

Paper E

A Comparison of Implementation Strategies for MPC

Lie, B., Dueñas Díez, M., and Hauge, T. A. (2002). *A Comparison of Implementation Strategies for MPC*, in Proceedings of International Symposium on Advanced Control of Industrial Processes, June 10-11, 2002, Kumamoto, Japan.

A few corrections are made to the original paper.

A Comparison of Implementation Strategies for MPC

Bernt Lie*, Marta Dueñas Díez, and Tor Anders Hauge
Telemark University College, P.O. Box 203, N-3901 Porsgrunn, Norway

Abstract

Four quadratic programming (QP) formulations of model predictive control (MPC) are compared with regards to ease of formulation, memory requirement, and numerical properties. The comparison is based on two example processes: a paper machine model, and a model of the Tennessee Eastman challenge process; the number of free variables range from 150 – 1400. Five commercial QP solvers are compared. Preliminary results indicate that dense solvers still are the most efficient, but sparse solvers hold great promise.

Keywords: Model predictive control; Quadratic programming; Problem formulation; Analysis of formulation; Comparison of QP solvers

1 Introduction

Model based predictive control (MPC) is the repeated use of optimal control over a given horizon; many introductory books dealing with MPC exist, e.g. Maciejowski (2002), Camacho and Bordons (1999), and Seborg *et al.* (1989). Most of the work on MPC has been centered on the use of linear models and quadratic performance indices. Common model types are impulse and step response models, ARMAX/CARIMA models, and state space models. In many cases, such models are input-output equivalent, and the choice of model is less important for the resulting value of the control input.

The performance index typically puts quadratic weights on the control deviation, the control variable, and/or quadratic weight on the control increment. In practice, control inputs will be constrained to lie within lower and upper bounds, while it is also of interest to introduce constraints on response variables, e.g. that the outputs are constrained to lie in a given region, etc. The most common MPC formulations are thus posed as quadratic programming (QP) problems.

The development of the MPC algorithms typically include relatively lengthy formula manipulations in order to end up with a QP problem with future control inputs

*Corresponding author: Bernt.Lie@hit.no

as the unknowns. An alternative approach is to keep variables such as states, outputs, control deviations, etc. as unknowns, and include the model and various definitions as linear equality constraints in the QP problem.

In this paper, we compare various formulations of the QP problem. In section 2, we formulate a standard MPC problem using state space models, and pose it as QP problems with a complete set of variables, with an intermediate set of variables, and with the basic future control inputs as variables (the common formulation). In section 3, we analyze the various formulations via two case studies. In section 4, we compare the computation time for various optimization algorithms and various QP formulations. In section 5, we draw some conclusions.

2 The MPC problem

We consider the state space model

$$x_{k+1} = Ax_k + Bu_k \quad (1a)$$

$$y_k = Cx_k + Du_k, \quad (1b)$$

where $u_k \in \mathbb{R}^{m \times 1}$ is the control input, $y_k \in \mathbb{R}^{r \times 1}$ is the controlled output, $x_k \in \mathbb{R}^{n \times 1}$ is the state, and the performance index J_k is:

$$J_k = \sum_{i=0}^{\infty} (e_{k+i}^T Q e_{k+i} + u_{k+i}^T R u_{k+i} + \Delta u_{k+i}^T S \Delta u_{k+i}).$$

Here, the output error e_k is

$$e_k = y_k - s_k, \quad (2)$$

where s_k is the set point, and the control increment Δu_k is

$$\Delta u_k = u_{ki} - u_{k-1}. \quad (3)$$

For open loop stable systems and some mild additional conditions, we can transform the infinite performance index into the following finite horizon index, see e.g. Muske and Rawlings (1993):

$$J_k = x_{k+N}^T \bar{Q} x_{k+N} + u_{k+N-1}^T S u_{k+N-1} + \sum_{i=0}^{N-1} (e_{k+i}^T Q e_{k+i} + u_{k+i}^T R u_{k+i} + \Delta u_{k+i}^T S \Delta u_{k+i}), \quad (4)$$

where \bar{Q} is found by solving the discrete Lyapunov equation:

$$\bar{Q} = C^T Q C + A^T \bar{Q} A. \quad (5)$$

With x_k known, denote the optimal input sequence by $u_{k+i|k}$, $i \in \{0, \dots, N-1\}$. By repeatedly solving the optimal control problem for each time index k , letting the control input be $u_k = u_{k|k}$ leads to a nominally stable closed loop system, Rawlings and Muske (1993).

One of the main advantages of MPC is the direct handling of constraints in the calculation of the control input. This feature is important, since all processes are subject to constraints. Actuators have a limited range of action

$$u^\ell \leq u \leq u^u \quad (6)$$

and a limited control increment

$$\Delta u^\ell \leq \Delta u_k \leq \Delta u^u. \quad (7)$$

Output constraints are mainly introduced for safety and quality reasons. Such constraints also arise when the exact values of some output variables y are less important as long as they remain within specified boundaries or “zones”. These constraints can be expressed as

$$y^\ell \leq y \leq y^u. \quad (8)$$

Other types of inequality constraints are viable, such as funnels and constraints on states. These extensions are in principle straightforward, and here we limit the discussion to the constraints discussed above.

3 The MPC problem formulated as QP problems

3.1 Standard QP problem

The general MPC formulation outlined above can be posed as a quadratic programming (QP) problem of the form

$$\begin{aligned} \min_z \quad & F(z) = \frac{1}{2} z^T H z + g^T z + k \\ \text{s.t.} \quad & \mathcal{A} z = a \\ & \mathcal{B} z \leq b \\ & z^\ell \leq z \leq z^u. \end{aligned} \quad (9)$$

where the value of k does not change the optimal solution z , and hence is not discussed further. If inequality constraints are passive, the solution can be found by solving the linear equation $\mathcal{L}v = \nu$ where

$$\mathcal{L} = \begin{pmatrix} H & \mathcal{A}^T \\ \mathcal{A} & 0 \end{pmatrix}, \quad v = \begin{pmatrix} z \\ \lambda \end{pmatrix}, \quad \nu = \begin{pmatrix} -g \\ a \end{pmatrix}, \quad (10)$$

and λ is the Lagrange multiplier.

Table 1: Notation used in MPC formulation.

Notation	Matlab form
$I_n \in \mathbb{R}^{n \times n}$	<code>eye(n)</code>
$I_{n,k} = \text{diag}(1_{n- k \times 1}, k)$	<code>diag(ones(n-abs(k),1),k)</code>
$0_{m \times n} \in \mathbb{R}^{m \times n}$	<code>zeros(m,n)</code>
$1_{m \times n} \in \mathbb{R}^{m \times n}$	<code>ones(m,n)</code>
$A \in \mathbb{R}^{m \times n} : \begin{array}{l} \dim_1 A = m \\ \dim_2 A = n \end{array}$	<code>[m,n] = dim(A)</code>
$A \otimes B$	<code>kron(A,B)</code>
$\text{diag}(A_1, \dots, A_n)$	— (block diagonal)
$\text{rot}_{90} I_N$	<code>rot(eye(N),1)</code>

3.2 Complete set of variables

Although not the most common formulation, we first define the vector of unknowns z as:

$$z^T = \begin{pmatrix} u_k^T & \dots & u_{k+N-1}^T & x_{k+1}^T & \dots & x_{k+N}^T \\ y_k^T & \dots & y_{k+N-1}^T & e_k^T & \dots & e_{k+N-1}^T \\ \Delta u_k^T & \dots & \Delta u_{k+N-1}^T & & & \end{pmatrix} \quad (11)$$

Matrix H and vector g of eq. 9 are determined from the requirement that J_k of eq. 4 should equal $F(z)$ in eq. 9. The constraints $\mathcal{A}z = a$ contain the dynamic model in eq. 1 and the definitions in eqs. 2 – 3. For the MPC problem defined here, inequality $\mathcal{B}z \leq b$ is empty, while physical, safety, and quality constraints of Section 2 is contained in z^ℓ and z^u .

In formulating the matrices, the notation of Table 1 is used. The following matrices result:

$$H = \text{diag}(2(I_{N-1} \otimes R), 2(R + S), \quad (12)$$

$$\begin{aligned} & 0_{(N-1) \dim_1 \bar{Q} \times (N-1) \cdot \dim_2 \bar{Q}}, \\ & 0_{\dim_1 Q \times N \cdot \dim y}, 2\bar{Q}, 0_{N \dim y \times N \cdot \dim y}, \\ & 2(I_N \otimes Q), 2(I_N \otimes S) \end{aligned}$$

$$g = 0_{N \cdot (2m+n+2r) \times 1} \quad (13)$$

$$\mathcal{A} = \begin{pmatrix} -(I_N \otimes B) & \mathcal{A}_{12} & 0 & 0 & 0 \\ -(I_N \otimes D) & \mathcal{A}_{22} & I_{N \cdot r} & 0 & 0 \\ 0 & 0 & I_{N \cdot r} & -I_{N \cdot r} & 0 \\ \mathcal{A}_{41} & 0 & 0 & 0 & I_{N \cdot m} \end{pmatrix} \quad (14)$$

where matrices \mathcal{A}_{ij} are defined in Table 2.

Table 2: Matrices for complete variable set QP-formulation.

$$\begin{aligned} \mathcal{A}_{12} &= I_{N \cdot n} - I_{N, -1} \otimes A \\ \mathcal{A}_{22} &= -I_{N, -1} \otimes C \\ \mathcal{A}_{41} &= -I_{N \cdot m} + I_{N, -1} \otimes I_m \end{aligned}$$

$$a = \left(\begin{pmatrix} Ax_k \\ 0_{(N-1) \cdot n \times 1} \end{pmatrix}^T, \begin{pmatrix} Cx_k \\ 0_{(N-1) \cdot r \times 1} \end{pmatrix}^T, \mathbf{s}^T, \begin{pmatrix} -u_{k-1} \\ 0_{(N-1)m \times 1} \end{pmatrix}^T \right), \quad (15)$$

$$z^\ell = \begin{pmatrix} 1_{N \times 1} \otimes u^\ell \\ -\infty \cdot 1_{N \cdot n \times 1}^T \\ 1_{N \times 1} \otimes y^\ell \\ -\infty \cdot 1_{N \cdot r \times 1}^T \\ 1_{N \times 1} \otimes \Delta u^\ell \end{pmatrix}, \quad z^u = \begin{pmatrix} 1_{N \times 1} \otimes u^u \\ \infty \cdot 1_{N \cdot n \times 1}^T \\ 1_{N \times 1} \otimes y^u \\ \infty \cdot 1_{N \cdot r \times 1}^T \\ 1_{N \times 1} \otimes \Delta u^u \end{pmatrix} \quad (16)$$

The dimensions of the complete variable set QP problem are given by $\dim z = N \cdot (n + 2m + 2r) \times 1$ and $\dim a = N \cdot (n + m + 2r) \times 1$. Typically, the definition of z as in eq. 11 leads to sparse matrices H and \mathcal{A} .

3.3 Intermediate set of variables

From the full QP formulation, we eliminate e_k , Δu_k , and y_k . The resulting vector of unknowns is:

$$z^T = (u_k^T \quad \dots \quad u_{k+N-1}^T \quad x_{k+1}^T \quad \dots \quad x_{k+N}^T). \quad (17)$$

The matrices and vectors in the QP formulation are

$$H = 2 \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}, \quad (18)$$

$$g = \begin{pmatrix} 2(x_k^T C^T Q D - u_{k-1}^T S)^T \\ 0_{(N-1)m + Nn \times 1} \end{pmatrix} \quad (19)$$

$$\mathcal{A} = (-I_N \otimes B \quad I_{N \cdot n} - I_{N, -1} \otimes A) \quad (20)$$

$$a = \begin{pmatrix} Ax_k \\ 0_{(N-1)n \times 1} \end{pmatrix} \quad (21)$$

$$z^\ell = \begin{pmatrix} 1_{N \times 1} \otimes u^\ell \\ -\infty \cdot 1_{N \cdot n \times 1} \end{pmatrix}, \quad z^u = \begin{pmatrix} 1_{N \times 1} \otimes u^u \\ \infty \cdot 1_{N \cdot n \times 1} \end{pmatrix} \quad (22)$$

$$\mathcal{B} = \begin{pmatrix} \mathcal{B}_{11} & \mathcal{B}_{12} \\ -\mathcal{B}_{11} & -\mathcal{B}_{12} \\ \mathcal{B}_{31} & 0_{N \cdot m \times N \cdot n} \\ -\mathcal{B}_{31} & 0_{N \cdot m \times N \cdot n} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}. \quad (23)$$

Table 3: Matrices for intermediate variable set QP formulation.

$$\begin{aligned}
H_{11} &= I_N \otimes (2S + R + D^T Q D) - I_{N,-1} \otimes S - I_{N,+1} \otimes S \\
H_{12} &= I_{N,+1} \otimes D^T Q C = H_{21}^T \\
H_{22} &= \begin{pmatrix} I_{N-1} \otimes C^T Q C \\ \bar{Q} \end{pmatrix} \\
\mathcal{B}_{11} &= I_N \otimes D, \quad \mathcal{B}_{12} = I_{N,-1} \otimes C \\
\mathcal{B}_{31} &= I_{N \cdot m} - I_{N,-1} \otimes I_m \\
b_1 &= \begin{pmatrix} y^u - C x_k \\ 1_{N-1 \times 1} \otimes y^u \end{pmatrix}, \quad b_2 = \begin{pmatrix} -y^\ell + C x_k \\ -1_{N-1 \times 1} \otimes y^\ell \end{pmatrix} \\
b_3 &= \begin{pmatrix} \Delta u^u + u_{k-1} \\ 1_{N-1 \times 1} \otimes \Delta u^u \end{pmatrix}, \quad b_4 = \begin{pmatrix} -\Delta u^\ell - u_{k-1} \\ -1_{N-1 \times 1} \otimes \Delta u^\ell \end{pmatrix}
\end{aligned}$$

The matrices encountered in equations 18 – 23 that have not been defined yet, are defined in Table 3. The dimensions of the intermediate variable set QP problem are given by $\dim z = N \cdot (n + r) \times 1$, $\dim a = N \cdot n \times 1$, and $\dim b = N(r + m)$. The definition of z as in eq. 17 leads to sparse matrices H , \mathcal{A} , and \mathcal{B} .

3.4 Basic set of variables

The most common QP formulation is found by using the equality constraints to eliminate all unknowns except the future control inputs, hence:

$$z^T = (u_k^T \quad \dots \quad u_{k+N-1}^T). \quad (24)$$

In this case, there are no equality constraints. The matrices and vectors of the QP formulation are

$$\begin{aligned}
H &= 2(\mathcal{H}_{N-1} (I_N \otimes Q) \mathcal{H}_{N-1} + (I_N \otimes R) \\
&\quad + \Psi^T (I_N \otimes S) \Psi + \Omega + P^T \mathcal{C}_k^T \bar{Q} \mathcal{C}_k P), \quad (25)
\end{aligned}$$

$$\begin{aligned}
g^T &= 2(\mathcal{O}_N x_k - \mathbf{s})^T (I_N \otimes Q) \mathcal{H}_{N-1} \\
&\quad + 2u_{k-1}^T L^T (I_N \otimes S) \Psi + 2x_k^T (A^N)^T \bar{Q} \mathcal{C}_k P \quad (26)
\end{aligned}$$

$$\mathcal{B} = \begin{pmatrix} \Psi \\ -\Psi \\ \mathcal{H}_{N-1} \\ -\mathcal{H}_{N-1} \end{pmatrix}, \quad b = \begin{pmatrix} 1_{N \times 1} \otimes \Delta u^u - L u_{k-1} \\ -1_{N \times 1} \otimes \Delta u^\ell + L u_{k-1} \\ 1_{N \times 1} \otimes y^u - \mathcal{O}_N x_k \\ -1_{N \times 1} \otimes y^\ell + \mathcal{O}_N x_k \end{pmatrix} \quad (27)$$

$$z^\ell = 1_{N \times 1} \otimes u^\ell, \quad z^u = 1_{N \times 1} \otimes u^u \quad (28)$$

The matrices encountered in equations 25 – 27 that have not been defined yet, are defined in Table 4.

The dimensions of the intermediate variable set QP problem are given by $\dim z = N \cdot r \times 1$ and $\dim b = 2N \cdot (m + r) \times 1$. The definition of z as in eq. 24 leads to dense matrices H and \mathcal{B} .

Table 4: Matrices for basic variable set QP formulation.

$$\begin{aligned}
& \Psi = I_{N \cdot m} - I_{N, -1} \otimes I_m, \\
L &= \begin{pmatrix} -I_m \\ 0_{(N-1) \times m} \end{pmatrix}, \quad \mathbf{s}^T = (s_k^T \quad \cdots \quad s_{k+N-1}^T) \\
\Omega &= \begin{pmatrix} 0_{N-1 \times N-1} & 0_{N-1 \times 1} \\ 0_{1 \times N-1} & 1 \end{pmatrix} \otimes S, P = \text{rot}_{90}(I_{N \cdot m}) \\
C_k &= (B \quad AB \quad \cdots \quad A^{k-1}B), \\
\mathcal{O}_k^T &= \begin{pmatrix} C^T & (CA)^T & \cdots & (CA^{k-1})^T \end{pmatrix}, \\
\mathcal{H}_k &= \begin{pmatrix} D & 0_{\dim D} & \cdots & 0_{\dim D} \\ CB & D & \cdots & 0_{\dim D} \\ \vdots & \vdots & \ddots & \vdots \\ CA^{k-1}B & CA^{k-2}B & \cdots & D \end{pmatrix}
\end{aligned}$$

3.5 Basic variable set from QR factorization

It is possible to eliminate equality constraints by means of e.g. QR factorization. This is an alternative to the formula manipulation needed to reach the results in the previous section. Denoting the matrices in section 3.2 by subscript c , we have:

$$\mathcal{A}_c = \tilde{Q}\tilde{R}, \quad (29)$$

where \tilde{Q} is orthogonal and \tilde{R} is an upper triangular matrix, and $\dim \tilde{R} = \dim \mathcal{A}_c$. \tilde{R} is then partitioned into:

$$\tilde{R} = \begin{pmatrix} R_{11} & R_{12} \end{pmatrix} \quad (30)$$

where \tilde{R}_{11} is square and invertible for well posed MPC problems; z_c is partitioned into:

$$z_c^T = \begin{pmatrix} z_1^T & z_2^T \end{pmatrix} \quad (31)$$

where $\dim z_1$ is the number of columns in R_{11} and $\dim z_2 = N \cdot m$. This leads to

$$z_1 = R_{11}^{-1} \left(\tilde{Q}^T a_c - R_{12} z_2 \right). \quad (32)$$

By eliminating the equality constraint, the matrices in the QP formulation become:

$$\begin{aligned}
H &= \begin{pmatrix} -R_{11}^{-1}R_{12} \\ I \end{pmatrix}^T H_c \begin{pmatrix} -R_{11}^{-1}R_{12} \\ I \end{pmatrix} \\
g^T &= \begin{pmatrix} R_{11}^{-1}\tilde{Q}^T a_c \\ 0 \end{pmatrix}^T H_c \begin{pmatrix} -R_{11}^{-1}R_{12} \\ I \end{pmatrix} \\
\mathcal{B} &= \begin{pmatrix} \mathcal{B}_1 \\ -\mathcal{B}_1 \end{pmatrix}, \quad \mathcal{B}_1 = \begin{pmatrix} -R_{11}^{-1}R_{12} \\ I_{N \cdot m} \end{pmatrix}
\end{aligned}$$

Table 5: Case studies.

	1a	1b	1c	2a	2b	2c
Process	PM	PM	PM	TE	TE	TE
N	50	100	200	50	100	200

$$b = \begin{pmatrix} z_c^u - \begin{pmatrix} R_{11}^{-1} \tilde{Q}^T a_c \\ 0_{N \cdot m \times N \cdot m} \end{pmatrix} \\ -z_c^\ell + \begin{pmatrix} R_{11}^{-1} \tilde{Q}^T a_c \\ 0_{N \cdot m \times N \cdot m} \end{pmatrix} \end{pmatrix}$$

$$z_2^\ell = -\infty \cdot \mathbf{1}_{N \cdot m \times 1}, \quad z_2^u = \infty \cdot \mathbf{1}_{N \cdot m \times 1}.$$

When z_2 is found, we can compute z_1 from eq. 32. However, since the first element of z_2 is Δu_k , we can find the desired u_k as $\Delta u_k + u_{k-1}$, hence z_1 is really not needed. The dimensions of the QR reduced basic variable set problem are given by $\dim z_2 = N \cdot m \times 1$ and $\dim b = 2 \cdot \dim z_c$. Since \tilde{a}_c may change with time index k , it is necessary to also store $R_{11}^{-1} \tilde{Q}^T$ which is of dimension $\dim a_c \times \dim a_c$. This formulation leads to dense matrices H and \mathcal{B} , and a dense \tilde{Q} matrix.

4 Analysis of QP problems

The formulations with a complete set of variables (**C**, section 3.2), an intermediate set of variables (**I**, section 3.3), the basic set of variables (**B**, section 3.4), and the basic set of variables as found via QR factorization (**QR**, section 3.5) are compared with regards to sparsity, the use of computer memory, and the conditioning of the formulations.

In the discussion of sparsity and conditioning, we assume that possible inequality constraints are passive, and thus consider the sparsity and condition number of matrix \mathcal{L} in eq. 10.

The comparisons are based on two example processes. The first example process is a linearized fourth order paper machine (PM) model, with three inputs and three outputs; see Appendix A.1 for some details. The second example process is an identified 23 order model of the Tennessee Eastman (TE) Challenge Process, with seven inputs and ten outputs; see Appendix A.2 for some details. All computations in this paper are based on Intel Pentium III PCs running at 750 MHz, and with 256 Mbyte RAM.

The case studies are described in Table 5, where the first row is our reference name for the case study, the second row describes which process is used (Paper Machine or Tennessee Eastman), and the third row is the prediction horizon used in the formulation.

The **B** and **QR** formulations have totally dense Lagrange matrices \mathcal{L} , while the sparsity patterns for the **C** and **I** formulations are displayed in Table 6.

Table 7 displays the memory used by the matrices and vectors in the MPC formulation.

Table 6: Sparsity patterns for sparse QP formulations, case 2a.

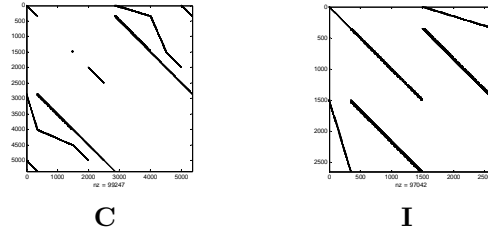


Table 7: Memory used (kbytes) for case studies.

	1a	1b	1c	2a	2b	2c
B	740	2919	11598	4962	19443	76965
QR	2888	11535	**	27380	**	**
I	79	158	317	955	1919	3846
C	78	154	308	722	1445	2892

Elements marked with “**” denotes that the computer ran out of memory during computation.

Table 8 displays the condition number of matrix \mathcal{L} .

5 Comparison of algorithms for solving the QP problems

The QP solvers used in this study are (i) `quadprog`, (The MathWorks, Inc. 2000); (ii) `qld`, available in Tomlab, Holmström (2001); as well as the following solvers which are available with a Tomlab interface: (iii) `lssol`, Gill *et al.* (1986); (iv) `qpopt`, Gill *et al.* (1995); and (v) `sqopt`, Gill *et al.* (1997). The `sqopt` solver is the only one of these that fully handles sparse matrices. The `quadprog` solver can be used with

Table 8: Condition number for QP formulations. Some computations required virtual memory. Computations were terminated after 1 hour of computing time, and are marked with “**”.

	B	QR	I	C
1a	2.1×10^3	8×10^5	1.3×10^{19}	8.9×10^{18}
1b	2.3×10^3	2.7×10^6	1.4×10^{19}	1.1×10^{19}
1c	2.3×10^3	**	1.4×10^{19}	1.1×10^{19}
2a	7×10^{12}	2.7×10^{15}	6.5×10^{13}	1.6×10^{14}
2b	7×10^{12}	**	6.6×10^{13}	**
2c	7×10^{12}	**	**	**

Table 9: Identification of solver and QP formulation.

Formulation	Solver	Notation
B	lssol	B ₁
B	qpopt	B ₂
B	qld	B ₃
B	quadprog	B ₄
QR	lssol	QR
I	sqopt	I
C	sqopt	C

Table 10: Total computation time (seconds) for case studies.

	1a	1b	1c	2a	2b	2c
B ₁	0.91	4.82	46.9	(7.84)*	(76.8)*	(592)*
B ₂	1.38	10.1	91.4	8.1	80.1	600
B ₃	1.42	7.45	54.5	15.4	97.1	3320
B ₄	15.3	49.7	283	19.5	96.9	**
QR	16.2	133	**	(754)*	**	**
I	3.86	20.3	85.8	96	(313)*	(994)*
C	7.10	32.1	119	196	(382)*	(139)*

sparse matrices only if there are no inequality constraints in the problem formulation.

We use the case studies of Table 5, with the notation of Table 9 to identify which solver is used in the formulations. In all cases, we simulate the controlled process for $T = 20$ time steps.

Table 10 displays the total time used by the computer to simulate the case studies with various MPC formulations and solvers. Table elements marked with “*” denotes that an optimization failure or optimization problem occurred. Elements marked with “**” denotes that the computer ran out of memory during computation.

Table 11 displays the time spent on the first optimization. The reason for including these results is that most solvers solve the optimization problem much slower the first time than the remaining iterations. Typical computation times for the remaining

Table 11: Computation time for first iteration (seconds) for case studies.

	1a	1b	1c	2a	2b	2c
B ₁	0.12	0.4	2.6	(0.33)*	(3.22)*	(27.2)*
B ₂	0.13	1.3	11	0.35	4.4	27
B ₃	0.15	0.67	3.26	0.72	4.28	200
B ₄	1.75	6.5	40	1.13	4.5	**
QR	0.12	1.11	**	(4.6)*	**	**
I	0.82	3.29	11.3	10	(35)*	(88.7)*
C	0.72	3.2	11	21	(57)*	(266)*

Table 12: Typical computation time for remaining iterations (seconds) for case studies.

	1a	1b	1c	2a	2b	2c
B₁	0.03	0.15	1.9	(0.29)*	(3.14)*	(23.9)*
B₂	0.025	0.15	1.94	0.30	3.18	24.2
B₃	0.05	0.29	2.28	0.67	4.14	150
B₄	0.6	1.6	8.2	0.85	4.2	**
QR	0.05	0.26	**	(0.6)*	**	**
I	0.15	0.9	3.9	4.5	(14.6)*	(3)*
C	0.33	1.5	5.6	9.0	(17.2)*	(37)*

iterations are given in Table 12.

6 Conclusions

In this paper, we have discussed four formulations of a standard MPC problem. The formulation of section 3.2 (**C**) is, in our view, the most straightforward formulation from the pedagogical point of view. The formulation in section 3.5 (**QR**) only requires knowledge of linear algebra in addition to formulation **C**, and is also easy to present. The formulations in sections 3.3 (**I**) and 3.4 (**B**) utilize various degrees of elimination of equality constraints, where formulation **B** is the most demanding to present, yet it is also the most common formulation.

Formulations **C** and **I** both lead to sparse matrices, Table 6, and thus the memory requirement increases more or less linearly with the horizon N of the performance index, Table 7, while for the dense matrix formulations **B** and **QR** the memory requirement increases quadratically with N ; the **QR** formulation is most demanding. In fact, the formulations **C** and **I** can be said to be *supersparse* in the sense that it is possible to introduce special sparse matrix structures that take advantage of the fact that the involved matrices are constructed from the Kronecker product, where typically the system matrices and horizon length N contain all necessary information, and the size becomes independent of N . To take advantage of this, it would, however, be necessary to develop special matrix libraries for such data structures. Table 8 indicates that the sparse formulations (**C**, **I**) may be poorly conditioned, but this may also be a result of how the conditioning is defined.

A number of commercially available QP solvers have been tested and compared. Overall, the best combination of formulation and solver in our investigation appears to be the **B** formulation of section 3.4 and the **qpopt** solver, which manages to solve all test problems where the number of free variables ranges from 150 to 1400, see Tables 10 – 12: the largest problem requires less than 30s computation time for each iteration. The relatively poor performance of the **QR** formulation is mainly caused by the added memory requirement. The sparse solvers give relatively poor performance. With the memory advantage of the sparse formulations, it is to be hoped that sparse solvers will be tailor made to handle the (super-) sparse structures found in MPC

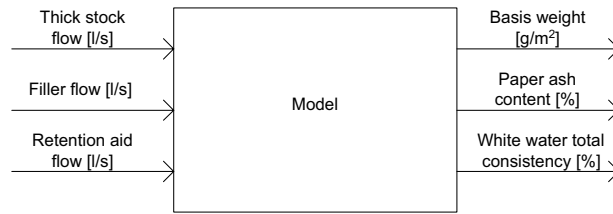


Figure 1: Inputs and outputs in PM6 model.

problems; such contributions are starting to appear, see Bartlett *et al.* (2002).

A Overview of example processes

A.1 Paper machine

A paper machine model has been developed for controlling certain key variables at paper machine 6 (PM6) at Norske Skog Saugbrugs, Norway. The original model is a fourth order nonlinear model with three inputs, three outputs and seven measured disturbances, and is described in detail in Hauge and Lie (2002). The model used in this paper is a linearized version where the measured disturbances are assumed constant. The inputs and outputs of the model are seen in Figure 1.

A.2 Tennessee Eastman Challenge Process

The Tennessee Eastman Challenge Process was defined in Downs and Vogel (1993), and a basic control structure for the process was suggested in McAvoy and Ye (1994). Recently, several subspace models for a part of this process were identified and compared, (Juricek, Seborg & Larimore 2001)Juricek *et al.* (2001). The subspace models all have 7 control inputs and 10 outputs, and the model that was found to give best predictions was based on the Canonical Variate Analysis (CVA) method and has 23 states. The seven inputs are (i) compressor recycle valve, (ii) condenser cooling water flow, (iii) setpoint for A feed, (iv) setpoint for D feed, (v) setpoint for C feed, (vi) set point for purge rate, and (vii) set point for reactor CW temp. The ten outputs are (i) recycle flow, (ii) reactor feed rate), (iii) reactor pressure, (iv) reactor temperature, (v) product separator temperature, (vi) product separator pressure, (vii) stripper pressure, (viii) stripper temperature, (ix) compressor work, and (x) separator CW temperature. The inputs and outputs have not been scaled, and the system that has been identified is rather stiff. The most promising prediction model from the subspace identification was graciously made available to the authors of this paper by the authors of (Juricek et al. 2001)Juricek *et al.* (2001).

References

- Bartlett, R. A., Biegler, L. T., Backstrom, J. & Gopal, V. (2002), 'Quadratic programming algorithms for large-scale model predictive control', *Journal of Process Control*. In press.
- Camacho, E. F. & Bordons, C. (1999), *Model Predictive Control*, Springer-Verlag London.
- Downs, J. & Vogel, E. (1993), 'A plant-wide industrial process control problem', *Computers and chemical engineering* **17**(3), 245–255.
- Gill, P. E., Hammarling, S. J., Murray, W., Saunders, M. A. & Wright, M. H. (1986), User's guide for lssol (version 1.0): A fortran package for constrained linear least-squares and convex quadratic programming, Technical Report SOL 86-1, Systems Optimization Laboratory (SOL), department of operations research, Stanford University.
- Gill, P. E., Murray, W. & Saunders, M. A. (1995), User's guide for qpopt 1.0: A fortran package for quadratic programming, Technical Report SOL 95-4, Systems Optimization Laboratory, Dept. Operations Research, Stanford University.
- Gill, P. E., Murray, W. & Saunders, M. A. (1997), 'User's guide for sqopt 5.3: A Fortran package for large-scale linear and quadratic programming'. (Draft, October 1997).
- Hauge, T. A. & Lie, B. (2002), 'Paper machine modeling at Norske Skog Saugbrugs: A mechanistic approach', *Modeling, Identification and Control* **23**(1), 27–52.
- Holmström, K. (2001), The TOMLAB optimization environment v3.0 user's guide, Technical report, HKH MatrisAnalys AB.
- Juricek, B. C., Seborg, D. E. & Larimore, W. E. (2001), 'Identification of the tennessee eastman challenge process with subspace methods', *Control Engineering Practice* **9**, 1337–1351.
- Maciejowski, J. (2002), *Predictive Control with Constraints*, Prentice Hall, Harlow, England.
- McAvoy, T. & Ye, N. (1994), 'Base control for the tennessee eastman problem', *Computers and chemical engineering* **18**(5), 383–413.
- Muske, K. R. & Rawlings, J. B. (1993), 'Model predictive control with linear models', *AIChE Journal* **39**(2), 262–287.
- Rawlings, J. B. & Muske, K. R. (1993), 'The stability of constrained receding horizon control', *IEEE Transactions on Automatic Control* **38**(10), 1512–1516.
- Seborg, D. E., Edgar, T. F. & Mellichamp, D. A. (1989), *Process Dynamics and Control*, John Wiley & Sons, Inc.

The MathWorks, Inc. (2000), 'Optimization toolbox for use with matlab, user's guide (version 2)'.