From state estimation to long horizon MPC for non-linear industrial applications

Andrei Tiagounov, Dennis van Hessem,
Mario Balenovic, Siep Weiland

Eindhoven University of Technology
Delft University of Technology

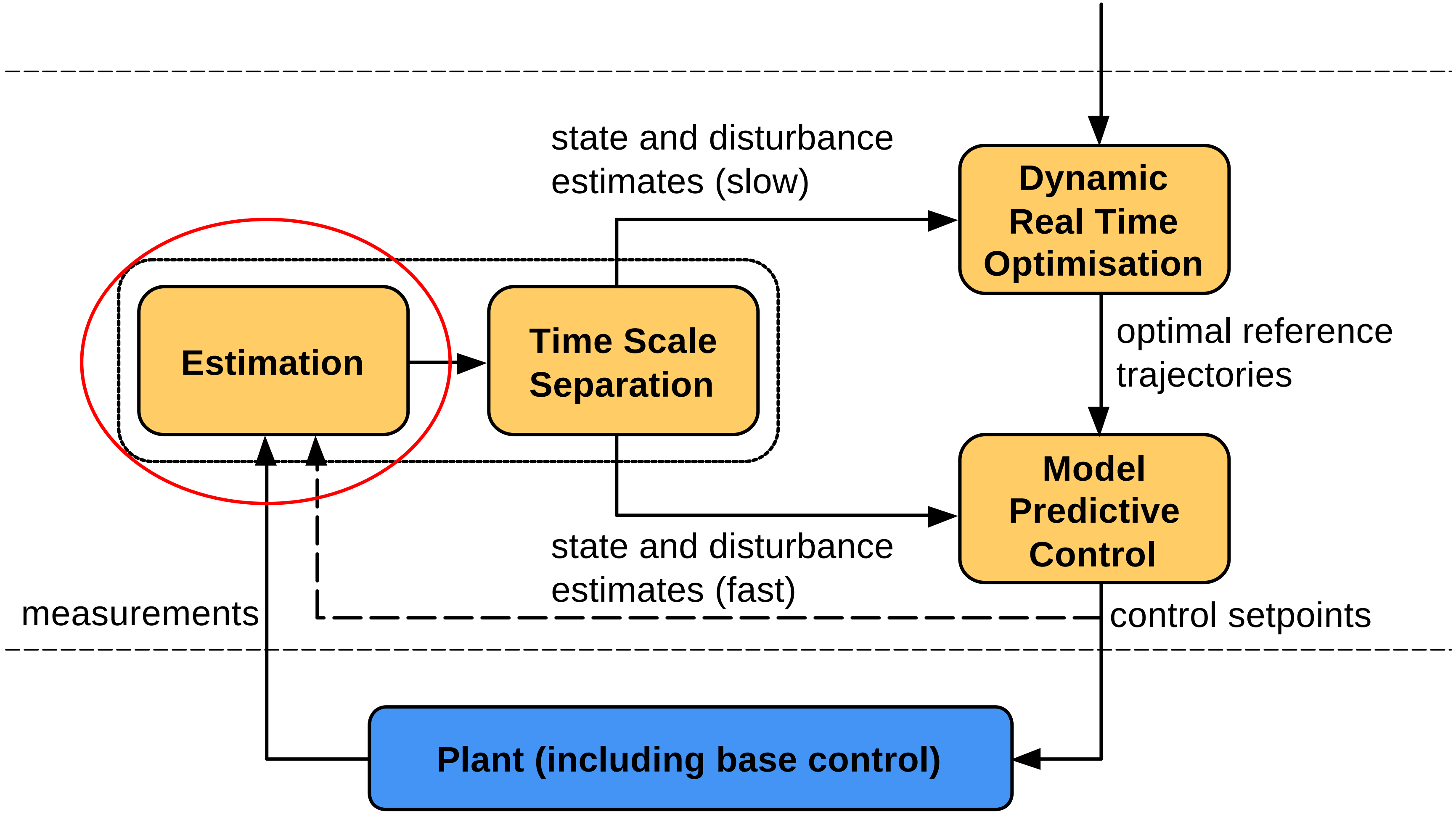
INCOOP Workshop
Düsseldorf, January 23 - 24, 2003

Objectives

- Development of techniques for data reconciliation exploiting a-priori knowledge of process behavior.
- Techniques for state reconstruction of approximate process models.
- Development of MPC techniques enabling broad bandwidth, high performance control along optimal trajectories.
- Integrated implementation of these techniques

Need control to implement optimal dynamic trajectories

- Disturbances are continuously upsetting the plant
  - actuator/sensor failures
  - set-point changes, feed fluctuations
- There always exists plant-model mismatch
  - uncertain reaction kinetics and physical properties
  - uncertain heat and mass transfer
- The initial conditions are always unknown
  - models suited for production are usually not suited for start-up simulation


Dynamic
Real Time
Optimisation
state and disturbance
estimates (slow)
Estimation
Time Scale
Separation
optimal reference
trajectories
Model
Predictive
Control
state and disturbance
estimates (fast)
measurements
control setpoints
Plant (including base control)

Contrary to linear MPC we need to initialize the model

- The input-output behavior depends on the state
  - along grade/load changes considerable change in dynamics
- The output prediction is based on simulation with the nonlinear model starting from an initial state
- Disturbances and uncertain parameters are estimated using the state estimator
  - the disturbance models are dynamic and have their own states

Set of measurements of process variables:

$$\{y^m_{k,} y^m_{k-1,} y^m_{k-2,} \ldots, y^m_{k-N+1,} y^m_{k-N,}\}$$

- Set of manipulated variables

$$\{u^m_{k,} u^m_{k-1,} u^m_{k-2,} \ldots, u^m_{k-N+1,} u^m_{k-N,}\}$$

- Dynamic model (simplified for presentation):

$$\dot{x} = f(x, u, w), \;\; x(t_0) = x_0$$

$$y = g(x, u, v)$$

Find the estimate $\hat{x}_{k|k}$ by finding function j ,(Cox,64, Lee,95):

$$\hat{x}_{k|k} = j\,(y_k^m,\ldots,y_{k-N}^m, u_{k-1},\ldots,u_{k-N}, \hat{x}_{k-N|k-N})$$

minimizing variance on estimation error:

$$J = \Delta\hat{x}_{k-N}^T P_{k-N}^{-1} \Delta\hat{x}_{k-N} + \sum_{i=k-N}^{-1} w_i^T W^{-1} w_i + \sum_{i=k-N}^{-1} v_i^T V^{-1} v_i$$

- w: process disturbance, covariance: W
- v: measurement noise, covariance: V
- $\Delta\hat{x}_{k-N}$ initial state update, covariance: P
- N length of data sets

Extended Kalman Filter

Earlier studies showed ext. Kalman filter (Lewis,86) was:

- easier to tune than horizon estimator
- not less accurate than horizon estimator
- much faster than horizon estimator
- regularizes itself in case of singular covariances

But if used, must adapt for constraints

- do regularization yourself via I/O model reduction
- need QP to find estimate

Choice to work recursively:

$$\hat{x}_{k|k} = j\,(y_k^m, u_{k-1}, \hat{x}_{k-1|k-1})$$

Make a specific linear choice for j !

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K(y_k^m - \hat{y}_{k|k-1})$$

Using linear dynamics to propagate variance:

$$\begin{pmatrix} \Delta\dot{x} \\ \Delta y \end{pmatrix} = \begin{pmatrix} A & G \\ C & F \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta w \end{pmatrix}$$

- x states
- w (all) disturbances
- y measured outputs

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} := \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial u} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial u} \end{pmatrix}$$

Compute state prediction using nonlinear model, can use any dynamic simulation tool such as GPROMS:

$$\hat{x}_{k+1|k} = \underbrace{F_{Ts}(\hat{x}_{k|k} \mid k, u_k, w_k)}_{\text{model-integration}}$$

$$= \hat{x}_{k|k} + \int_0^{Ts} f(x(t), u_k, 0)\,dt$$

And corresponding output prediction, (automatic)

$$\hat{y}_{k+1|k} = g(\hat{x}_{k+1|k}, u_k)$$

QP formulation

Define output error: $e_k := y_k^m - \hat{y}_{k|k-1}$

And arrive at QP:

$$\min_{\Delta\hat{x}_{k-1}, w_{k-1}} \frac{1}{2} \begin{pmatrix} \Delta\hat{x}_{k-1}^T \\ w_{k-1} \end{pmatrix} \begin{pmatrix} P_{k-1}^{-1} & 0 \\ 0 & W^{-1} \end{pmatrix} \begin{pmatrix} \Delta\hat{x}_{k-1} \\ w_{k-1} \end{pmatrix} + v_k^T V^{-1} v_k$$

$$v_k = e_k - C\Delta\hat{x}_k = e_k - (CA \quad C\,B) \begin{pmatrix} \Delta\hat{x}_{k-1} \\ w_{k-1} \end{pmatrix}$$

- can add arbitrary linear constraints on x,w
- must regularize P such that it has inverse
- ‘input’-’output’ Gramian based model-reduction provides one way (Moore, 76).

Unconstrained solution

Note that unconstrained case:

$$\Delta x_k = -H^{-1} g \qquad \mathrm{H} = \begin{pmatrix} P_{k-1}^{-1} + A^T C^T V^{-1} CA & A^T C^T V^{-1} CB \\ B^T C^T V^{-1} CA & W^{-1} + B^T C^T V^{-1} CB \end{pmatrix}$$

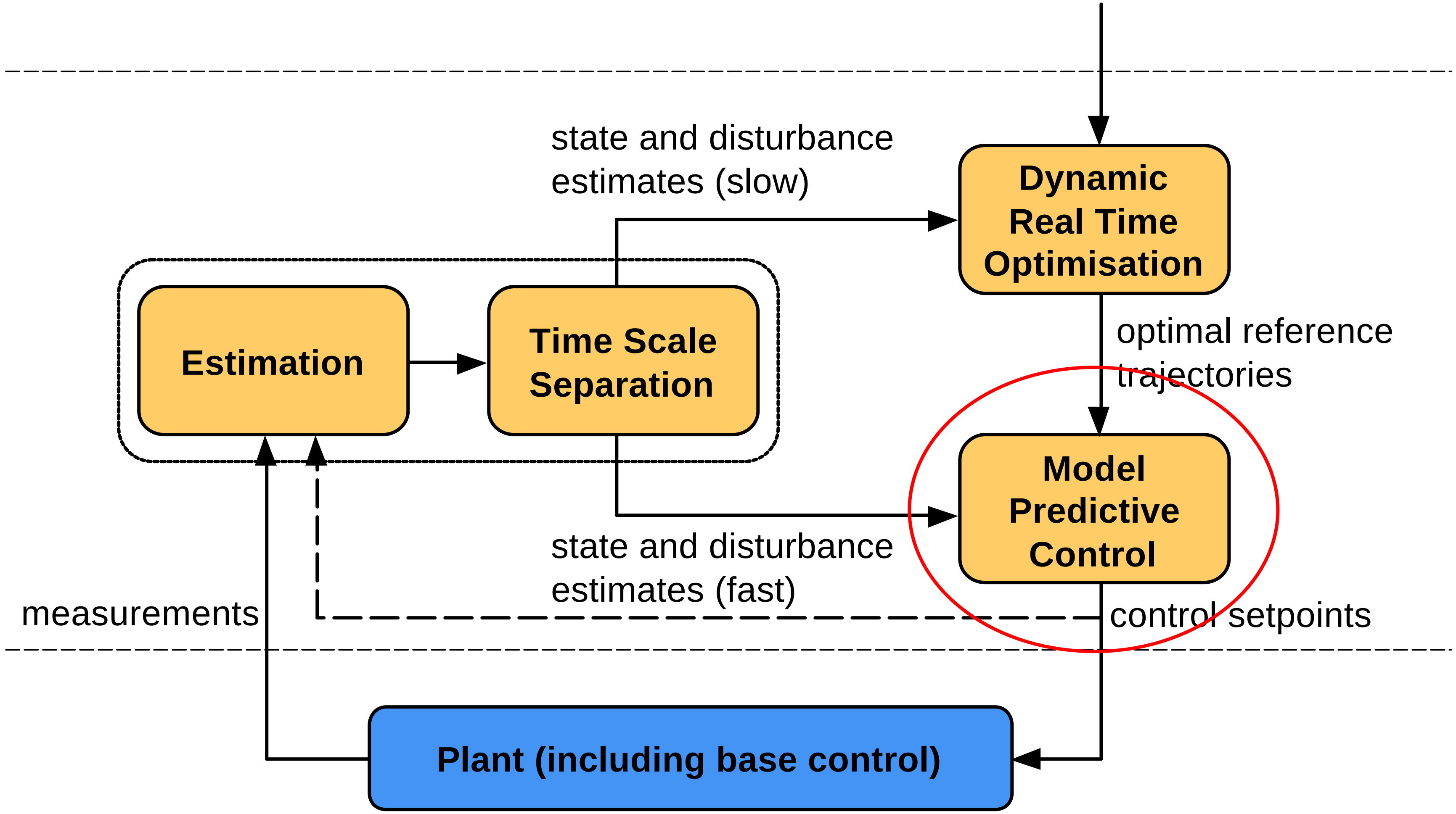
$$g = -A^T C^T V^{-1} \boldsymbol{e}_k$$

compares to familiar Riccati solution:

$$\Delta x_k = K\boldsymbol{e}_k \qquad \begin{aligned} P &= APA^T - APC^T (CPC' + V)^{-1} CPA + W \\ K &= PC^T (CPC^T + V)^{-1} \end{aligned}$$

Note: here only $CPC^T + V$ needs to be invertible!

MPC in the INCOOP project



The problem of finding a control sequence for a continuous time plant

$$\dot{x} = f(x,u), x(t_0) = \hat{x}_0$$

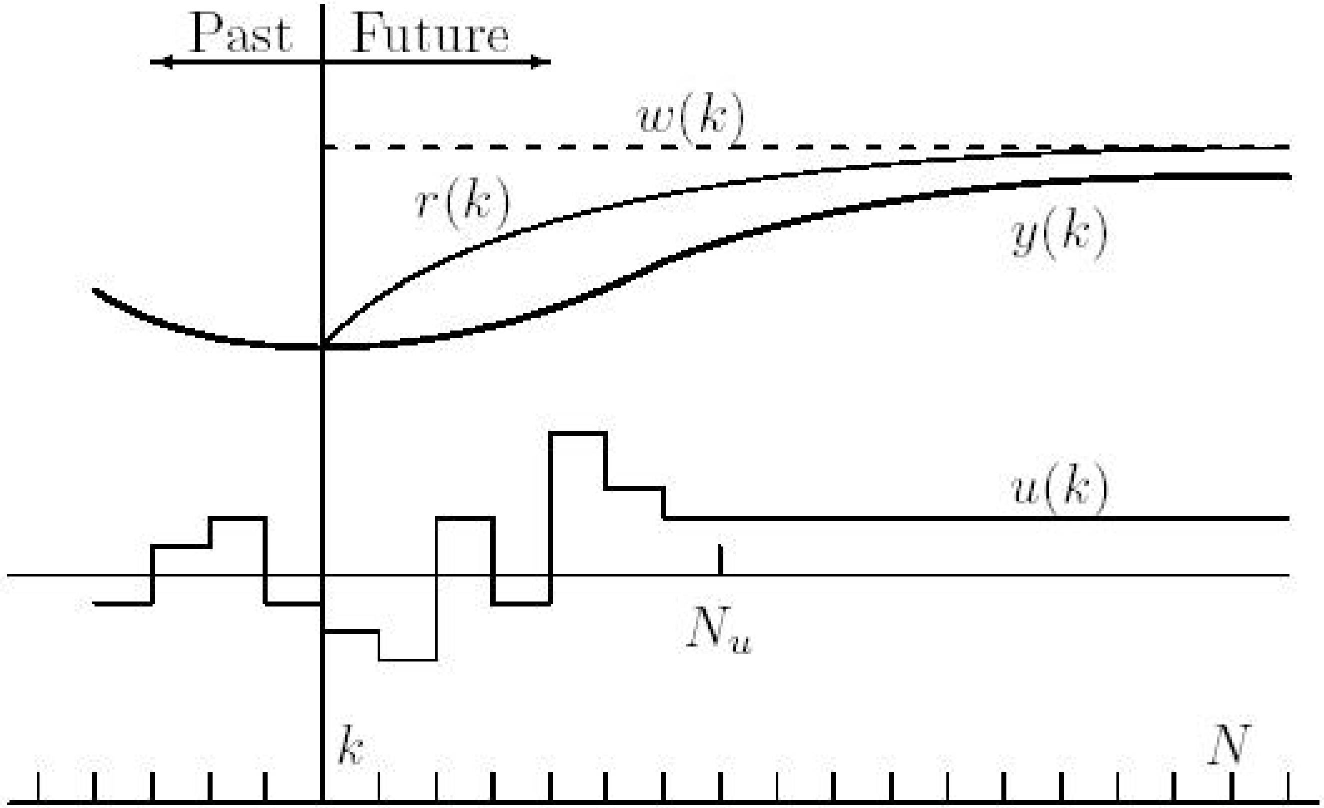
$$y = g(x,u)$$

Minimizing some continuous time control objective:

$$J(u) = x(t_f)^T P x(t_f) + \int_{t_0}^{t_f} l(y(t),u(t))dt$$

One can go many ways !

MPC concept



Typical choices

A full NMPC problem generally much too much time consuming to enable high performance.

Desire small sample time and long prediction horizon!

Approach:

- Discretize continuous time objective (trapezoidal rule)
- Use local dynamics to approx. sensitivity functions: Linear Time Varying (LTV) control
- Very reliable for proper choice of sample time
- Much faster for many of optimization variables
- After a few iterations you get `SQP' behaviour

- Integrate nonlinear model along previous input sequence: gives output and state predictions:

$$\{y_k^{pred}, y_{k+1}^{pred}, \ldots, y_{k+N}^{pred}\}, \{x_k^{pred}, x_{k+1}^{pred}, \ldots, x_{k+N}^{pred}\}$$

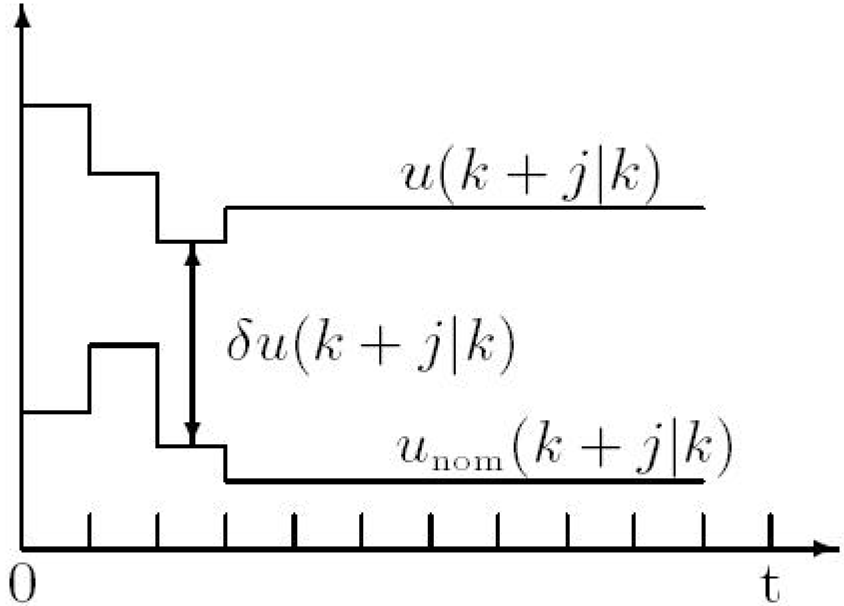
- Derive linear time varying (LTV) model along this trajectory (time-discretize local dynamics)

$$\delta y = G_U \delta u, \ \delta u = (\delta u_k, \ldots, \delta u_{k+N-1})^T$$

$$G_U = \begin{pmatrix} D_k & 0 & \ldots & \ldots \\ C_{k+1}B_k & D_{k+1} & 0 & \ldots \\ C_{k+2}A_{k+1}B_k & C_{k+2}B_{k+1} & D_{k+2} & 0 \\ C_{\ldots}A_{\ldots}A_{\ldots}B_{\ldots} & C_{\ldots}A_{\ldots}B_{\ldots} & C_{..}B_{..} & \ldots \end{pmatrix}$$

Optimizing control inputs:

$$\delta u_{k+j|k} = u_{k+j|k} - u^{nom}_{k+j|k}$$



An effective choice for the nominal control input:

$$u^{nom}_{k+j|k} = u^{opt}_{k+j|k-1}$$

•LTV-MPC problem amounts to find an optimal control sequence $\{u(k+j)\}_{j=1}^{N-1}$ minimizing the objective function:

$$J_k = x_{k+N}^T P x_{k+N} + \sum_{j=1}^{N} \begin{pmatrix} y_{k+j}^{pred} - y_{k+j}^{ref} \\ u_{k+j}^{pred} - u_{k+j-1}^{pred} \end{pmatrix}^T \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \begin{pmatrix} y_{k+j}^{pred} - y_{k+j}^{ref} \\ u_{k+j}^{pred} - u_{k+j-1}^{pred} \end{pmatrix}^T$$

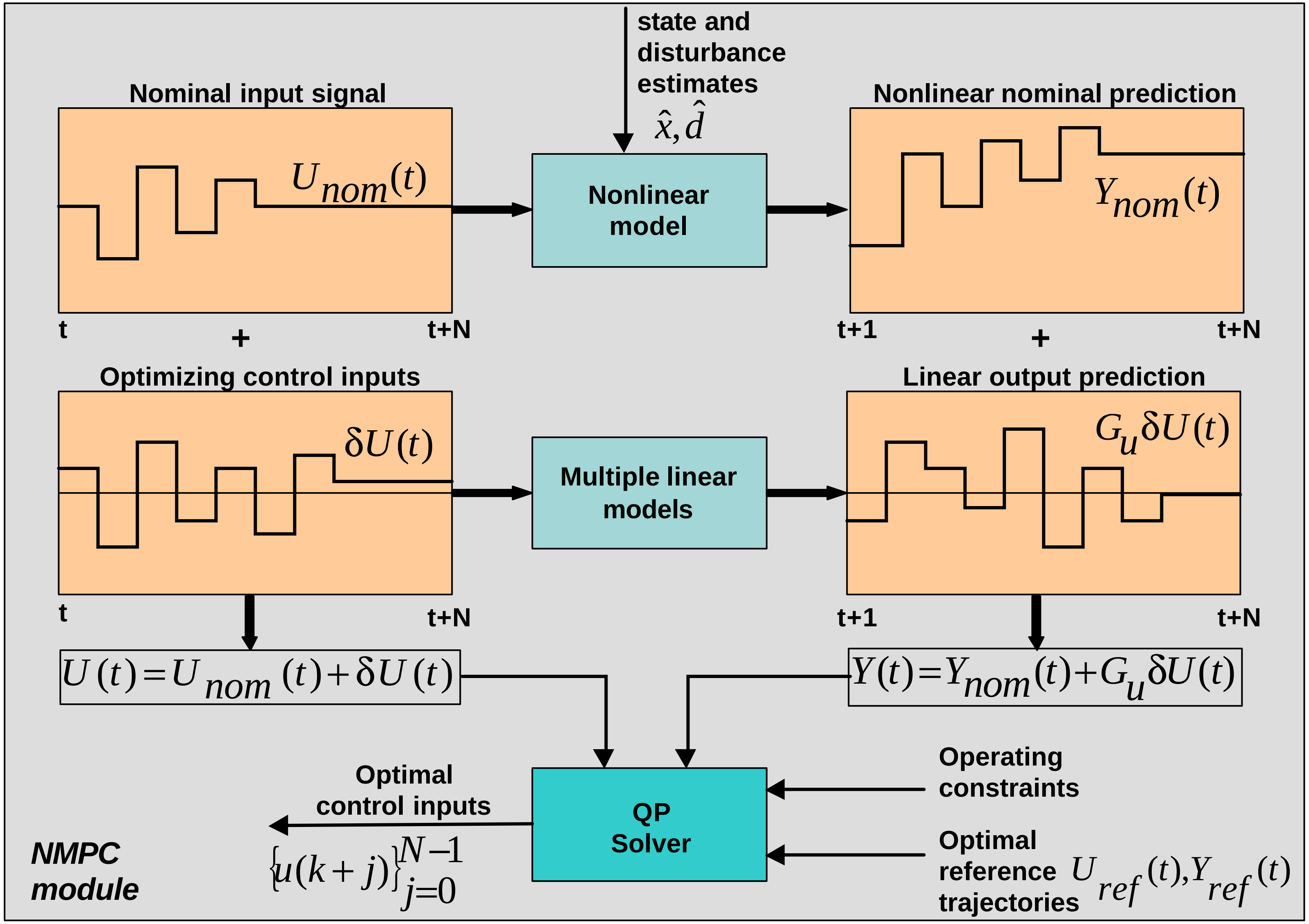
subject to constraints:

$$y_{\min} \le y(k+j) \le y_{\max}$$
$$u_{\min} \le u(k+j) \le u_{\max}$$
$$\Delta u_{\min} \le \Delta u(k+j) \le \Delta u_{\max}$$
$$j = 0, \ldots, N$$

•Then solve resulting QP and add solution to previous control sequence:

$$u^{pred}_{k+j} = u^{nom}_{k+j} + \delta u_{k+j}$$

Nonlinear MPC



Standard QP solvers:

State variables are eliminated

•Active Set Methods (ASM)

•Interior-Point Methods (IPM): Mosek, etc.

Computational time increases with the 3rd power of the number of variables Nn_u

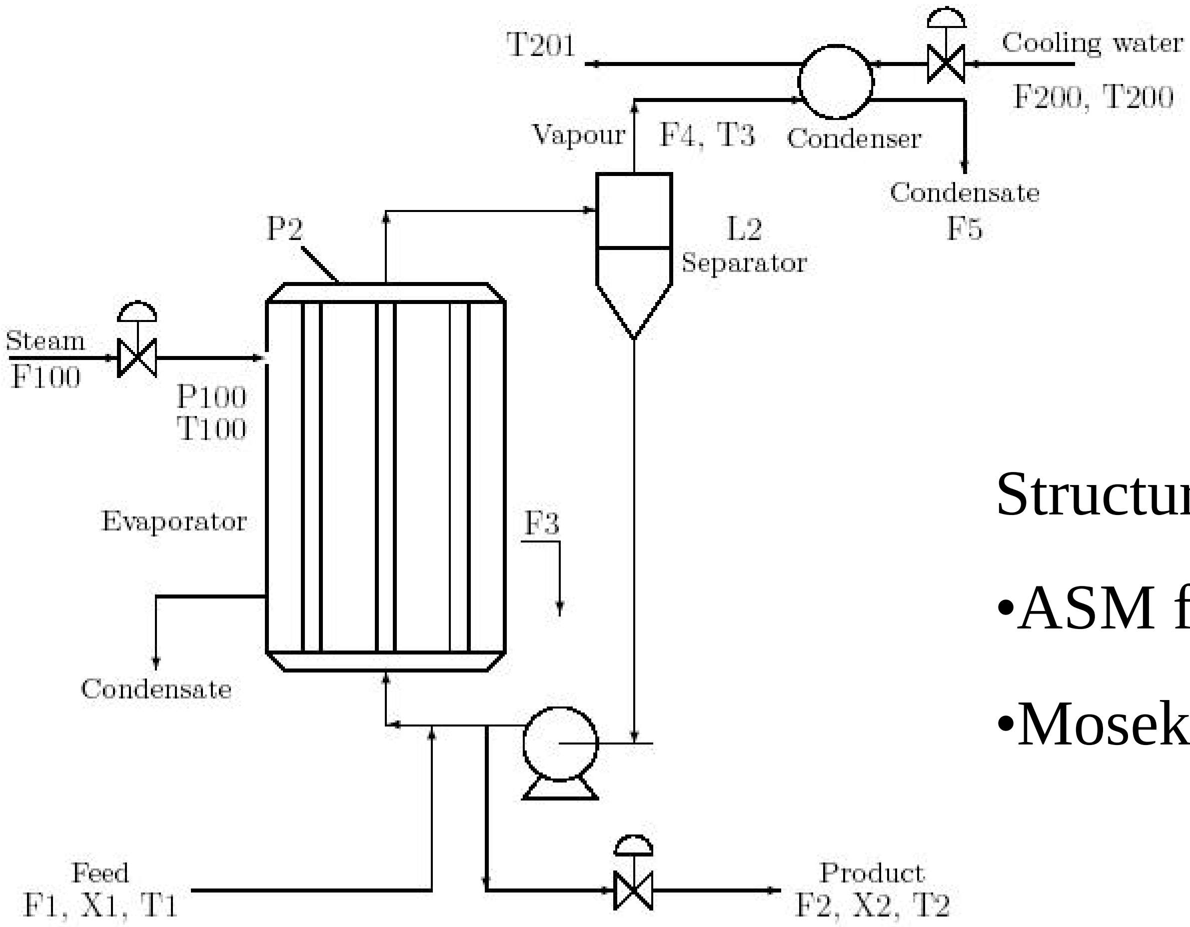
Structured IPM:

State variables are *not* eliminated

Computational time increases linearly with the number of variables

-> **Allows long horizon prediction/ large bandwidth**

Example: Evaporation process



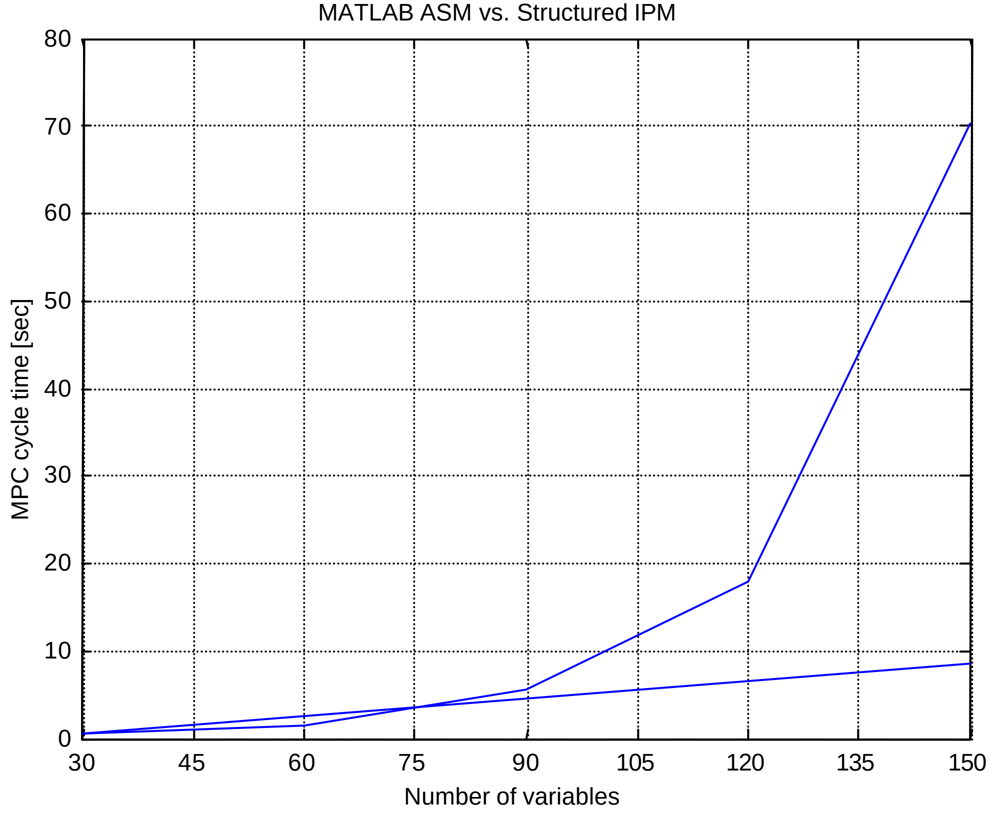
3 MVs:

F_2, P_{100}, F_{200}

Structured IPM is faster than

- ASM for N > 25
- Mosek for N > 160

Evaporation process results



Example 2: Distillation process

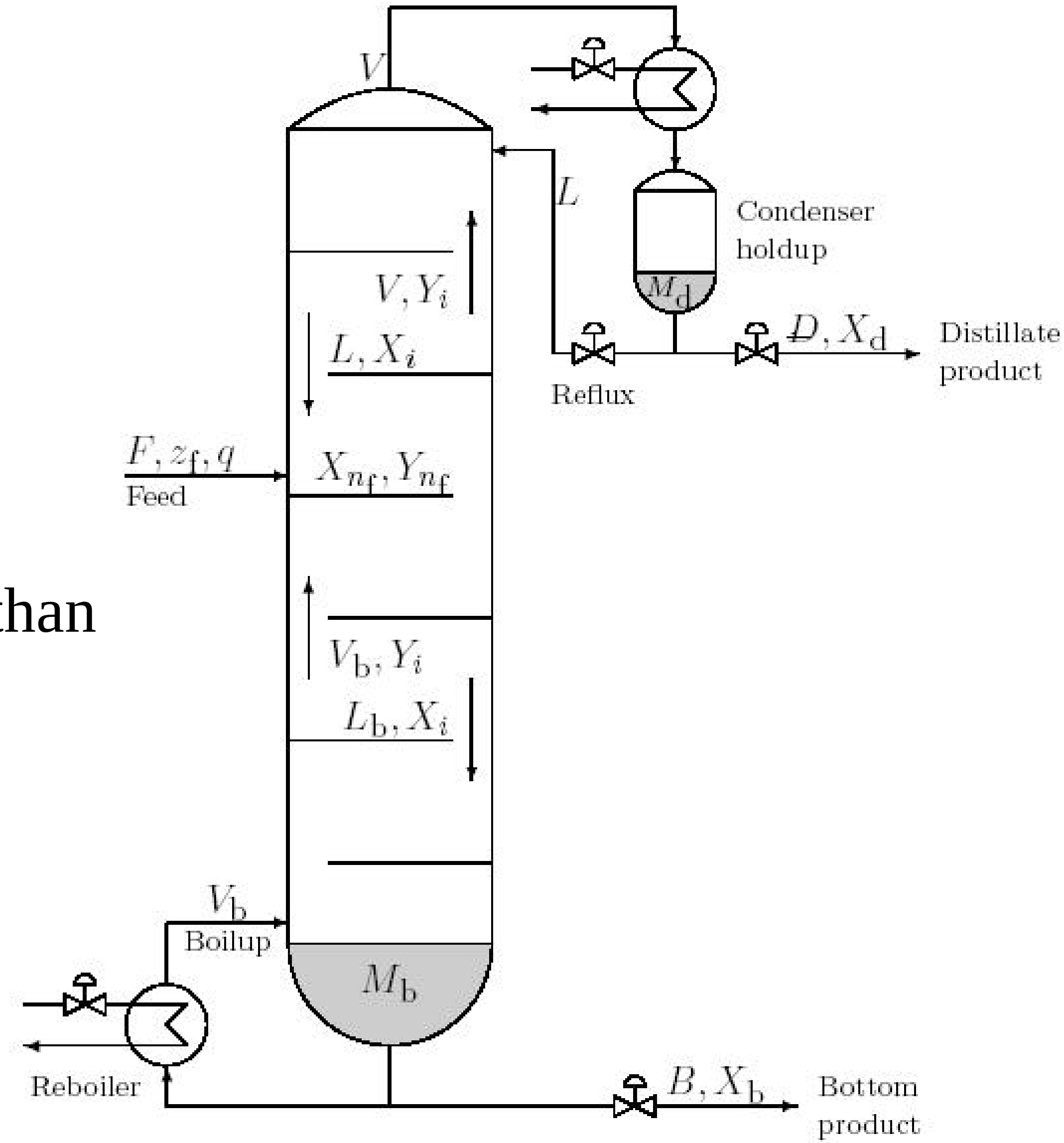
2 MVs:

Reflux L

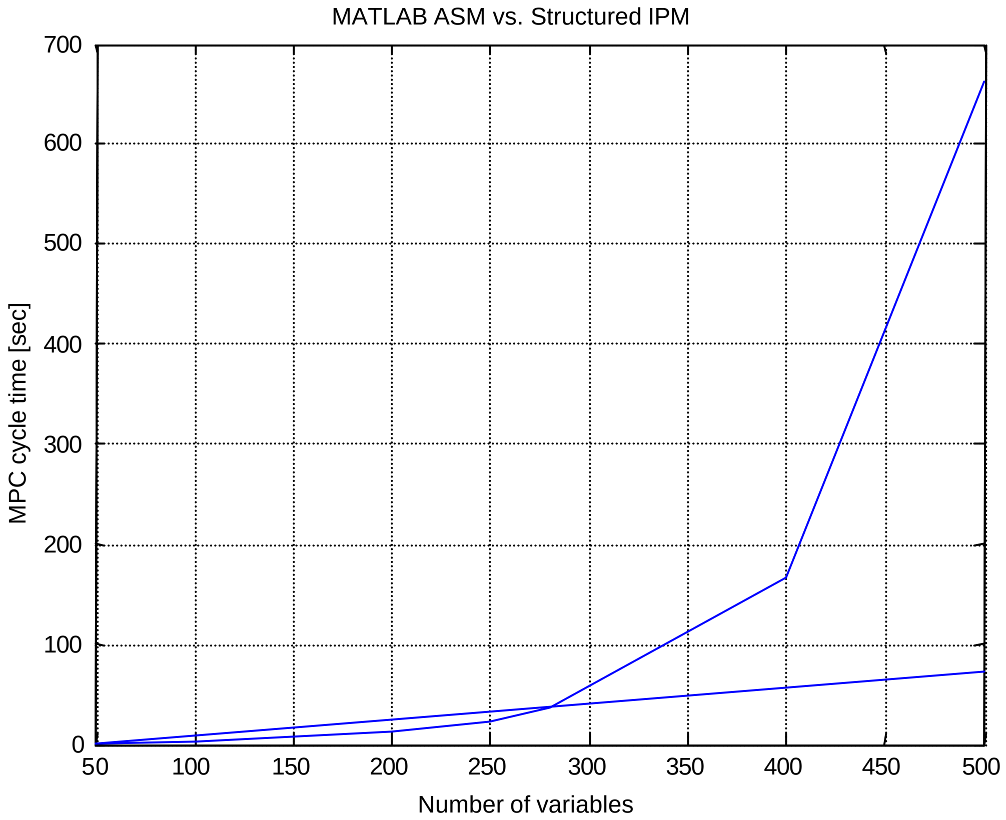
Boilup V_b

Structured IPM is faster than

ASM for N > 140



Distillation process results



Conclusions

- Estimator+MPC have been implemented within INCOOP software architecture.
- Tested and operative for both Bayer and Shell process.
- For large scale problems QP computation time is no longer a bottleneck. Model simulation (MPC prediction) is the main computational burden in the environment.