Zakharov, Alexey; Boriouchkine, Alexander; Jämsä-Jounela, Sirkka-Liisa

A two-phase objective function for a constrained MPC and its application to a grinding plant

Published in:
IFAC PROCEEDINGS VOLUMES

DOI:
10.3182/20091014-3-CL-4011.00007

Published: 01/01/2009

Document Version
Peer reviewed version

Please cite the original version:

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.
A Two-phase Objective Function for a Constrained MPC and Its Application to a Grinding Plant

Alexey Zakharov*, Alexandre Boriouchkine*, Sirkka-Liisa Jämsä-Jounela*

* Helsinki University of Technology, P.O. box 6100, 02015, TKK, Finland. e-mail: alexey.zakharov@tkk.fi

Abstract: In the paper a two-phase MPC controller is described that uses a non-quadratic terminal cost. The terminal cost is computed with respect to a predefined control strategy, and it can therefore be treated as the second phase of the method. Because the problem of optimizing the MPC objective is not a quadratic programming problem, medium-scale, gradient-based optimization methods are used for the MPC implementation. The method is efficient for problems which have a lot of soft constraints, because it avoids the introduction of slack variables. The stability of the MPC is established in the paper, and the method is compared with an unconstrained MPC in the grinding plant optimization problem.

Keywords: MPC, MPC stability, MPC optimality, Grinding Plant optimization

1. INTRODUCTION

Model Predictive Control (MPC) is a form of control in which the current control action at every instant is determined as a solution of the finite time horizon optimal control problem, while using the current state of the plant as the initial state. Nowadays, the main focus in the literature is on problem, while using the current state of the plant as the solution of the finite time horizon optimal control which the current control action at every instant is determined. Model Predictive Control (MPC) is a form of control in which the current state of the plant is used to determine the control action at the next instant. MPC is a recent development in control theory, and it has been applied to a wide range of applications, such as chemical processes, power systems, and automotive systems.

Although early implementations of MPC ignored the constraints, modern MPC products, implemented as Quadratic Programming (QP) problems, are able to work with both soft and hard constraints. However, hard constraints can cause feasibility problems, especially when large disturbances appear. This is the reason why most of the MPC products enforce soft output constraints in dynamic optimization (Qin, Badgwell 2003). On the other hand, soft constraints increase the number of slack variables in QP problems, thus significantly increasing the computational requirements of MPCs. At the same time, the computation requirements of controllers are relatively critical for many applications, especially in the case of very fast or very large processes. This is the reason why a relatively large number of researchers have concentrated their efforts on reducing online computations (Bemporad et al. 2002, Pannocchia et al. 2007, Rao et al. 1998).

Another important requirement for MPC controllers is stability. Because of the finite horizon formulation, improperly tuned MPC controllers can be unstable. Early attempts to achieve stability included different prediction and control horizon approaches and the introduction of a terminal cost to the MPC objective. These methods were criticized in Bitmead et al. (1990) as ‘playing games’, because there were no clear conditions to guarantee stability. The stability of MPC was studied actively during the 90’s, and a comprehensive review of the studies is provided in Mayne et al. (2000). Finally, the Lyapunov function approach was employed to ensure stability. In brief, stability can be guaranteed if the terminal cost of the MPC objective function is not less than Bellman’s cost function $V(x)$ of the infinite horizon problem.

Although the optimality of MPC is not as critical as its stability, it is however also a highly desirable property. In Mayne’s et al. (2000) review, the discussion of sub-optimality research is presented with respect to Hamilton-Jacobi-Bellman theory. In other words, MPC sub-optimality can be achieved if the MPC terminal cost is close to the infinite horizon Bellman’s cost function $V(x)$. From the practical point of view, the role of sub-optimality is clear: the better the approximation of Bellman’s function used as the terminal cost, the lower the controller horizon that can provide an acceptable performance to the controller.

To summarize, from the theoretical point of view, Bellman’s cost function $V(x)$ of the infinite horizon formulation is the ideal candidate for the terminal cost of the MPC objective. It provides both stability and optimality to the controller. However, Bellman’s function $V(x)$ cannot be derived and used in an MPC controller, except in linear-quadratic regulator problems or in extremely low dimensional problems. Thus, one practically and feasible solution is to use an accurate upper estimation of Bellman’s cost function $V(x)$.

The idea of identifying the terminal cost of the MPC objective in the form of a quadratic function by solving the Riccati equations can be found in many papers: Chmielewski, Manousiouthakis (1996) and (2005), Szaier, and Damborg (1987). Although stability is attained within the approach, the class of quadratic functions seems to be very narrow, and it is
generally not possible to expect that a ‘good’ approximation of Bellman’s function can be found in this class. In addition, the class of quadratic functions is used because the quadratic programming technique is intended to be used for MPC objective optimization. However, if the number of soft constraints (and, as a result, the number of slack variables and constraints in related QP problems) is relatively high compared to the number of decision variables, then the QP technique stops being an attractive formulation due to the significant computational requirements related to slack variables.

In this paper, an attempt to use general optimization techniques (such as gradient-based methods) instead of quadratic programming for MPC objective optimization is presented. A more accurate approximation of Bellman’s infinite horizon cost function \( V(x) \) can be employed as the MPC terminal cost instead of the solution of the Riccati equation. As a result, the performance of the controller must be better than the quadratic programming oriented constrained MPCs. At the same time, the stability of the proposed controller is still ensured. The proposed methodology is tested with a grinding plant model, and the results are compared with QP-based MPCs.

The paper is organized as follows. Section 2 contains a description of the proposed MPC controller, and Section 3 a description of the grinding process. In Section 4 the results are presented and compared, and Section 5 contains the conclusions.

2. DESCRIPTION OF THE TWO-PHASE MPC OBJECTIVE FUNCTION

A simple linear discrete state space model is used to present the method:

\[
x(k+1) = A x(k) + B u(k),
\]

\[
y(k+1) = C x(k+1),
\]

where \( x = (x_1, x_2, ..., x_d) \) is the vector of the current state of the system, and \( y = (y_1, y_2, ..., y_m) \) is the vector of the controlled variables. For the sake of simplicity, let us assume that there is no noise in neither the dynamics or in the measurements, and that the system is observable (i.e. the state of the system can be defined through the values of its controlled variables). In addition, the process is assumed to have M linear constraints, which are treated as soft constraints due to the nature of the process or feasibility problems:

\[ P_i y(k) \leq q_i, i = 1,2,...,M . \]

Under the dynamics (1) and constraints (2), the optimal steady state of the system \( x^* \), corresponding to the optimal steady state control \( u^* \) and the controlled variable value \( y^* \), are usually defined by a higher level of the control hierarchy.

Now, let us consider a typical MPC controller with the time horizon equal to \( N \) and the following objective function:

\[
J(x(0),u) = \sum_{k=1}^{N} \sum_{i=1}^{M} a_i (\max(P_i y(k) - q_i,0))^2 + \sum_{k=1}^{N} \sum_{i=1}^{m} b_i (y_i(k) - y^*)^2 + F(x(N))
\]

The first item on the right hand side of (3) reflects the soft constraint penalties, while the second item introduces the penalty for non-optimal states of the system. The positive weights \( a_i \) and \( b_i \) are chosen to give the correct priority to the soft constraints and to the optimal setpoints. In actual fact, when \( F(x) \) is a quadratic function, then equation (3) is a typical form of the MPC objective for a problem with soft constraints. The difference between the proposed approach and traditional MPC techniques with the presence of soft constraints is the general nonlinear (non-quadratic) terminal cost \( F(x) \).

The Bellman’s cost function \( V(X) \), which is the ‘ideal candidate’ for the terminal cost \( F(x) \), can be introduced as the optimal value of the infinite horizon optimization problem:

\[
V(x) = \min_u \sum_{k=1}^{\infty} \sum_{i=1}^{M} a_i (\max(P_i y(k) - q_i,0))^2 + \sum_{k=1}^{\infty} \sum_{i=1}^{m} b_i (y_i(k) - y^*)^2
\]

Unfortunately, Bellman’s cost function cannot be computed without solving the infinite horizon optimization problem, and this is not practically feasible. However, if the optimal control strategy \( u^* \) is known, then the function \( V(x) \) can be easily estimated to any reasonable accuracy, because the row in (4) converges. Let us introduce the function \( V_u(x) \), which is obtained from equation (4) by using any predefined control strategy \( u \) instead of the optimal one:

\[
V_u(x) = \sum_{k=1}^{\infty} \sum_{i=1}^{M} a_i (\max(P_i y(k) - q_i,0))^2 + \sum_{k=1}^{\infty} \sum_{i=1}^{m} b_i (y_i(k) - y^*)^2
\]

First, it is obvious that \( V_u(x) \geq V(x) \), and therefore the stability is ensured if \( V_u(x) \) is used as the terminal cost \( F(x) \) in equation (3). However, the choice of \( u \) is still important, because it affects the performance of the MPC controller.

The choice of the control strategy \( u \), which is used to define the terminal cost as \( F(x) = V_u(x) \), is relatively arbitrary and can be performed in many ways. The control strategy \( u \) used in this work is given in this paragraph. First, the infinite horizon optimization problem (4) is transformed into the linear-quadratic regulator problem:

\[
\min_u \sum_{k=1}^{\infty} \sum_{i=1}^{M} a_i (P_i y(k) - q_i^*)^2 + \sum_{k=1}^{\infty} \sum_{i=1}^{m} b_i (y_i(k) - y^*)^2
\]
the optimal solution of which, \( u^{LQ}(x) \), can be obtained explicitly:
\[
G = (B'RB)^{-1}B'RA,
\]
\[
\begin{equation}
\begin{aligned}
\end{aligned}
\end{equation}
\]
where \( R \) is the solution of the following discrete Riccati equation:
\[
\begin{aligned}
R &= \sum_{i=1}^{M} C'P_{i}a_{i}P_{i} + \sum_{i=1}^{m} C'_{i}b_{i}C_{i} + A' (R - RB(B'RB)^{-1}B'R)A \\
\end{aligned}
\]
(8)
In this paper, the control strategy \( u^{LQ} \) and equation (5) are used for defining the terminal cost
\[
F(x) = V_{adQ}(x).
\]
(9)
Equation (5) contains the infinite rows but, in practice, only a finite number of first components can be taken into account. Since both \( u^{LQ}(x) \) and the dynamics (1) are linear, the rows converge with exponential speed, and the rate of convergence can be estimated on the basis of the largest eigenvalue of the matrix \( A-BG \). Thus, it is easy to choose a number of components \( K \) that give a satisfactory level of accuracy in (5):
\[
F(x) = \sum_{k=1}^{K} \sum_{i=1}^{M} a_{i} (\max(P_{i}y(k) - q_{i}, 0))^{2} + \sum_{k=1}^{K} \sum_{i=1}^{m} b_{i} (y_{i}(k) - y^{*})^{2}
\]
(10)
with \( u(x) = -Gx \).

Finally, the current control state is derived from the optimal solution of the problem of minimization of the MPC objective \( J(x(0), u) \), which is defined by equations (3),(7),(8),(10). The optimization problem no longer is a quadratic programming problem, but the optimization methods of general nonlinear programming are effective for this purpose. High efficiency can be achieved if the gradient of the objective function (3) is also entered, together with its value, in the optimization algorithm. Luckily, the gradient of \( F(x) \) can be easily computed as follows:
\[
\frac{\partial F(x)}{\partial x_{j}} = \sum_{k=1}^{K} \sum_{i=1}^{M} a_{i} 2(\max(P_{i}y(k) - q_{i}, 0)) \frac{\partial y_{i}(k)}{\partial x_{j}} + \sum_{k=1}^{K} \sum_{i=1}^{m} b_{i} 2(y_{i}(k) - y^{*}) \frac{\partial y_{i}(k)}{\partial x_{j}}
\]
\[
\frac{\partial y_{i}(k)}{\partial x_{j}} = C \left( A' e_{j} - \sum_{l=1}^{L} A'^{-1} B G e_{j} \right),
\]
(11)
where \( e_{j} \) is the vector with the \( j^{th} \) component that equals 1, and the other components are zero.

3. DESCRIPTION OF THE GRINDING PLANT
In this section, the grinding process is described, which is a suitable model for testing the two phase MPC controller. In particular, the process contains a large number of constraints that must be treated as soft constraints due to the feasibility problems. In this situation, it is reasonable to expect that the proposed method is highly efficient.

Communion is a huge consumer of electrical power because crushing rocks into powder requires a lot of energy. According to Pomerleau et al. (2000), grinding typically accounts for almost 50% of the costs of a concentrator and, as a result, the optimization of grinding mills is an extremely important research topic. The aim of economic optimization is to maximize the feed rate or to achieve the desired particle size distribution, thus making production more profitable.

In this paper we consider the ball-mill grinding circuit of a typical rod-mill. The ore is fed to the rod mill and then discharged into the pump sump. The slurry is then fed to a hydrocyclone, where it is separated into the underflow product and a recycled part, which is fed back to the ball-mill (for more details, see Lestage et al. 2002). The whole circuit is presented in Figure 1.

Fig. 1. Grinding circuit, Lestage et al. 2002.

There are two manipulated variables available in the model:
- \( u_{1} \): the rod-mill feed (t/h)
- \( u_{2} \): the pump sump water addition (m³/h)

and four output variables:
- \( y_{1} \): the hydrocyclone overflow density (% solids)
- \( y_{2} \): the fraction of particles smaller than 325 mesh (47Am) in the product (%)
- \( y_{3} \): the tonnage through the ball mill (t/h)
- \( y_{4} \): the pump sump level (%)

Typically, the grinding process can be described by means of a transfer function of the second order. We will use the transfer function presented in Lestage et al. 2002:
The model was converted into the discrete time state space form (1) with a sample time equal to 300s. The system is described with 15 state variables and four controlled variables.

The constraints are defined as product specifications: the hydrocyclone overflow density \( y_1 \) must be above 48% to meet the flotation requirements, and must be below 52% to avoid sedimentation problems. Product specification fineness \( y_2 \) is defined as 47% of the particles smaller than 47 \( \mu \)m. In order not to overload the ball mill, the throughput \( y_3 \) must not exceed 820t/h. The pump sump level \( y_4 \) must remain between 15% and 85%. The constraints are defined in the following order: lower and upper constraints on \( y_1 \), lower and upper constraints on \( y_2 \), upper constraint on \( y_3 \), and lower and upper constraints on \( y_4 \). Thus, matrices \( P \) and \( q \) take the following form:

\[
P = \begin{bmatrix}
1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & -1
\end{bmatrix},
\]

\[
q = [52\%, -48\%, 47\%, -47\%, 820\text{t/h}, 85\%, -15\%]^T.
\]

4. RESULTS

In this section the controller is compared with the unconstrained MPC described in Lestage et al. (2002) with a control horizon equal to 10 and prediction horizon equal to 20, and without any terminal costs:

\[
J(u) = \sum_{k=1}^{N} (u(k) - u(k-1))^T R (u(k) - u(k-1)) + \sum_{k=1}^{N} \sum_{i=AC}^M (y_i(k) - y_i^*)^T Q_i (y_i(k) - y_i^*)
\]

where AC is the set of variables the constraints of which are active, and \( R \) and \( Q \) contain the controls and constraints weights.
Table 2. Setpoints

<table>
<thead>
<tr>
<th>Time period (h)</th>
<th>Overflow solids (%)</th>
<th>Particles smaller 47μm (%)</th>
<th>Ball mill throughput t/h</th>
<th>Pump sump level (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-0.25</td>
<td>51,83</td>
<td>47.0</td>
<td>820.0</td>
<td>62.06</td>
</tr>
<tr>
<td>0.25-5</td>
<td>48.0</td>
<td>49.0</td>
<td>811.95</td>
<td>64.5</td>
</tr>
<tr>
<td>5-10</td>
<td>50.0</td>
<td>49.0</td>
<td>739.32</td>
<td>30.87</td>
</tr>
<tr>
<td>10-15</td>
<td>51,83</td>
<td>47.0</td>
<td>820.0</td>
<td>62.06</td>
</tr>
</tbody>
</table>

The controlled variables are compared in Figure 2, and the manipulated variables are presented in Figure 3. It is obvious that the two-phase controller reacts faster than the unconstrained one. At the same time, the manipulated variables generated by the two-phase controller are as smooth as those of the unconstrained MPC.

Fig. 2. Controlled variables: 2a – % of particles < 47 Am, 2b – overflow % solids, 2c – ball mill throughput, 2d – the pump sump level.

Fig. 3. Manipulated variables: 3a – rod mill feedrate, 3b – water to pump sump.

However, since the purpose of the constrained controller is to minimize the constraint violations, it is interesting to test both controllers with respect to it. The sums of the squares of the constraint violations, deviations from the optimal setpoints and squared variations of the manipulated variables are compared in Table 3:

Table 3. Constraints violation and setpoint deviations of unconstrained and two-phase MPC controllers

<table>
<thead>
<tr>
<th></th>
<th>Unconstr. MPC</th>
<th>Two-phase MPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraint 1</td>
<td>26.27</td>
<td>9.87</td>
</tr>
<tr>
<td>Constraint 2</td>
<td>54.47</td>
<td>55.30</td>
</tr>
<tr>
<td>Constraint 3</td>
<td>15 040</td>
<td>222</td>
</tr>
<tr>
<td>Constraint 4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Setpoint 1</td>
<td>91.98</td>
<td>68.34</td>
</tr>
<tr>
<td>Setpoint 2</td>
<td>54.47</td>
<td>55.30</td>
</tr>
<tr>
<td>Setpoint 3</td>
<td>123 614</td>
<td>117 261</td>
</tr>
<tr>
<td>Setpoint 4</td>
<td>20 760</td>
<td>18 833</td>
</tr>
<tr>
<td>Squared Variat.</td>
<td>205,66</td>
<td>41,13</td>
</tr>
<tr>
<td>Ball Mill Through.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Squared Variat.</td>
<td>339,68</td>
<td>252,25</td>
</tr>
<tr>
<td>Water to Pump</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To conclude, the presented constrained controller is able to significantly reduce the constraints violation compared the unconstrained MPC, which is an expected result. In addition, however, the two-phase MPC is also able to reduce the deviations of controlled variables from their setpoints, although constrained controllers typically sacrifice setpoint approximation quality in order to meet the constraint requirements.

5. CONCLUSION

A two-phase MPC controller is described in the paper that uses a non-quadratic terminal cost. Because the terminal cost can no longer be parameterized, it must be computed approximately as the sum of a finite number of components.
of row (5). Such an approximation is computed in each step of the optimization of the MPC objective function (3).

The control strategy $u$, which is used to compute the terminal cost (13), must stabilize the dynamics (1). However, the control strategy can be chosen arbitrarily within this requirement. In actual fact, the optimization used in the presented MPC can be considered as follows: the aim is to minimize the infinite horizon (and non-quadratic) objective function with respect to the first $N$ inputs when the rest of the inputs are defined by the predetermined control strategy $u$. The stability of the resulting controller is established in this work, but the choice of $u$ is still important because it can affect the performance of the MPC.

However, it should also be stated that large-scale optimization methods, based on the second derivative matrix (Hessian), are not effective for optimizing the described MPC objective, because the Hessian matrix has discontinuities when the active constraint set changes (a constraint is added or removed from the active constraints set). Although large-scale methods are more effective than medium-scale ones in a general situation, numerical experiments show that medium scale methods are preferable for two-phase MPC.

Although only piecewise-quadratic functions are used in objectives (3) and (5) in this paper, the method will work with any smooth convex functions. Thus, different nonlinear functions can be used to achieve specific aims. For example, very rapidly growing penalties, such as fourth order polynomials (instead of the traditional second order penalties), can be used to exclude ‘large’ violations of the constraints.

Non-linear dynamics is a problem for traditional quadratic programming oriented methods, because the optimization problem cannot be considered as a quadratic programming problem. However, since the optimization of objective (3) is already a general nonlinear programming problem, the proposed approach is able to deal with a non-linear dynamics in a very similar manner to the linear case (1). The only difference is in the computations of the partial derivatives of the controlled variables with respect to the state variables, which becomes slightly more complex in the nonlinear dynamics case.

In general, this method can be efficient in cases where many constraints must be treated as soft constraints due to feasibility problems or specific problems. The grinding plant example, which is used to demonstrate the power of the method, is one such case. For these kind of problems the unconstrained MPC controllers are fast, but their objective functions are symmetrical and the constraints are not taken into account properly. The constrained controllers require a large number of slack variables and their computational demands are therefore usually significantly higher than the method presented here. Another possible application of the method is for problems with non-linear dynamics or non-quadratic objective functions.

REFERENCES


