Embedded Model Predictive Control on a PLC Using a Primal-Dual First-Order Method

D. K. M. Kufoalor (Giorgio)[1], S. Richter[2], L. Imsland[1], T. A. Johansen[1], M. Morari[2], and G. O. Eikrem[3]

[1]Dept. of Engineering Cybernetics, NTNU Trondheim
[2] Automatic Control Laboratory ETH Zurich,
[3]Statoil ASA, Rotvoll Norway

March 19, 2014

Outline

Outline

Emb-Opt: Embedded MPC From Industrial PC-based MPC

Key aspects of the Emb-Opt project:

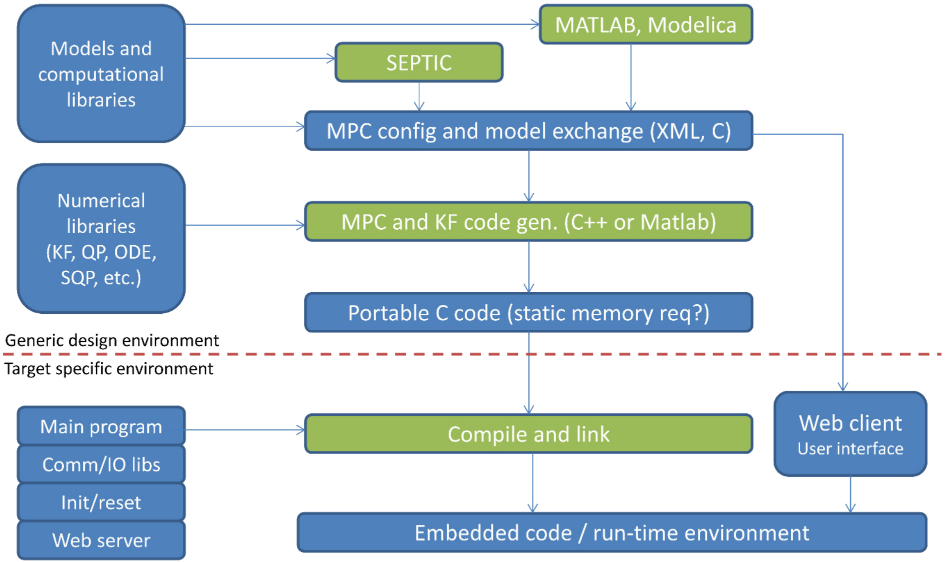
- explore industrially proven MPC packages as design tools for embedded use
  - ✓ Constraints and feasibility:
    hard and soft constraints, priority hierarchy
  - ✓ Tuning possibilities:
    scaling that captures acceptable variation/span of variables
  - ✓ Problem reduction:
    move blocking, evaluation points

Emb-Opt: Embedded MPC From Industrial PC-based MPC

Key aspects of the Emb-Opt project:

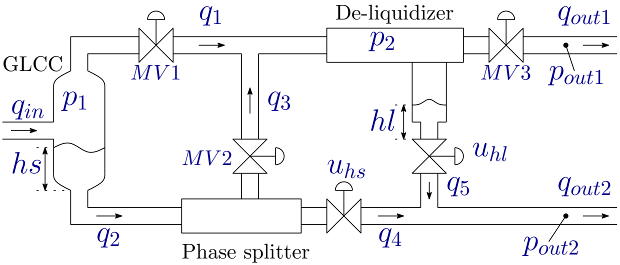
- ◇ explore industrially proven MPC packages as design tools for embedded use
  - ✓ Constraints and feasibility:
    hard and soft constraints, priority hierarchy
  - ✓ Tuning possibilities:
    scaling that captures acceptable variation/span of variables
  - ✓ Problem reduction:
    move blocking, evaluation points
- ◇ Automatic code generation
- ◇ Incorporation of custom high-speed solvers
- ◇ Real-time guarantee considerations
- ◇ Efficient use of limited computational resources

Emb-Opt: Embedded MPC Open Source Architecture

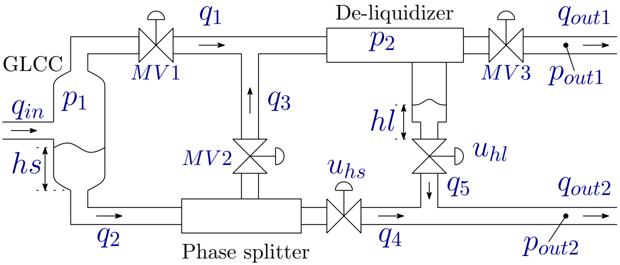


Outline

Prototype of Statoil's Subsea Compact Separation Process



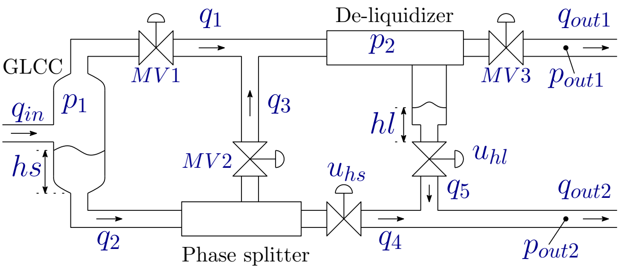
Prototype of Statoil's Subsea Compact Separation Process



Control Objectives*:

- Control gas volume fraction (GVF) in liquid and gas outlets.
- Control pressure p_1 and p_2, and the pressure difference.
- Keep process within operational limits on pressure, and GVF in outlets.
- Respect physical open limits on valves.

Prototype of Statoil's Subsea Compact Separation Process



Control Objectives*:

- Control gas volume fraction (GVF) in liquid and gas outlets.
- Control pressure p_1 and p_2, and the pressure difference.
- Keep process within operational limits on pressure, and GVF in outlets.
- Respect physical open limits on valves.

* Performance for worst case inlet flow scenario required!

The Target Platform: AC500 PM592-ETH PLC

- Freescale MPC603e RISC CPU: 500MIPS, 250MFLOPS @ 200 MHz
- FPU: fully IEEE 754-compliant for both single- and double-precision
- 4 MB RAM *program memory*, 4 MB integrated *data memory*

The Target Platform: AC500 PM592-ETH PLC

- Freescale MPC603e RISC CPU: 500MIPS, 250MFLOPS @ 200 MHz
- FPU: fully IEEE 754-compliant for both single- and double-precision
- 4 MB RAM *program memory*, 4 MB integrated *data memory*
- C-code in a PLC Software architecture and runtime environment
- C-Code application is a part of IEC 61131-3 application
- ANSI C89, C99 with a restricted set of standard library functions
- GNU GCC 4.7.0 compiler toolchain, external library linkage not allowed
- Certified proprietary hardware/software

Outline

The MPC Problem (Corresponding to design in SEPTIC)

$$\min \sum_{k=k_0+H_w}^{k_0+H_p} [y_e(k)^T \bar{Q}_y y_e(k)] + \rho_h^T \epsilon_h + \rho_l^T \epsilon_l + \sum_{k=k_0}^{k_0+H_u} [u_e(k)^T \bar{Q}_u u_e(k) + \Delta u(k)^T \bar{P} \Delta u(k)] \quad (1)$$

subject to

$$\begin{aligned}
&\underline{y} - \epsilon_l \leqslant y(k) \leqslant \bar{y} + \epsilon_h, && k \in [k_0 + H_w, \ldots, k_0 + H_p],\\
&\epsilon_h \geqslant 0,\ \epsilon_l \geqslant 0, && \\
&y(k) = y(k|k_0), \text{ e.g. step resp. model} && k \in [k_0 + H_w, \ldots, k_0 + H_p],\\
&\underline{u} \leqslant u(k) \leqslant \bar{u}, && k \in [k_0, \ldots, k_0 + H_u],\\
&\underline{\Delta u} \leqslant \Delta u(k) \leqslant \overline{\Delta u}, && k \in [k_0, \ldots, k_0 + H_u],\\
&u(k) = u(k-1) + \Delta u(k), && k \in [k_0, \ldots, k_0 + H_u],
\end{aligned}$$

The QP Problem (Formulated in MPC code generator)

$$\min \; Y(k)^T Q_y Y(k) + U(k)^T Q_u U(k) + \Delta U(k)^T P \Delta U(k) - \mathcal{V}(k)^T Q_u U(k) - \mathcal{T}(k)^T Q_y Y(k) + \rho_h^T \epsilon_h + \rho_l^T \epsilon_l \tag{2}$$

subject to

$$\begin{aligned}
\Omega Y(k) &\leqslant \omega + M_h \epsilon_h + M_l \epsilon_l, \quad \epsilon_h \geqslant 0, \; \epsilon_l \geqslant 0, \\
Y(k) &= \Psi \Delta \bar{U}(k-1) + \Upsilon u(k-N) + \Theta \Delta U(k) + \mathbf{1}\nu(k), \\
E \Delta U(k) &\leqslant e, \\
F U(k) &\leqslant f, \\
\kappa U(k) &= \tau u(k-1) + \Delta U(k),
\end{aligned}$$

According to the notations of *Maciejowski, (2002)*

Standard QP

The variables of (2) can be grouped together to form the decision vector

$$x(k) = \begin{bmatrix}\Delta U(k)^T & U(k)^T & Y(k)^T & \epsilon_h^T & \epsilon_l^T\end{bmatrix}^T,$$

leading to the problem

$$\min\left\{\tfrac{1}{2}x(k)^T H x(k) + g(k)^T x(k) \mid \bar{A}_i\, x(k) \leqslant \bar{b}_i,\ A_e\, x(k) = b_e\right\}, \qquad (3)$$

where

$$\begin{aligned}
H &= 2 \cdot \text{blkdiag}(P,\ Q_u,\ Q_y,\ 0,\ 0),\\
g(k) &= \begin{bmatrix}0 & -\mathcal{V}(k)^T Q_u & -\mathcal{T}(k)^T Q_y & \rho_h^T & \rho_l^T\end{bmatrix}^T,\\
\bar{A}_i &= \begin{bmatrix} E & 0 & 0 & 0 & 0\\ 0 & F & 0 & 0 & 0\\ 0 & 0 & \Omega & -M_h & -M_l\\ 0 & 0 & 0 & -I & 0\\ 0 & 0 & 0 & 0 & -I \end{bmatrix}, \quad \bar{b}_i = \begin{bmatrix} e\\ f\\ \omega\\ 0\\ 0\end{bmatrix},\\
A_e &= \begin{bmatrix} -I & \kappa & 0 & 0 & 0\\ -\Theta & 0 & I & 0 & 0\end{bmatrix},\\
b_e &= \begin{bmatrix} \tau u(k-1)\\ \Psi\Delta\bar{U}(k-1) + \Upsilon u(k-N) + \mathbf{1}\nu(k)\end{bmatrix}.
\end{aligned}$$

A Primal-Dual First-Order QP Solver

$$\min\left\{\tfrac{1}{2}x^T H x + g^T x \;\middle|\; x \in \mathbb{X},\ A_i x \leqslant b_i,\ A_e x = b_e\right\}, \tag{4}$$

$x \in \mathbb{R}^n$, $b_e \in \mathbb{R}^{m_e}$ (updated online), and $H \in \mathbb{R}^{n \times n}$ is positive semi-definite.
$\mathbb{X}$: bounds on $\Delta U(k)$, $U(k)$, ϵ_h, ϵ_l, for which the *projection operator*

$$\pi_{\mathbb{X}}(\hat{x}) = \arg\min_{x \in \mathbb{X}} \tfrac{1}{2}\|x - \hat{x}\|^2$$

is cheap to compute. Note: Not for $Y(k)$ constraints (>30k regions in explicit solution).

A Primal-Dual First-Order QP Solver

$$\min\left\{\tfrac{1}{2}x^THx + g^Tx \;\middle|\; x \in \mathbb{X},\;\; A_i\,x \leqslant b_i,\;\; A_e\,x = b_e\right\}, \tag{4}$$

$x \in \mathbb{R}^n$, $b_e \in \mathbb{R}^{m_e}$ (updated online), and $H \in \mathbb{R}^{n\times n}$ is positive semi-definite.
$\mathbb{X}$: bounds on $\Delta U(k)$, $U(k)$, ϵ_h, ϵ_l, for which the *projection operator*

$$\pi_{\mathbb{X}}\left(\hat{x}\right) = \arg\min_{x\in\mathbb{X}} \tfrac{1}{2}\left\|x - \hat{x}\right\|^2$$

is cheap to compute. Note: Not for $Y(k)$ constraints (>30k regions in explicit solution).
Transform (4) into

$$\min_{x}\max_{\lambda} x^TA^T\lambda + \gamma(x) - \theta^*(\lambda)$$

γ and θ^* are convex and closed functions, and their *preconditioned proximity operators*, e.g.

$$\text{prox}_{\gamma}\left(\hat{x}\right) = \arg\min_{x} \gamma(x) + \tfrac{1}{2}\left\|x - \hat{x}\right\|^2_{T^{-1}},$$

$T \succ 0 \in \mathbb{R}^{n\times n}$, can be evaluated in closed-form or computed efficiently.

A Primal-Dual First-Order QP Solver

$$\min\left\{\tfrac{1}{2}x^T H x + g^T x \;\middle|\; x \in \mathbb{X},\; A_i x \leqslant b_i,\; A_e x = b_e\right\}, \tag{4}$$

$x \in \mathbb{R}^n$, $b_e \in \mathbb{R}^{m_e}$ (updated online), and $H \in \mathbb{R}^{n\times n}$ is positive semi-definite.
$\mathbb{X}$: bounds on $\Delta U(k)$, $U(k)$, ϵ_h, ϵ_l, for which the *projection operator*

$$\pi_{\mathbb{X}}(\hat{x}) = \arg\min_{x\in\mathbb{X}} \tfrac{1}{2}\|x - \hat{x}\|^2$$

is cheap to compute. Note: Not for $Y(k)$ constraints (>30k regions in explicit solution).

Transform (4) into

$$\min_x \max_\lambda x^T A^T \lambda + \gamma(x) - \theta^*(\lambda)$$

γ and θ^* are convex and closed functions, and their *preconditioned proximity operators*, e.g.

$$\mathrm{prox}_\gamma(\hat{x}) = \arg\min_x \gamma(x) + \tfrac{1}{2}\|x - \hat{x}\|^2_{T^{-1}},$$

$T \succ 0 \in \mathbb{R}^{n\times n}$, can be evaluated in closed-form or computed efficiently.

$$A = \begin{bmatrix} A_e \\ A_i \end{bmatrix},\; b = \begin{bmatrix} b_e \\ b_i \end{bmatrix},\; \theta^*(\lambda) = \lambda^T b + \iota_\Lambda(\lambda),\; \gamma(x) = \tfrac{1}{2}x^T H x + g^T x + \iota_{\mathbb{X}}(x)$$

$\iota(\cdot)$ is the indicator function of the corresponding set, e.g.

$$\iota_\Lambda(\lambda) = \begin{cases} 0 & \text{if } \lambda \in \left\{(\lambda_e, \lambda_i) \in \mathbb{R}^{m_e} \times \mathbb{R}^{m_i} \,\middle|\, \lambda_i \geqslant 0\right\}, \\ +\infty & \text{otherwise.} \end{cases}$$

A Primal-Dual First-Order QP Solver

Algorithm 1 Preconditioned primal-dual first-order method

Require: $\lambda_0 \in \mathbb{R}^m$, $x_0 \in \mathbb{R}^n$ and $\bar{x}_0 = x_0$; A, b
preconditioner matrices $T \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{m \times m}$

1: **loop**
2: $\quad \lambda_{i+1} = \pi_\Lambda \left(\lambda_i + \Sigma(A\bar{x}_i - b)\right)$
3: $\quad x_{i+1} = \arg\min_{x \in \mathbb{X}} \frac{1}{2} x^T H x + g^T x + \frac{1}{2} \left\| x - (x_i - T A^T \lambda_{i+1}) \right\|_{T^{-1}}^2$
4: $\quad \bar{x}_{i+1} = 2x_{i+1} - x_i$
5: **end loop**

Originally developed for imaging applications (T. Pock and A. Chambolle, 2011)

A Primal-Dual First-Order QP Solver

Algorithm 1 Preconditioned primal-dual first-order method

Require: $\lambda_0 \in \mathbb{R}^m$, $x_0 \in \mathbb{R}^n$ and $\bar{x}_0 = x_0$; A, b preconditioner matrices $T \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{m \times m}$

1: **loop**

2: $\lambda_{i+1} = \pi_\Lambda \left(\lambda_i + \Sigma(A\bar{x}_i - b) \right)$

3: $x_{i+1} = \arg\min_{x \in \mathbb{X}} \frac{1}{2} x^T H x + g^T x + \frac{1}{2} \left\| x - (x_i - TA^T \lambda_{i+1}) \right\|_{T^{-1}}^2$

4: $\bar{x}_{i+1} = 2x_{i+1} - x_i$

5: **end loop**

Originally developed for imaging applications (T. Pock and A. Chambolle, 2011)

$\{x_i\}$, $\{\lambda_i\}$ converge to (x^*, λ^*) of (4) if the preconditioner matrices are chosen as the diagonal matrices $\Sigma = \alpha \cdot \mathrm{diag}(\sigma)$, $\alpha \in (0, 1)$, and $T = \mathrm{diag}(\tau)$ where

$$\sigma_i = \frac{1}{\sum_{j=1}^{n} |A_{ij}|},\ i = 1 \dots m, \text{ and } \tau_j = \frac{1}{\sum_{i=1}^{m} |A_{ij}|},\ j = 1 \dots n. \qquad (5)$$

A Primal-Dual First-Order QP Solver

Algorithm 1 Preconditioned primal-dual first-order method

Require: $\lambda_0 \in \mathbb{R}^m$, $x_0 \in \mathbb{R}^n$ and $\bar{x}_0 = x_0$; A, b
preconditioner matrices $T \in \mathbb{R}^{n\times n}$, $\Sigma \in \mathbb{R}^{m\times m}$

1: **loop**
2: $\lambda_{i+1} = \pi_\Lambda\left(\lambda_i + \Sigma(A\bar{x}_i - b)\right)$
3: $x_{i+1} = \arg\min_{x\in\mathbb{X}} \frac{1}{2}x^THx + g^Tx + \frac{1}{2}\left\|x - (x_i - TA^T\lambda_{i+1})\right\|^2_{T^{-1}}$
4: $\bar{x}_{i+1} = 2x_{i+1} - x_i$
5: **end loop**

Originally developed for imaging applications (T. Pock and A. Chambolle, 2011)

$\{x_i\}$, $\{\lambda_i\}$ converge to (x^*, λ^*) of (4) if the preconditioner matrices are chosen as the diagonal matrices $\Sigma = \alpha \cdot \mathrm{diag}(\sigma)$, $\alpha \in (0,1)$, and $T = \mathrm{diag}(\tau)$ where

$$\sigma_i = \frac{1}{\sum_{j=1}^{n}|A_{ij}|},\ i = 1\ldots m, \text{ and } \tau_j = \frac{1}{\sum_{i=1}^{m}|A_{ij}|},\ j = 1\ldots n. \tag{5}$$

– ensures that the convergence criterion

$$\|\Sigma^{\frac{1}{2}} A T^{\frac{1}{2}}\|^2 < 1 \tag{6}$$

is satisfied.

A Primal-Dual First-Order QP Solver

Remark:
Line 3 in Algorithm 1 is a crucial step:

Good performance can be expected only if the minimization problem in x can be solved efficiently.

Note:
Set $\mathbb{X}$ contains upper/lower bounds on some components of x only, whereas the Hessian H is diagonal.

Since the preconditioner matrix T is diagonal positive definite, the minimizer in line 3 is

$$x_{i+1} = \pi_{\mathbb{X}} \left(-\left(H + T^{-1}\right)^{-1} \left(g - T^{-1}(x_i - TA^T \lambda_{i+1})\right) \right),$$

where the projection on set $\mathbb{X}$ is a component-wise saturation.

The MPC Main Loop (Part of MPC C code framework)

```
while(1){
    // read measurements from plant
    readMeas();
    // calculate unmeasured disturbance
    calcUnmeasuredDisturbance();
    // solve the QP problem
    warmStart(); solve();
    //Send the optimal MVs to plant
    sendData();
    // Make a time shift
    shiftTime();
}
```

The MPC Main Loop (Part of MPC C code framework)

```c
while(1){
    // read measurements from plant
    readMeas();
    // calculate unmeasured disturbance
    calcUnmeasuredDisturbance();
    // solve the QP problem
    warmStart(); solve();
    //Send the optimal MVs to plant
    sendData();
    // Make a time shift
    shiftTime();
}
```

FiOrdOs was used to generate a library-free QP solver C code that implements the proposed first-order primal-dual method

Outline

MPC Problem Size and Real-time Specification

- 4 controlled variables (CVs) with up to 10 evaluation points each
- 3 manipulated variables (MVs), each with 6 blocking indices
- 2 measured fast-changing process disturbances (DVs)
- 6 slack variables
- matrix sizes: $Q_y \in \mathbb{R}^{40\times40}$, $Q_u \in \mathbb{R}^{18\times18}$, $P \in \mathbb{R}^{18\times18}$, $\rho \in \mathbb{R}^{1\times6}$, $E \in \mathbb{R}^{36\times18}$, $\Omega \in \mathbb{R}^{60\times40}$, $F \in \mathbb{R}^{36\times18}$, $\kappa \in \mathbb{R}^{18\times18}$, $\tau \in \mathbb{R}^{18\times3}$, $\Psi \in \mathbb{R}^{40\times445}$, $\Upsilon \in \mathbb{R}^{40\times6}$, and $\Theta \in \mathbb{R}^{40\times18}$.
- 58 equality constraints, 138 inequality constraints, and 82 decision variables
- A sampling frequency of 1 Hz was used, requiring real-time computational time much less than a second.

Embedded MPC Performance on AC500 PM592-ETH PLC

— Hydrodynamic Slugging Case

Table : Real-time closed-loop results on the PLC for 600 time steps of the subsea compact separation process. Abbreviation *sp* denotes single precision floating point.

QP Solver	**Time** (ms) avg./max	**Iterations** avg./max	**Mean Square Error** CV1/CV2/CV3/CV4	**C / PLC** Code (MB)
1: IP-cold	72.2/84.9	15/18	0.04/0.008/2.68/0.31	0.96/2.16
2: IP-cold,sp	63.8/65.4	18/18	0.04/0.008/2.35/0.31	0.92/2.14
3: Alg.1-cold	114.6/116.4	785/785	0.04/0.008/2.56/0.31	0.56/1.35
4: Alg.1-cold,*sp*	102.9/104.7	785/785	0.04/0.008/2.20/0.33	0.54/1.33
5: Alg.1-warm	18.2/19.8	100/100	0.02/0.003/2.81/0.28	0.56/1.35
6: Alg.1-warm,*sp*	15.3/16.9	100/100	0.02/0.003/2.96/0.31	0.54/1.33

The warm-start variant of Alg.1 outperforms the interior-point method obtained from CVXGEN by a factor of 4 while occupying 40% less memory.

Outline

Conclusion

Our work provides a viable approach to achieve a functional MPC on a PLC.

Conclusion

Our work provides a viable approach to achieve a functional MPC on a PLC.

Essential aspects include:

- ✓ automatic code generation,
- ✓ MPC problem size reduction methods,
- ✓ MPC structure preserving transformations,
- ✓ a primal-dual first-order method, and
- ✓ embedded real-time considerations for the PLC

The results further motivate the use of first-order methods in embedded MPC.