Proceedings of the American Control Conference
Albuquerque, New Mexico June 1997
0-7803-3832-4/97/$10.00 © 1997 AACC

Implementation issues in quadratic model predictive control

Atsushi Aoyama, Geoffrey Bell and Stephen P. Walsh
Centre for Process Systems Engineering
Imperial College of Science, Technology and Medicine
Prince Consort Road, London SW7 2BY (s.p.walsh@ic.ac.uk)

1 Introduction

Model predictive control (MPC) is the term used for a general class of multi-variable controllers that, at each sample interval, calculates a set of optimal inputs to optimise the future performance of the plant. The first of these calculated inputs is applied to the plant. It is especially applicable to processes with large time-constants for which the required closed loop bandwidth is low, facilitating the on-line solution of this optimisation problem.

Typically the optimisation problem the MPC controller solves is a quadratic program (QP), i.e. a quadratic objective function with linear constraints. The optimisation problem must be solved on-line at each iteration. Developments in the use of MPC are increasing the demand for computationally efficient algorithms. Larger sections of plant are being controlled with a single MPC controller and processes with faster response time are being controlled with MPC (e.g. power plant). One potential response to this demand is to use sparse QP methods. The first part of this paper compares a dense active set method of solving QPs with a sparse interior point method, providing an estimate of the problem size for which the sparse method becomes superior (section 2).

Another important trend in MPC is the demand for improved performance driven by the maturing of the technology, the competition between technology suppliers and the continuing pressure on the user companies for improved margins. Improving performance beyond that of a low bandwidth constraint management system — arguably typical of early applications — brings issues of robustness more to the fore. Two approaches to enhancing the trade-off between performance and robustness can be distinguished — systematic tuning of conventional MPC algorithms and modification of the algorithms themselves. In this paper we will investigate the singular value thresholding method of modifying the basic QP MPC method (section 3). This modification has been advocated as improving robustness margins [7] but we are not aware of any papers systematically investigating this claim.

Overall conclusions are discussed in section 4.

2 Comparison of two QP methods

Two methods of solving the MPC QP, a sparse interior point method [5] and a dense active set method similar to that of [1], are compared below. The comparison is based on a 3-input 3-output heavy oil fractionator model taken from the Shell Process Control Problem [2].

$$y(s) = \begin{pmatrix} \frac{4.05e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.72e^{-14s}}{60s+1} & \frac{6.90e^{-15s}}{40s+1} \\ \frac{4.38e^{-33s}}{20s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.20}{19s+1} \end{pmatrix} u(s) \quad (1)$$

The form of the process model used for construction of the controller is given in equation 2.

$$y_i(t) = a_i y_{past} + b_i u_{past} \qquad i = 1, \ldots, r \quad (2)$$

where

$$\begin{aligned} y_{past} &= [y_1(t-1), \ldots, y_1(t-l_1), \ldots, y_r(t-l_r)]^T \\ u_{past} &= [u_1(t-1), \ldots, u_1(t-k_1), \ldots, u_n(t-k_n)]^T \\ a_i &= [a_{1i}^1, \ldots, a_{1i}^{l1}, \ldots, a_{ri}^1, \ldots, a_{ri}^{lr}] \\ b_i &= [b_{1i}^1, \ldots, b_{1i}^{k1}, \ldots, b_{ni}^1, \ldots, b_{ni}^{kn}] \end{aligned}$$

a_i and b_i are arrays of coefficients for the ARX model (l and k are the orders), y_{past} and u_{past} are the past values and, r and n are the number of outputs and inputs respectively.

The QP problem may be expressed as in equation 3.

$$\min_u \left\{ \sum_{i=1}^{r} (y_i - y_{set,i})^2 + \lambda \sum_{j=1}^{n} \Delta u_j{}^2 \right\} \quad (3)$$

where

$$\begin{aligned} y_i &= [y_i(t+p_i), \ldots, y_i(t+1)]^T \\ y &= [y_1{}^T, \ldots, y_r{}^T]^T \\ u_i &= [u_i(t+m_i-1), \ldots, u_i(t)]^T \\ u &= [u_1{}^T, \ldots, u_n{}^T]^T \\ y_{set,i} &= [y_{set,i}(t+p_i), \ldots, y_{set,i}(t+1)]^T \\ \Delta u &= u(t) - u(t-1) \end{aligned}$$

λ is a scalar input weighting and $y_{set,i}(t)$ is the reference value for output y_i at time t. The prediction horizon, p, and control horizon, m, are the same in all simulations in this section. Relative weighting of different outputs and of different inputs is accomplished by scaling.

The matrix form of the objective function, which is used in solving the QP, is shown in equation 4.

$$\min_u \left\{ \sum_{i=1}^{r} \left({y_i}^T Q_i y_i + 2 q_i y_i \right) + \lambda \sum_{j=1}^{n} \left({u_j}^T S_j u_j + 2 s_j u_j \right) \right\} \tag{4}$$

where

$$Q_i = I$$

$$S_j = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

$$q_i = (-y_{set,i}(t+p), \dots, -y_{set,i}(t+1))$$

$$s_j = (0, \dots, 0, -u_i(t))$$

The objective function is subject to the equality constraints,

$$D \left(y^T u^T \right)^T = E \left({y_{past}}^T {u_{past}}^T \right)^T \tag{5}$$

where D and E are constructed from the ARX model of equation 2 describing the process response. There are additional box constraints on the input and output variables,

$$u_j^{lo} \le u_j \le u_j^{up} \qquad j = 1, \dots, n$$
$$y_i^{lo} \le y_I \le y_i^{up} \qquad i = 1, \dots, r$$

where y^{lo} and u^{lo} are the lower bounds, y^{up} and u^{up} are the upper bounds and y and u are the future values of the outputs and inputs respectively.

Simulations with the objective function in the form of equation 4 revealed that the sparse interior point method is faster than the dense active set method for horizon values larger than 10. This is because of the sparsity of the matrices Q_i and S_j, D and E are exploited by the sparse method.

The number of variables in the QP may be reduced, and the matrices made more dense at the same time, by substitution for the output variables, y in terms of the inputs u alone. Table 1 shows the number of variables and constraints of each type for the problem with and without reduction of the variable space. For the plant model being used, where $r \times p = n \times m$, substitution halves the variable space but results in inequality constraints on u replacing simple box constraints on the output variables.

The speed of solution using the dense active set method has a cubic relationship with the number of optimisation variables, so the halving the variable space may be expected to reduce the solution time by a factor of 8. However the solution time was found to be around 30 times faster, indicating that fewer iterations were required for the dense QP formulation. Solution times when using the sparse interior point method on the reduced variable QP were much longer than for the original problem as, for this example, the loss of sparsity outweighed the benefits of fewer variables. Based on these results, the appropriate basis of comparison for the two methods is between the sparse interior point method implemented on the QP without reduction and the dense active set method used on the reduced variable space QP form. Three cases were considered – unconstrained, input constraints active and output constraints active. Figure 1 shows the computation time for the two methods with horizon values up to two hundred on a SPARC 10 workstation. The relationship between computation time and the horizon values (number of variables) may be seen to be linear for the interior point method and cubic for the active set method. In the unconstrained case, the active set method on the reduced variable space problem was faster up to a horizon value of 80, while the interior point method was faster for higher horizon values. For the constrained cases, the interior point method was faster for horizons greater than about 60. The presence of active constraints increases the overall computational cost for both methods.

Based on these test results, the following conclusions are drawn.

Number of	without reduction	with reduction
variables	$r \times p + n \times m$	$n \times m$
equality constraints	$r \times p$	0
box constraints	$r \times p + n \times m$	$n \times m$
inequality constraints	0	$2 \times r \times p$

Table 1: Number of variables and constraints for each problem formulation

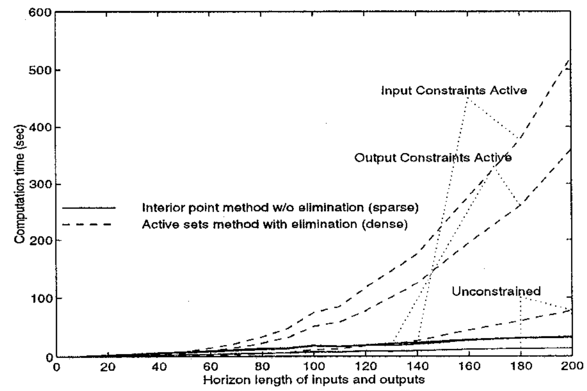

Figure 1: Comparison of computation time for dense and sparse QP methods with changing horizon length

The dense active set method should be used with output variables eliminated from the quadratic objective function. The sparse interior point method should be used without reduction of the variable space in the quadratic objective function.

The break-point in performance of the two algorithms has been observed to be around a horizon length of 80 for the 3 input, 3 output system considered, a total of 240 input variables (480 optimisation variables in the sparse method) in the unconstrained case.

For the constrained cases the break-point was around a horizon length of 60. The paper presented at the CPC-V by Stephen Wright [9] addresses the problem of efficient solution of the QP resulting from the MPC problem. It points out that the elimination of outputs may not be preferable when the output constraints are active. However, the test results above did not suggest major changes in the relative efficiencies of the methods in the presence of active output constraints.

Control horizons are often chosen smaller than prediction horizons and input blocking (setting blocks of inputs to a common single optimisation variable) is often used. Commonly only 10 optimisation variables are used per physical input.

The largest problems being tackled by most MPC vendors have about 25 physical inputs [3], suggesting about 250 optimisation variables in a dense QP. This corresponds closely with the estimated problem size at which sparse methods become competitive. As MPC is applied to larger problems, sparse methods may become commercially attractive.

The above conclusions are subject to the qualification that general QP algorithms were used. The algorithms were in fact developed with a view to solving QPs in sequential quadratic programming algorithms. The performance of both sparse and dense methods may be potentially improved from that shown above through specialisation to this particular QP application and further research is appropriate. However, the qualitative conclusions above seem likely to be robust.

3 Robustness and performance with ill-conditioned models

3.1 Introduction

An ill-conditioned model is one in which the process gain is highly dependent on the direction of the inputs. Given an input output model $y = Pu$ the degree of ill-conditioning may be quantified by a singular value decomposition of P. Large singular values correspond to high gain directions in which y changes substantially for small changes in u.

For tight control of ill-conditioned processes the controller should compensate for the strong variation of gain with input direction by applying large input signals in the direction of low process gain. If, due to plant-model mismatch, the calculated direction of the input does not correspond to the low gain direction but instead excites a high gain direction in the plant, performance will be poor and instability may result.

A strategy which has been proposed to deal with this problem is singular value thresholding, in which large singular values are eliminated from the controller, preventing the large, rapid input adjustments which may result in poor robustness. The performance and robustness of this method is compared to that of the usual approach of reducing input variation by increasing the penalty λ on input moves in the optimisation problem.

3.2 Methods to be compared

For linear process models, the predicted future output response may be expressed as the sum of the free and forced responses. The free response, y_{fr} is the expected behavior of the output assuming zero future control actions and is generated by recursive solution of the ARX model starting from measured past values. The forced response, y_{fo}, is the additional component of the output due to the predicted future controls, u, and may be expressed as,

$$y_{fo} = G\Delta u = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T \Delta u \tag{6}$$

where

$$\begin{aligned} \Sigma &= diag\{\sigma_i\} \\ y_{fo} &= [y_{fo,1}(t+p), \dots, y_{fo,1}(t+1), \dots, y_{fo,r}(t+1)]^T \end{aligned}$$

With the objective function of equation 3, the solution of the unconstrained MPC controller with a control weighting λ and reference signal y_{set}, can be shown to be

$$\Delta u = \left(G^T G + \lambda I\right)^{-1} G^T (y_{set} - y_{fr}) \tag{7}$$

Upon substitution of the singular value decomposition for G this becomes

$$\Delta u = \left[V \;\; diag\left\{ \frac{\sigma_i}{\sigma_i{}^2 + \lambda} \right\} U^T \right] (y_{set} - y_{fr}) \tag{8}$$

In the input weighting method (IW), increasing λ is used to suppress the weak singular values of the plant model G, as shown in equation 8. The weighting method affects all the controller singular values with each direction used to a greater or lesser extent in the controller. For $\lambda = 0$ it can be seen that the controller gain is $1/\sigma_i$. Therefore small singular values in the plant model result in high gain directions in the MPC controller.

Singular value thresholding (SVT) sets a limit on the smallest singular value of the plant which is to be used in the construction of the MPC controller. Those below the threshold are removed from the SVD and the new controller is reconstructed. This method has been claimed in the literature to improve the robust performance of the resulting MPC controller (eg. [7]). Due to the discontinuous nature of the method only a finite set of controllers may be generated (each particular direction is either used or not used). Applying singular value thresholding at σ_{thr} gives,

$$(G^T G)^{-1} G^T = V diag \left\{ \begin{array}{cc} 1/\sigma_i & \sigma_i \geq \sigma_{thr} \\ 0 & \sigma_i < \sigma_{thr} \end{array} \right\} U^T \quad (9)$$

3.3 Example Problem

Traditionally modelling of distillation columns for controller design has been performed using single state models of the form of equation 10.

$$\begin{pmatrix} dy_d \\ dx_b \end{pmatrix} = \frac{1}{194s+1} \begin{pmatrix} g_1 & g_2 \\ g_3 & g_4 \end{pmatrix} \begin{pmatrix} dL \\ dV \end{pmatrix} \quad (10)$$

This is because the slow time constant of the column, represented in this equation, is the one most easily identified. The faster transients are often unmodelled. This simple model is ill-conditioned over the entire frequency range. A more realistic model of a distillation column, as given in [4], which includes modelling of the internal flows of the column is shown in equation 11. This model is ill-conditioned at low frequencies but well-conditioned at high frequencies.

$$\begin{pmatrix} dy_d \\ dx_b \end{pmatrix} = \frac{1}{(194s+1)(15s+1)} \quad (11)$$
$$\begin{pmatrix} g_1(16.4s+1) & g_2(13.5s+1) \\ g_3(16.1s+1)e^{-2.5s} & g_4(16.1s+1) \end{pmatrix} \begin{pmatrix} dL \\ dV \end{pmatrix}$$

where $g_1 = 0.878$, $g_2 = -0.864$, $g_3 = 1.082$ and $g_4 = -1.096$. In the two models dy_d and dx_b are the overhead and bottoms concentrations while dL and dV are the reflux and boilup respectively.

The method described in [8] was modified to use the ARX model instead of the state space model and used to construct a linear time invariant controller for simulations of the unconstrained MPC controller. The results obtained are dependent not just on the optimisation problem solved but on the feedback mechanism used. In the results below, we predict the future $\hat{y}$ in using

$$\hat{y(k)} = y_m(k-1) + (1 - A(q^{-1}))\Delta y_m + B(q^{-1})\Delta u (12)$$

This prediction method uses the historical changes in measured output values, y_m, and inputs as the state estimates.

3.3.1 Performance comparison of singular value thresholding and input weighting

Simulations examining the performance of the singular value thresholding (SVT) and input weighting (IW) methods have been performed using the simple model for construction of the controller and the complex model for the simulation of the plant output response. This gives a very realistic plant-model mismatch. Horizon values of $p = 50$ and $m = 20$ have been used for the simulations.

A set-point change is applied in the direction requiring the greatest steady state input adjustment (weak direction $y = [-0.0078\ 0.0062]^T$).

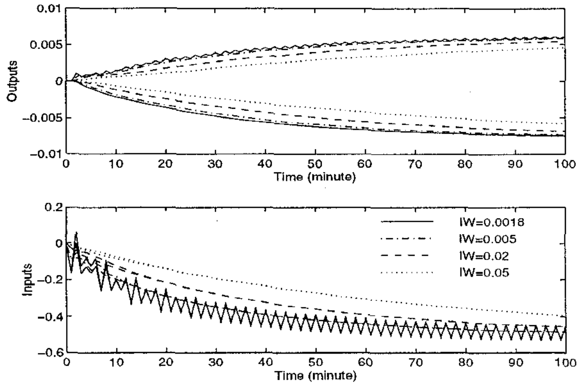

Figure 2: IW method output(top) and input(bottom) response for a step change in the weak singular value direction

Simulations showed that singular value thresholding with an incremental control weighting was unable to stabilise the system and provide set-point tracking simultaneously. With the removal of up to 35 smallest singular values, the closed loop response was observed to be unstable, while for upon removal of 36 or more the response stabilised but the set-point was not tracked.

The minimum input weight to stabilise the closed loop is about $\lambda = 1.8 \times 10^{-3}$. The continuous nature of the IW method means that an infinite set of stabilising controllers may be generated, allowing various closed loop response times to be obtained, as shown in Figure 2. To enable some comparison of performance for the IW and SVT methods to be performed the smallest stabilising input weight ($\lambda = 1.8 \times 10^{-3}$) was implemented in conjunction with singular value thresholding. For the horizon values of $p = 50$, $m = 20$ used, the responses for successive removal of the smallest plant singular values are shown in Figure 3. The results show that, for removal of 35 singular values, the closed loop response for a set-point change in the weak singular value direction is stable. As with the unweighted SVT case, when 36 singular values are removed the controller is unable to provide tracking of the set-point change. There is a noticeable gap in the possible performance of the controller using SVT due to the construction method being discontinuous and hence only able to generate a finite set of controllers.

3.3.2 Robustness comparison of singular value thresholding and input weighting

The system with a realistic form of plant-model mismatch represented in equations 10 and 11 was unable to be stabilised by pure singular value thresholding. Further evaluation of robustness was performed using the simple model as the plant and considering the effect of variation in the actuator gains as in [6].

For a setpoint change in the weak singular value direction, the simulation results for the SVT method with re-

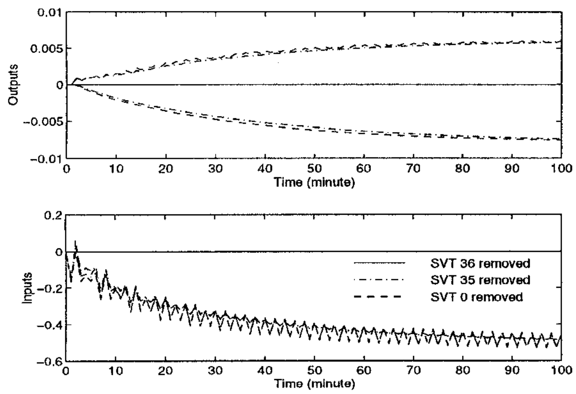

Figure 3: SVT method output(top) and input(bottom) response for step change in the weak singular value direction with stabilising input weight

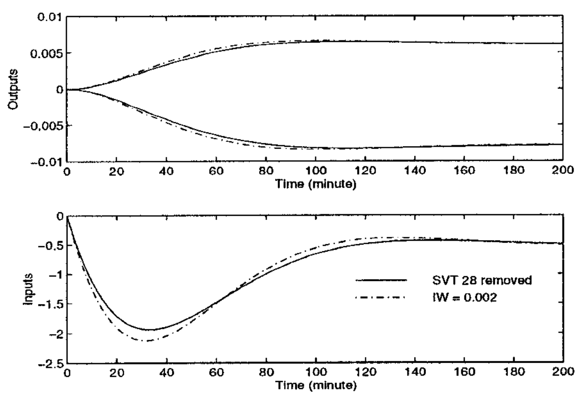

Figure 4: Performance of SVT (28 removed) and IW (λ = 0.002) for horizons m = 20 and p = 50

moval of the 28 smallest singular values are given in Figure 4. Through iteration, the input weight which gives a comparable performance has been found to be $\lambda = 0.002$. For these two controllers the amount of diagonal multiplicative input uncertainty which can be tolerated, before the closed loop becomes unstable, has been calculated using the MATLAB μ synthesis toolbox (Table 2). The results indicate that the controller constructed using singular value thresholding is significantly more robust to the diagonal input uncertainty used.

3.4 Conclusion

The results above show that the discontinuous nature of the SVT method may make it impossible to achieve a desirable performance robustness trade-off for a realistic model error. On the other hand, SVT may show superior robustness for a given performance level. Consequently SVT and IW should be viewed as complementary and not exclusive methods.

method	SVT	control weighting
input uncertainty fraction	0.469	0.215

Table 2: Robustness margins for SVT and IW methods

4 Conclusions

Sparse methods appear to be advantageous around the largest size of QPMPC problem being tackled commercially. Such methods may therefore become significant in the near future and deserve further study.

While singular value thresholding can sometimes offer robustness advantages over simple input weighting, it does not remove the need for the latter. The two methods should be viewed as complementary and be tailored to particular applications. Further work is needed to compare the methods in the presence of constraints.

References

[1] M. J. D. Powell, "On the Quadratic Programming Algorithm of Goldfarb and Idnani",*Mathematical Programming Study*,1985, 25, 46-61

[2] D. M. Prett and M. Morari, "Shell process Control Workshop", 1987, Butterworths

[3] Qin, S. J. and Badgwell, T. A. "An Overview of Industrial Model Predictive Control Technology" *Chemical Process Control - V*, 1996

[4] M. F. Sagfors and K. V. Waller, "Dynamic Low-Order Models for Capturing Directionality in Nonideal Distillation", *Ind. Eng. Chem. Res.*, 1995, 34(6), 2038-2050

[5] R. W. H. Sargent, "Infeasible-Interior-Point Method for Generalized Monotone Linear Complementarity Problems", pres. at*15th Int. Symp. on Mathematical Programming*, 1994

[6] S. Skogestad, M. Morari and J. C. Doyle, "Robust Control of Ill-Conditioned Plants: High Purity Distillation",*IEEE Transactions on Automatic Control*, 1988, 33(12), 1092-1105

[7] A. Ogden-Swift and J. Lu, "Practical Application of Robust Control", *Advances in Process Control IV*, 1995, 109-116

[8] N. M. C. de Oliveira and L. T. Biegler, "Constraint Handling and Stability Properties of Model Predictive Control",*AIChE journal*, 1994, 40(7), 1138-1155

[9] S. J. Wright, "Applying New Optimization Algorithms to Model Predictive Control",*Chemical Process Control - V*, 1996