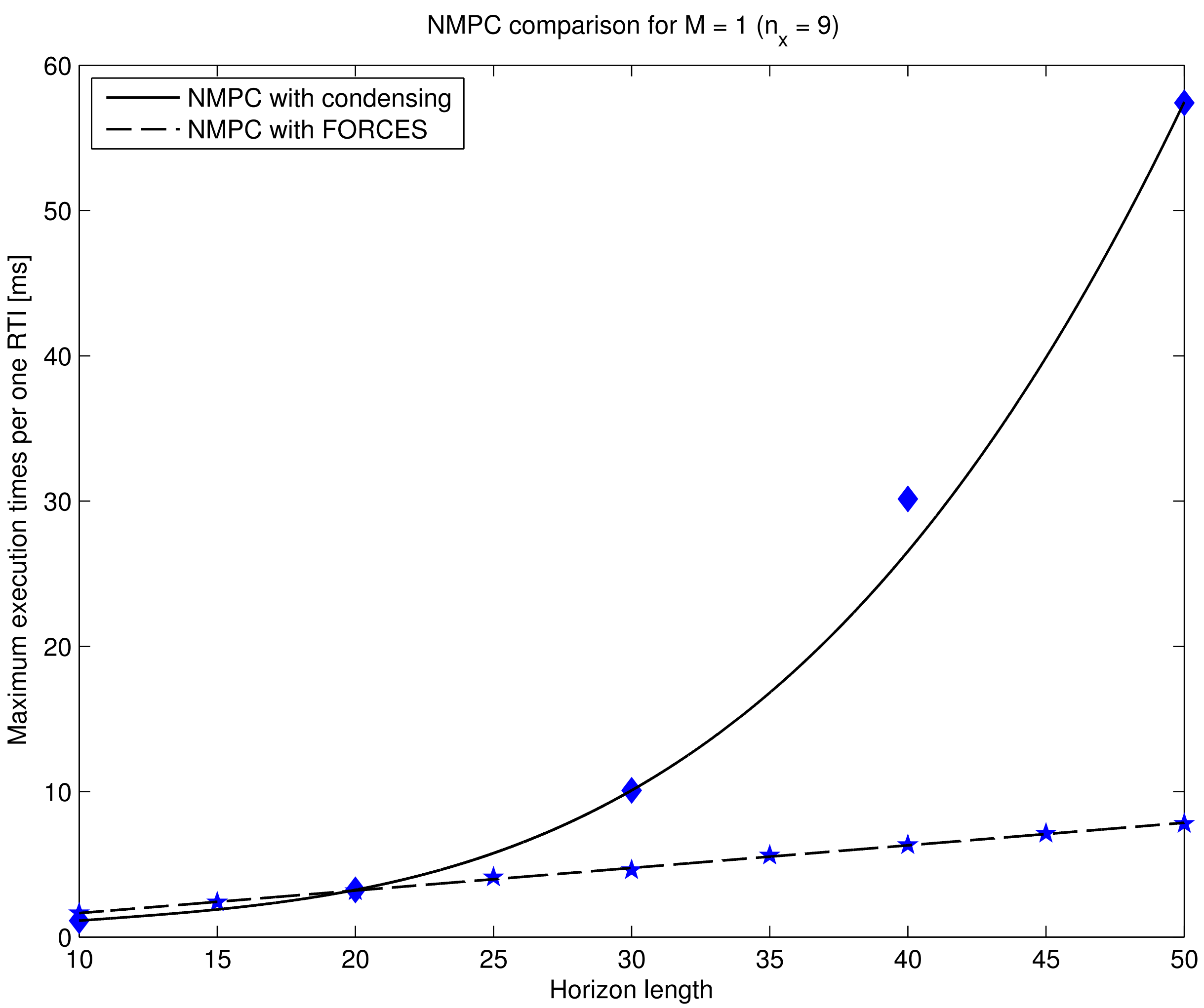



Old

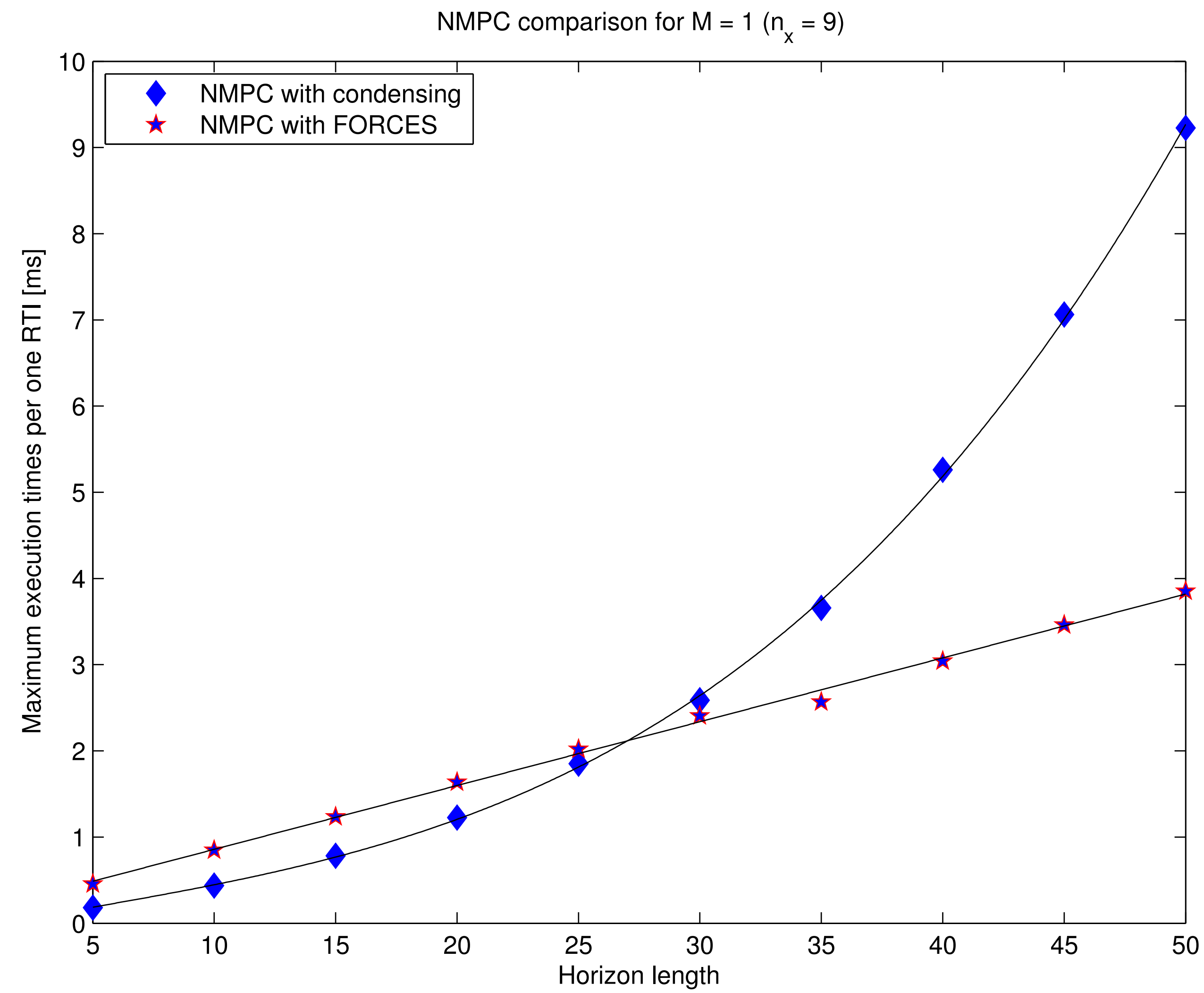



New

# $M = 1, N = 5..50; n_x = 9, n_u = 3$



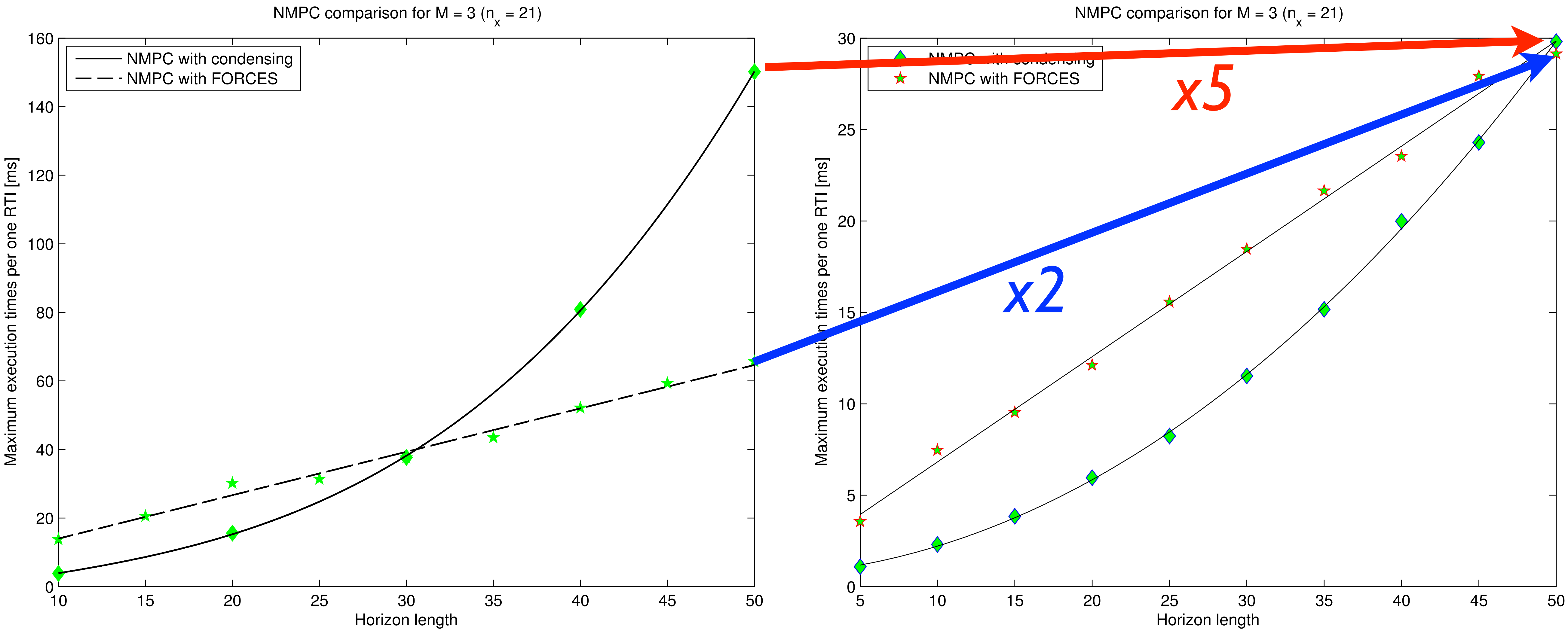

Old




New

Improvements: x6 for condensing & x3 for FORCES

# $M = 3, N = 5..\ 50; n_x = 21, n_u = 3$

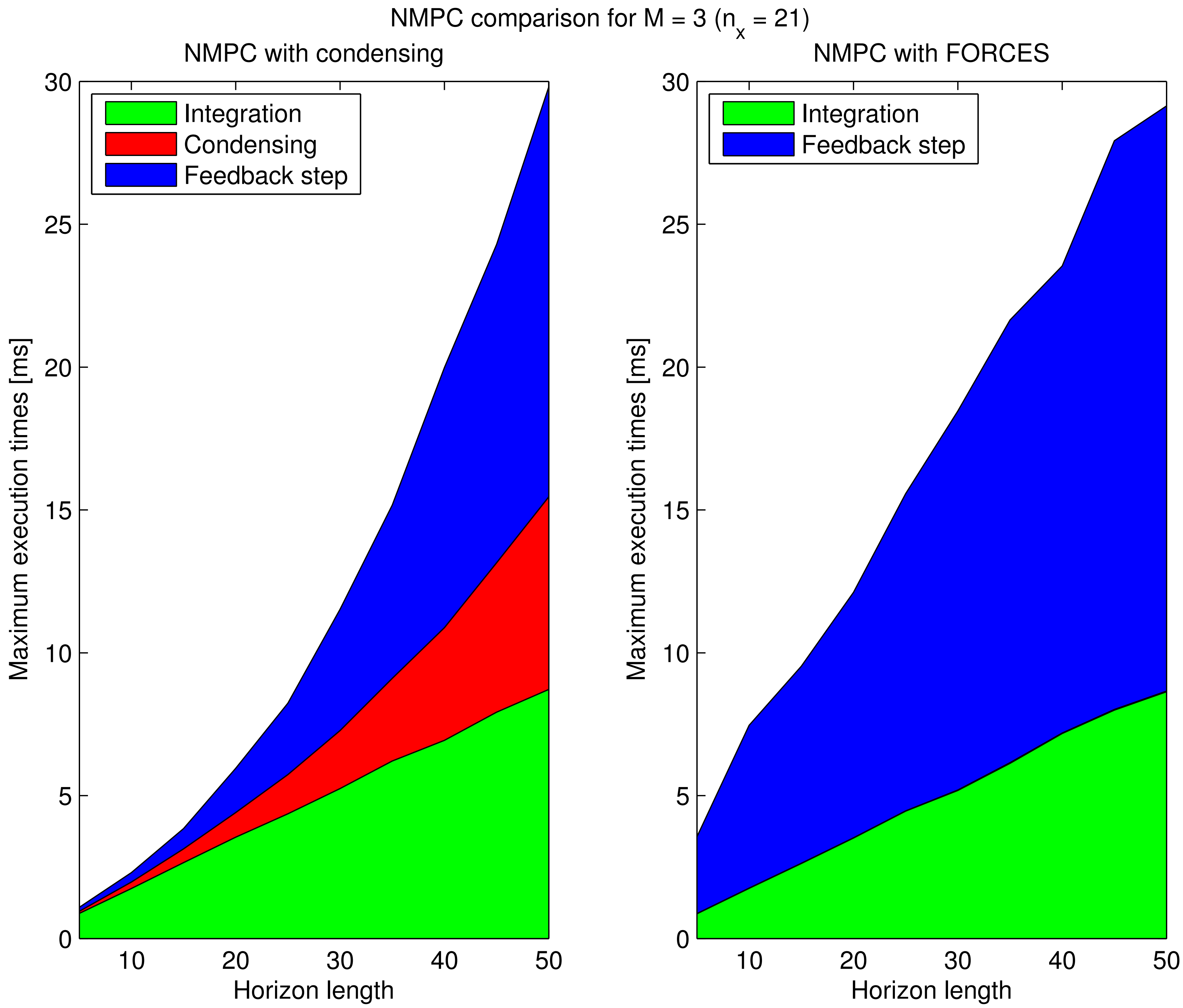



Old

New

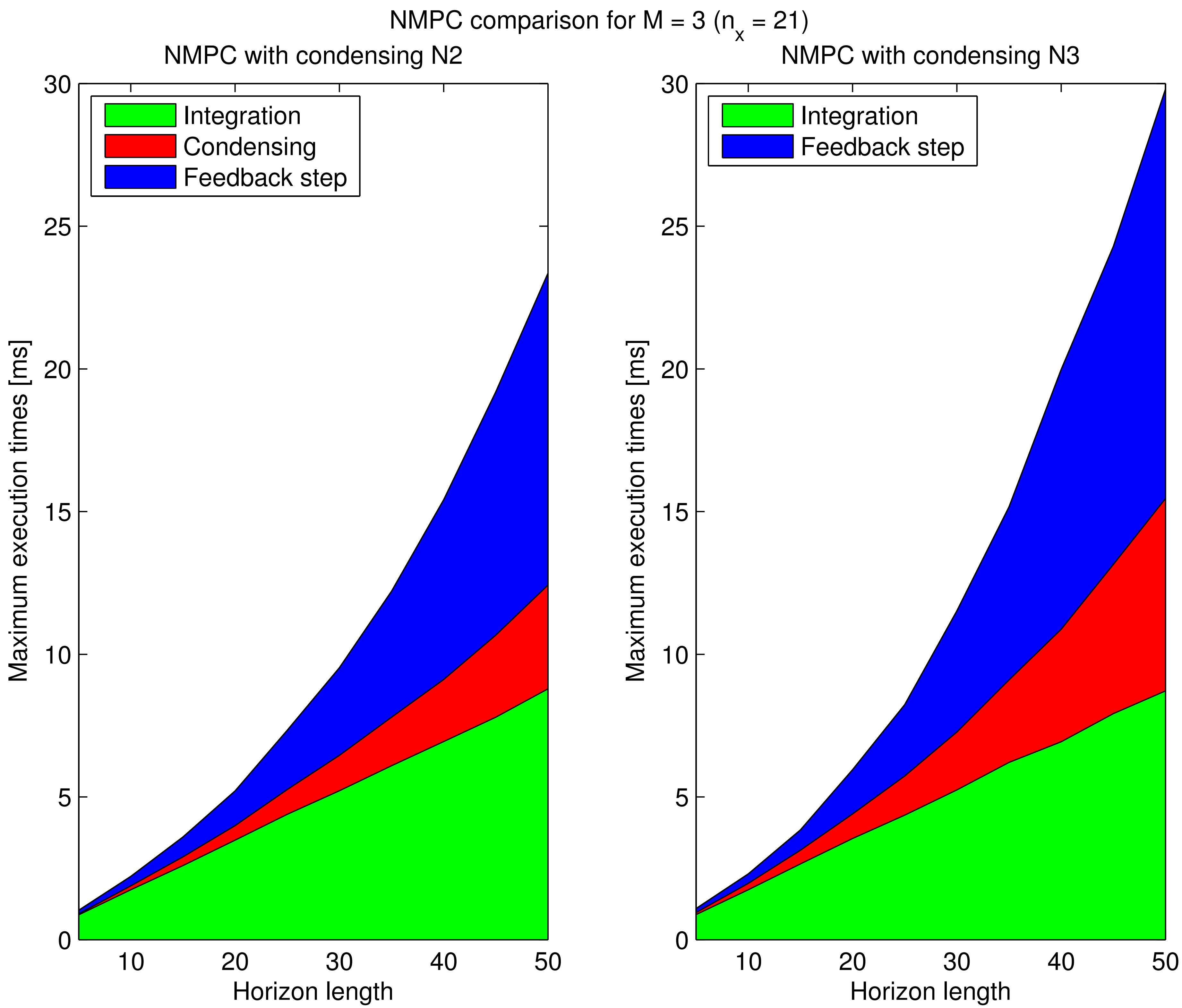
# $N^3$ Condensed vs FORCES NMPC

## $n_x = 21, n_u = 3$

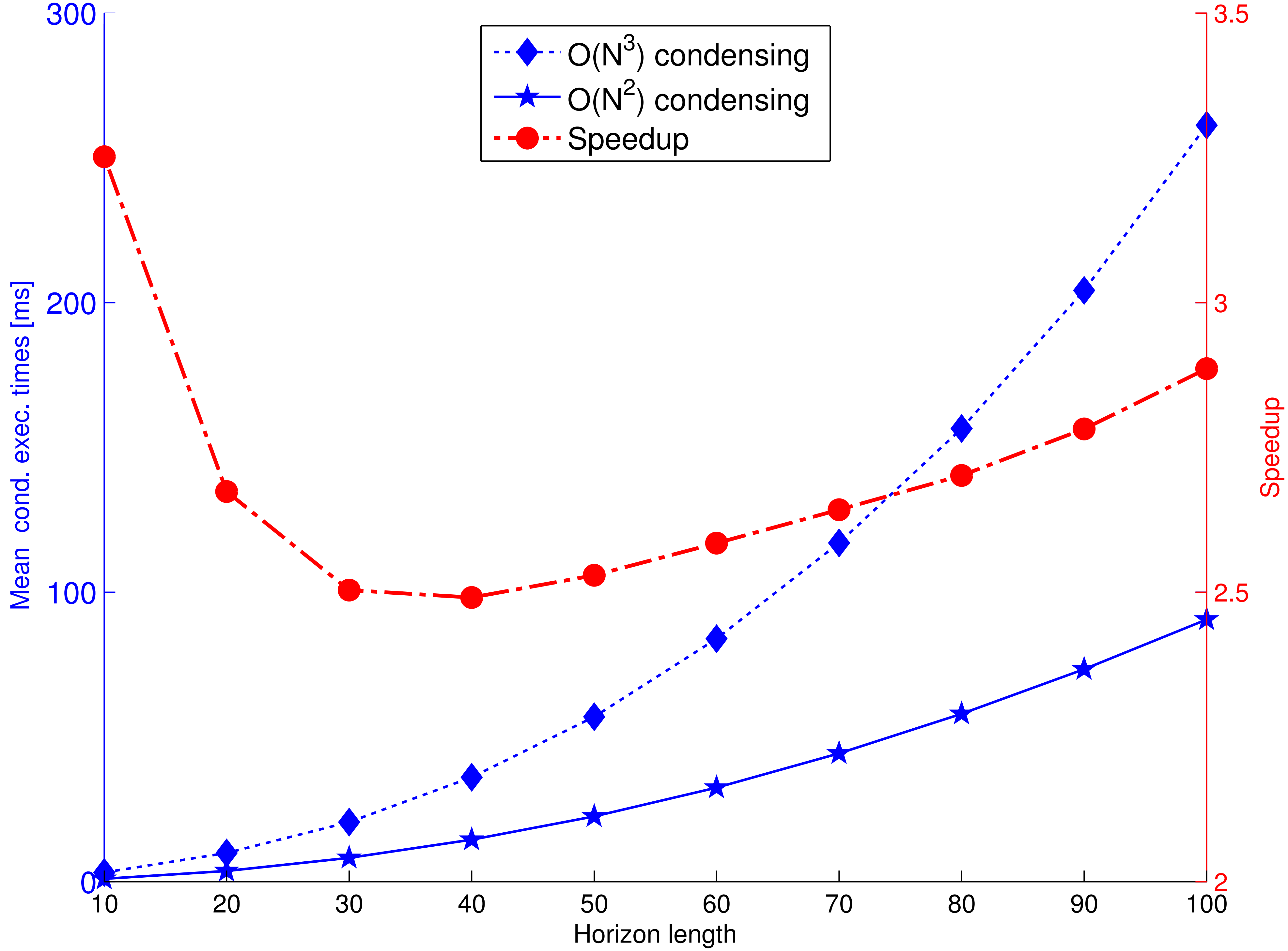


# $N^3$ Condensed vs $N^2$ Cond. NMPC

## $n_x = 21, n_u = 3$

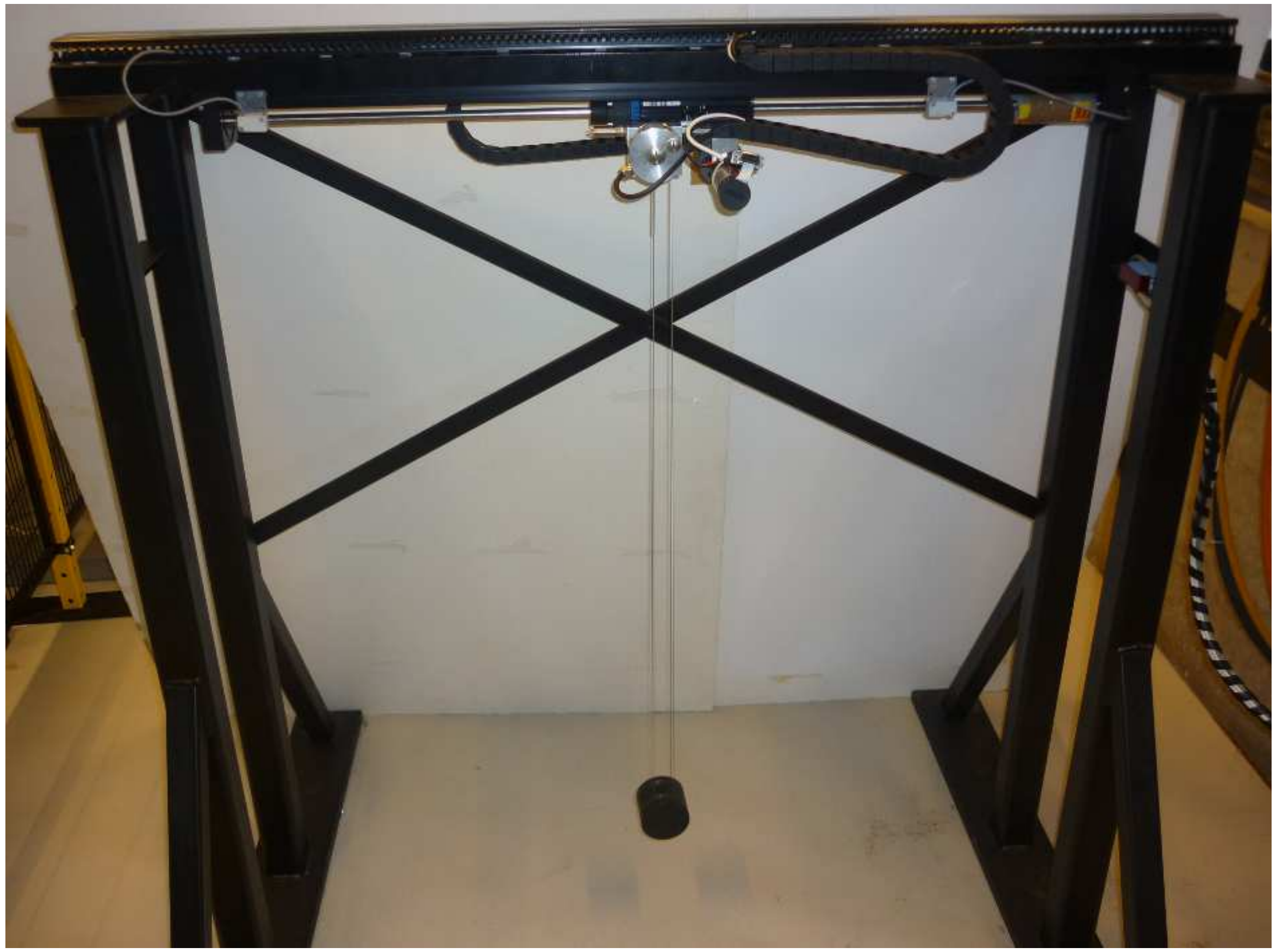


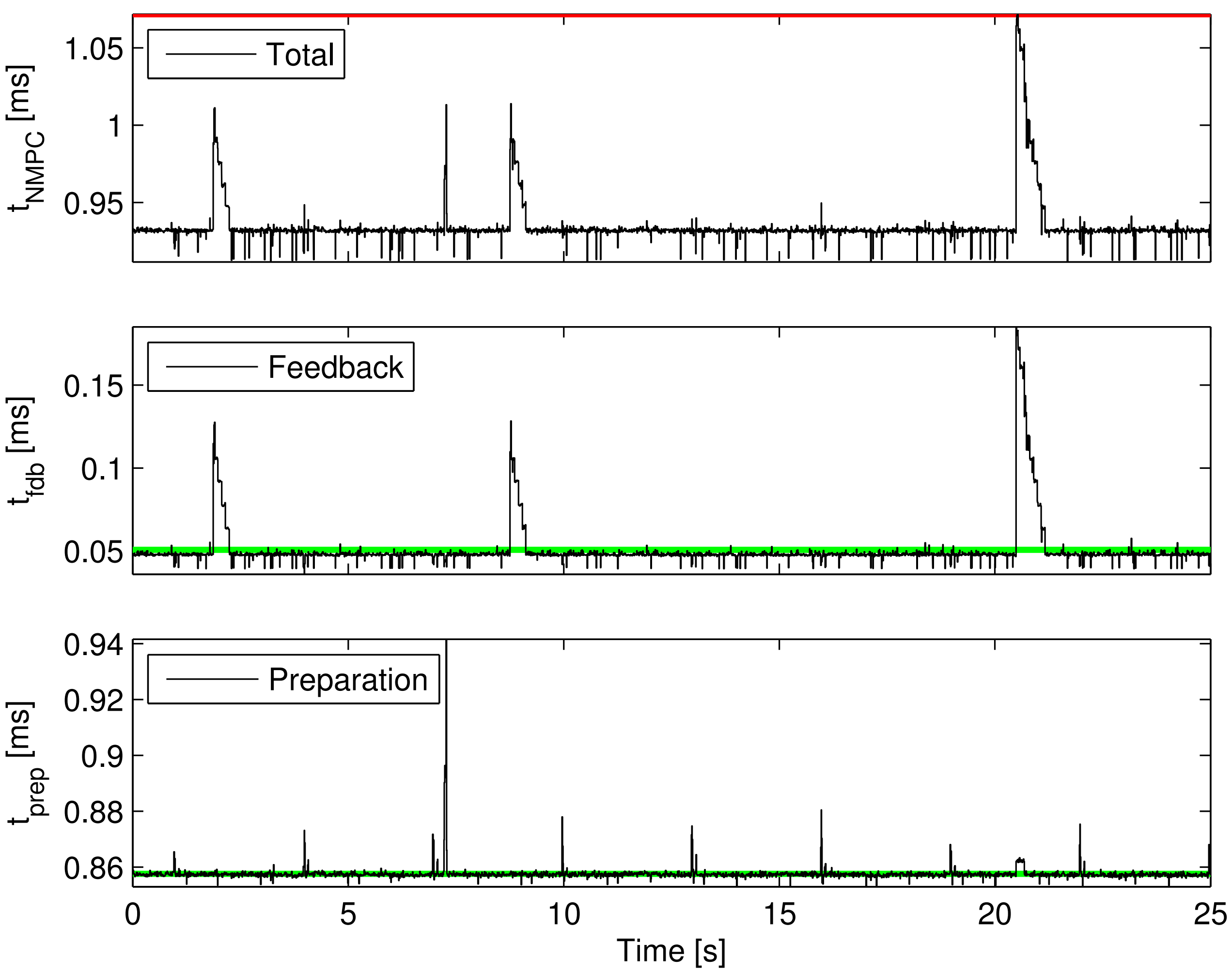
# $N^2$ vs $N^3$ condensing
# $n_x = 57, n_u = 3$



# Real-world apps

# The Overhead Crane



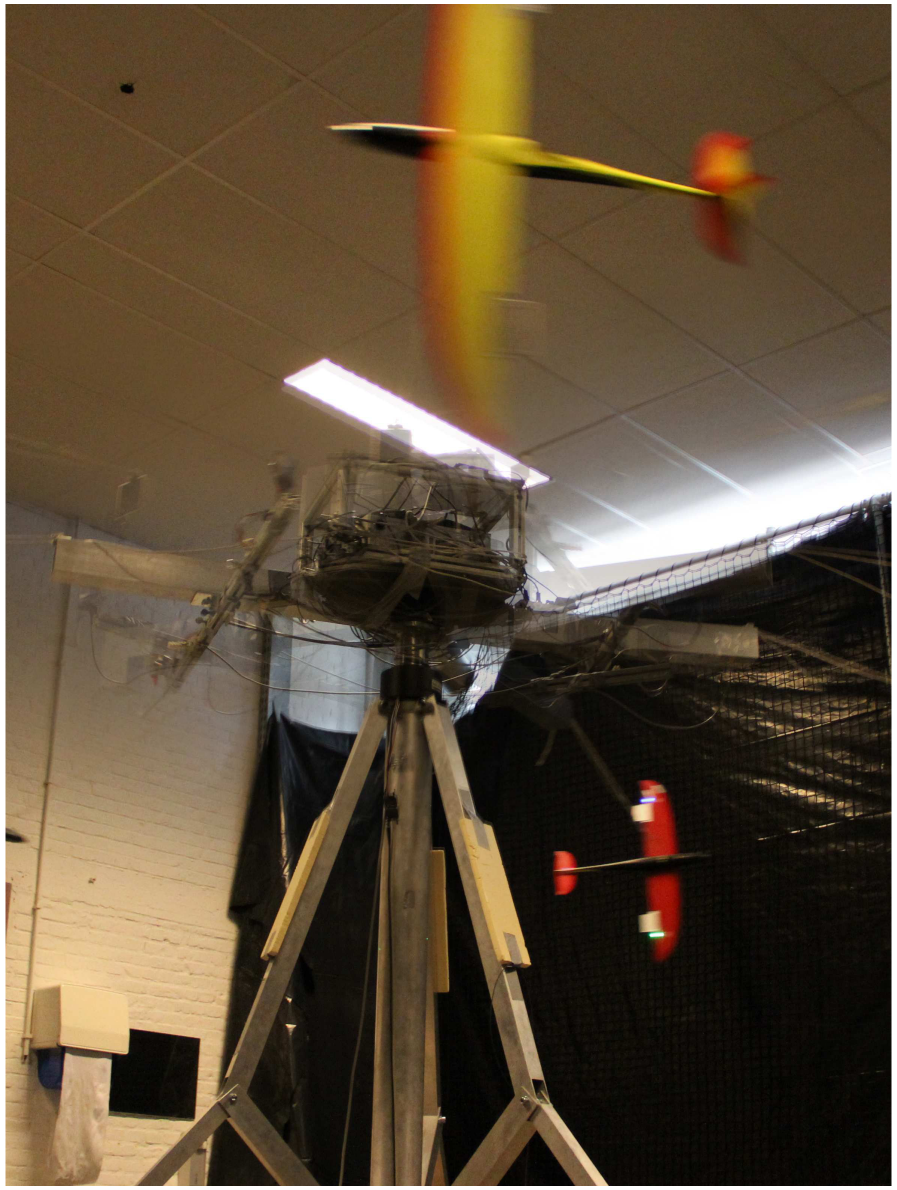



First validation of code-generated NMPC [Vukov2012]

# Estimation & Control for tethered kites

MHE and NMPC implementation on an experimental test set-up for launch/recovery of an airborne wind energy (AWE) system, located at KU Leuven.

ERC HIGHWIND



# Estimation & Control for tether kites

- Nonlinear dynamics: **27** states and **4** controls
- Nonlinear measurement functions (for camera subsystem and IMU)
- Multi-rate sensor fusion:
  - Camera measurements @ 12.5 Hz (+ images are delayed)
  - IMU measurements @ 500 Hz encoder measurements @ 10 Hz
  - Encoder measurements @ 1 kHz (snapshotting)
- MHE & NMPC update frequency: 25 Hz
- Maximum execution time for MHE: **11 ms (N = 20)**
- Maximum execution time for NMPC: **< 25 ms (N =40)**

# State estimation for induction motors* KUL & ETHZ

- Dynamic system properties:
  - 5 states, 2 controls
  - 6 estimation intervals
  - employing arrival cost
  - sampling freq.: **1.5 kHz**
- Execution times:
  - one RTI on a 3 GHz Intel CPU: **30 µs** (double precision)
  - one RTI on a 1 GHz TI low power DSP: **270 µs** (single precision)

* Frick2012

# Even more 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*
- University of Linz, Austria:
  *MHE and NMPC for diesel engine air system control*
- ABB, Switzerland:
  1) *Anti-surge control for centrifugal compressors*
  2) *MPC for torque control in power el. applications*

# Acknowledgements

EMBOCON

ERC HIGHWIND

Hans Joachim Ferreau, Boris Houska, Rien Quirynen, Joel Andersson, Janick Frasch, Alex Domahidi, Gianluca Frison

# Thank you very much for your attention!

Questions?