Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Output Tracking for a Milling Circuit using Model Predictive Static Programming

Johan D. le Roux * **Ian K. Craig** * **Radhakant Padhi** [1] *

* *Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, Pretoria, South Africa.*
(Corresponding author: derik.leroux@up.ac.za)

Abstract: A novel suboptimal control technique for output tracking was developed using model predictive static programming (MPSP). The control technique was applied to a nonlinear model of a single-stage closed grinding mill circuit in simulation. The control technique could determine a suboptimal input trajectory to maintain the output variables at their desired values within 7 iterations. The input trajectory determined after only 2 iterations was already close to the final solution. This technique shows promise to significantly reduce computational time to find a suboptimal input trajectory for output tracking in large industrial processes.

Keywords: grinding mill, model predictive control, model predictive static programming

1. INTRODUCTION

A grinding mill circuit forms a crucial part in the energy and cost-intensive comminution process of extracting valuable metals and minerals from mined ore. The ability to control the grinding mill circuit is of primary importance to achieve the desired product specification with regards to quality and production rate. The control objectives are mostly to improve the quality of the product, to maximise the throughput, to decrease the power consumption, to reduce the usage of grinding media and to improve process stability. These objectives are interrelated and necessitates trade-offs to be made (Hodouin, 2011). The challenges when controlling a grinding process are the strong coupling between variables, large time delays, uncontrollable disturbances, the variation of parameters over time, the nonlinearities in the process and instrumentation inadequacies (Chen et al., 2008).

In Coetzee et al. (2010), robust nonlinear Model Predictive Control (MPC) was applied to a grinding mill circuit in simulation and showed good results in the presence of large disturbances and parameter uncertainties, but the computational cost was such that the controller was not suitable for online application. In an effort to address this problem, Bemporad et al. (2000) showed that for a discrete-time linear time-invariant system the control law can be reduced to a simple linear function evaluation instead of an expensive quadratic program. Another technique for reducing the computational cost of MPC can be found in Wang and Boyd (2010), where the dimensionality of the problem is reduced by restructuring the quadratic programs found in MPC and only a few iterations are performed to solve the quadratic program using an appropriate interior-point method. Application of this method to systems with nonlinear dynamics has not yet been fully tested.

Recently, a novel suboptimal control design technique called Model Predictive Static Programming (MPSP) was developed by Padhi and Kothari (2009) for finite-horizon nonlinear problems with terminal constraints. This technique combines the philosophies of MPC and approximate dynamic programming to reduce a dynamic optimisation problem to a static optimisation problem, which significantly reduces computational complexity. The computational effectiveness of MPSP in the aerospace industry is well illustrated in Oza and Padhi (2012) and Halbe et al. (2013) for a class of systems described by

$$\begin{aligned} X_{k+1} &= F_k\left(X_k, U_k\right) \\ Y_k &= h\left(X_k\right) \end{aligned} \tag{1}$$

where the primary objective is to obtain an input projection U_k, $k = 1, 2, ..., N-1$ so that the output at the final time step Y_N goes to the desired output value Y_N^*.

Because only the final output value is considered in the technique described above, an output trajectory cannot be followed. To follow an output trajectory, MPSP was extended to include output tracking for a class of systems where the output is a function of the states only (Kumar and Padhi, 2014). However, this still presents a limitation for systems where the output is a function of both the states and the input. To use the MPSP control technique in comminution, this paper shows how MPSP is extended to include a class of systems where the output is a function of both the states and the input, and the control aim is to track a desired output setpoint. As far as the authors are aware, this is a novel technique. In order to illustrate the concept of this new control law, the technique is applied to a grinding mill circuit.

Results indicate that the control law developed in this paper is capable of maintaining the desired output setpoint for the grinding mill circuit. Although it takes 7 iterations of the algorithm to meet the user-defined toleration specifications, a good input trajectory is already available after only 1 iteration.

[1] Radhakant Padhi works as an Associate Professor in the Dept. of Aerospace Engineering, Indian Institute of Science, Bangalore, India. He is currently visiting University of Pretoria, South Africa.

2. OUTPUT TRACKING USING MPSP

The state dynamics and output equation of a general discrete nonlinear system can be written as

$$\begin{aligned} X_{k+1} &= F_k\left(X_k, U_k\right) \\ Y_k &= h\left(X_k, U_k\right) \end{aligned} \tag{2}$$

where $X \in \Re^n$, $U \in \Re^m$, $Y \in \Re^p$ represent the states, the input and the output of the system respectively, and k are time steps.

The primary objective of output tracking by means of MPSP is to find an input projection U_k, $k = 1, 2, ..., N$ so that the output Y_k goes to the desired output value Y_k^* for time steps N_{st} to N, i.e. $Y_k \to Y_k^* \; \forall \; k = N_{st}, N_{st}+1, ..., N$, where N is the full simulation time and $N_{st}|(N_{st} < N)$ is the settling time of the output. It is important to note that the output Y_k is a function of both the states X_k and the input U_k of the system.

For the suboptimal control technique presented here, it is necessary to start with an estimated input projection. The method to obtain a good estimated input projection is problem specific. Although the objective will not necessarily be met by the estimated input, the input can be improved by an iterative process where i is the iteration index which increases until the algorithm converges, i.e. $\frac{\|Y_k^i - Y_k^*\|}{\|Y_k^*\|} < \epsilon_k$, $\forall \; k = N_{st}, N_{st}+1, ..., N$, where Y_k^* is the desired output and ϵ_k is a user defined tolerance limit on the output error. The system shown in (2) can now be written as

$$\begin{aligned} X_{k+1}^i &= F_k\left(X_k^i, U_k^i\right) \\ Y_k^i &= h\left(X_k^i, U_k^i\right) \end{aligned} \tag{3}$$

The relationship of variables between consecutive iterations i and $i+1$ at time step k are

$$\begin{aligned} Y_k^{i+1} &= Y_k^i + \Delta Y_k^i \\ X_k^{i+1} &= X_k^i + \Delta X_k^i \\ U_k^{i+1} &= U_k^i + \Delta U_k^i \end{aligned} \tag{4}$$

The output Y_k^{i+1} at time step k and iteration $(i+1)$ can be expanded by Taylor series expansion, retaining only first order terms

$$\begin{aligned} Y_k^{i+1} &= h\left(X_k^{i+1}, U_k^{i+1}\right) \\ &= h\left(X_k^i + \Delta X_k^i, U_k^i + \Delta U_k^i\right) \\ &\approx Y_k^i + \left[\tfrac{\partial Y_k}{\partial X_k}\right] \Delta X_k^i + \left[\tfrac{\partial Y_k}{\partial U_k}\right] \Delta U_k^i \end{aligned} \tag{5}$$

Because $Y_k^{i+1} = Y_k^i + \Delta Y_k^i$, it is possible to write

$$\begin{aligned} \Delta Y_k^i &= Y_k^{i+1} - Y_k^i \\ \Delta Y_k^i &\approx \left[\tfrac{\partial Y_k}{\partial X_k}\right] \Delta X_k^i + \left[\tfrac{\partial Y_k}{\partial U_k}\right] \Delta U_k^i \end{aligned} \tag{6}$$

where ΔY_k^i is the error in the output at time k and iteration i.

The state X_{k+1}^{i+1} at time step $(k+1)$ and iteration $(i+1)$ can be expanded by Taylor series expansion retaining only first order terms and disregarding higher order terms (HOT)

$$\begin{aligned} X_{k+1}^{i+1} &= F_k\left(X_k^{i+1}, U_k^{i+1}\right) \\ &= F_k\left(X_k^i + \Delta X_k^i, U_k^i + \Delta U_k^i\right) \\ &= F\left(X_k^i, U_k^i\right) + \left[\tfrac{\partial F_k}{\partial X_k}\right] \Delta X_k^i + \left[\tfrac{\partial F_k}{\partial U_k}\right] \Delta U_k^i + HOT \\ &\approx X_{k+1}^i + \left[\tfrac{\partial F_k}{\partial X_k}\right] \Delta X_k^i + \left[\tfrac{\partial F_k}{\partial U_k}\right] \Delta U_k^i \end{aligned} \tag{7}$$

Because $X_{k+1}^{i+1} = X_{k+1}^i + \Delta X_{k+1}^i$, it is possible to write

$$\begin{aligned} \Delta X_{k+1}^i &= X_{k+1}^{i+1} - X_{k+1}^i \\ \Delta X_{k+1}^i &\approx \left[\tfrac{\partial F_k}{\partial X_k}\right] \Delta X_k^i + \left[\tfrac{\partial F_k}{\partial U_k}\right] \Delta U_k^i \end{aligned} \tag{8}$$

where ΔX_k^i is the error in the state and ΔU_k^i is the error in the input solution at time step k and iteration i. If small input deviations ($\Delta U_k^i = dU_k^i$), small state deviations ($\Delta X_k^i = dX_k^i$) and small output errors are assumed ($\Delta Y_k^i = dY_k^i$), the output error dY_k^i can be written in terms of the state error and input error at time step $(k-1)$

$$\begin{aligned} dY_k^i &= \left[\tfrac{\partial Y_k}{\partial X_k}\right] dX_k^i + \left[\tfrac{\partial Y_k}{\partial U_k}\right] dU_k^i \\ dY_k^i &= \left[\tfrac{\partial Y_k}{\partial X_k}\right] \left[\tfrac{\partial F_{k-1}}{\partial X_{k-1}}\right] dX_{k-1}^i + \\ &\quad \left[\tfrac{\partial Y_k}{\partial X_k}\right] \left[\tfrac{\partial F_{k-1}}{\partial U_{k-1}}\right] dU_{k-1}^i + \\ &\quad \left[\tfrac{\partial Y_k}{\partial U_k}\right] dU_k^i \end{aligned} \tag{9}$$

The error in the state dX_{k-1}^i can be expanded further in terms of dX_{k-2}^i and dU_{k-2}^i. And the error in the state dX_{k-2}^i can be expanded further in terms of dX_{k-3}^i and dU_{k-3}^i, and so on. This process can continue until state error dX_1^i

$$\begin{aligned} dY_k^i = {} & \left[A^k\right]^i dX_1^i + \left[B_1^k\right]^i dU_1^i + \left[B_2^k\right]^i dU_2^i + \\ & \left[B_3^k\right]^i dU_3^i + ... \left[B_{k-1}^k\right]^i dU_{k-1}^i + \left[\tfrac{\partial Y_k}{\partial U_k}\right] dU_k^i \end{aligned} \tag{10}$$

where

$$\left[B_j^k\right]^i = \left[\tfrac{\partial Y_k}{\partial X_k}\right] \left[\tfrac{\partial F_{k-1}}{\partial X_{k-1}}\right] \times ... \left[\tfrac{\partial F_{j+1}}{\partial X_{j+1}}\right] \left[\tfrac{\partial F_j}{\partial U_j}\right] \tag{11}$$

Because it is assumed that the initial condition is known, there is no error in the initial term, i.e. $dX_1 = 0$. The error in the output reduces to

$$dY_k^i = \sum_{j=1}^{k-1} \left[B_j^k\right]^i dU_j^i + \left[\frac{\partial Y_k}{\partial U_k}\right]_{\left(X_k^i, U_k^i\right)} dU_k^i \tag{12}$$

Note that in the derivation of (12) it has been assumed that the input variables at each time step are independent of the previous values of states and/or inputs. The input variables are seen as decision variables and independent decisions can be made at every point in time. Equation (12) represents the output sensitivity at time step k with respect to change in the input at all time steps prior to k. In order to reduce the computational requirements of the algorithm, $\left[B_j^k\right]^i$ can be computed recursively. It is intuitively clear that the effect of input changes at future time steps will not change the output vector at the current time step. Therefore, $\left[B_j^k\right]^i$ can be defined for all $k = 2, 3, ..., N$ and $j = 1, 2, ..., N$

$$\begin{aligned} &\left.\begin{aligned} \left[\phi_k^k\right]^i &= I_{n\times n} \\ \left[\phi_j^k\right]^i &= \left[\phi_{j+1}^k\right]^i \left[\tfrac{\partial F_j}{\partial X_j}\right] \\ \left[B_j^k\right]^i &= \left[\tfrac{\partial Y_k}{\partial X_k}\right] \left[\phi_{j+1}^k\right]^i \left[\tfrac{\partial F_j}{\partial U_j}\right] \end{aligned}\right\} \forall j < k \\ &\left[B_j^k\right]^i = [0]_{p\times m} \qquad\qquad\qquad\quad \forall j \geq k \end{aligned} \tag{13}$$

The primary objective of the suboptimal control technique can be defined by the following cost function

$$\begin{aligned} J^i &= \tfrac{1}{2}\textstyle\sum_{k=2}^{N}\left(Y_k^{i+1}-Y_k^*\right)^T Q_k\left(Y_k^{i+1}-Y_k^*\right)+ \\ &\quad \tfrac{1}{2}\textstyle\sum_{k=1}^{N}\left(U_k^{i+1}-U_k^i\right)^T R_k\left(U_k^{i+1}-U_k^i\right) \\ &= \tfrac{1}{2}\textstyle\sum_{k=2}^{N}\left(Y_k^i+dY_k^i-Y_k^*\right)^T Q_k\left(Y_k^i+dY_k^i-Y_k^*\right)+ \\ &\quad \tfrac{1}{2}\textstyle\sum_{k=1}^{N}\left(dU_k^i\right)^T R_k\left(dU_k^i\right) \\ &= \tfrac{1}{2}\textstyle\sum_{k=2}^{N}\left(dY_k^i-dY_k^{*i}\right)^T Q_k\left(dY_k^i-dY_k^{*i}\right)+ \\ &\quad \tfrac{1}{2}\textstyle\sum_{k=1}^{N}\left(dU_k^i\right)^T R_k\left(dU_k^i\right) \end{aligned} \tag{14}$$

where $dY_k^{*i} = Y_k^i - Y_k^*$. Using (12), the cost function can be written in terms of the input error dU_k

$$\begin{aligned} J^i &= \tfrac{1}{2}\textstyle\sum_{k=2}^{N}\left(\sum_{j=1}^{k-1}\left[B_j^k\right]^i dU_j^i+\left[\frac{\partial Y_k}{\partial U_k}\right]dU_k^i-dY_k^{*i}\right)^T \\ &\quad \times Q_k\left(\textstyle\sum_{j=1}^{k-1}\left[B_j^k\right]^i dU_j^i+\left[\frac{\partial Y_k}{\partial U_k}\right]dU_k^i-dY_k^{*i}\right)+ \\ &\quad \tfrac{1}{2}\textstyle\sum_{k=1}^{N}\left(dU_k^i\right)^T R_k\left(dU_k^i\right) \end{aligned} \tag{15}$$

The iteration index i is dropped throughout the rest of the document for the sake of simplicity. The objective is to minimize the cost function J^i for $dU_1, dU_2, ..., dU_N$. The equation corresponding to $\frac{\partial J}{\partial(dU_1)} = 0$ can be simplified to

$$\begin{aligned} \frac{\partial J}{\partial(dU_1)} &= \textstyle\sum_{k=2}^{N}\left(B_1^{k\,T} Q_k \sum_{j=1}^{k-1} B_j^k dU_j\right)+ \\ &\quad \textstyle\sum_{k=2}^{N} B_1^{k\,T} Q_k\left[\frac{\partial Y_k}{\partial U_k}\right]dU_k- \\ &\quad \textstyle\sum_{k=2}^{N} B_1^{k\,T} Q_k dY_k^* + R_1 dU_1 \end{aligned} \tag{16}$$

The first term in (16) can be simplified further as

$$\begin{aligned} &\textstyle\sum_{k=2}^{N}\left(B_1^{k\,T} Q_k \sum_{j=1}^{k-1} B_j^k dU_j\right) \\ &= \textstyle\sum_{l=2}^{N} B_1^{l\,T} Q_l B_1^l dU_1 + \sum_{l=3}^{N} B_1^{l\,T} Q_l B_2^l dU_2 + ... \\ &\quad B_1^{N\,T} Q_N B_{N-1}^N dU_{N-1} \\ &= C_{11}dU_1 + C_{12}dU_2 + ... + C_{1(N-1)}dU_{N-1} \end{aligned} \tag{17}$$

The second term in (16) can also be simplified to

$$\begin{aligned} &\textstyle\sum_{k=2}^{N}\left(B_1^k\right)^T Q_k\left[\frac{\partial Y_k}{\partial U_k}\right]dU_k \\ &= B_1^{2\,T} Q_2\left[\frac{\partial Y_2}{\partial U_2}\right]dU_2 + ...B_1^{N\,T} Q_N\left[\frac{\partial Y_N}{\partial U_N}\right]dU_N \\ &= D_{12}dU_2 + ... + D_{1N}dU_N \end{aligned} \tag{18}$$

Thus, $\frac{\partial J}{\partial(dU_1)} = 0$ can now be written as

$$\begin{aligned} &\textstyle\sum_{k=2}^{N} B_1^{k\,T} Q_k dY_k^* \\ &= C_{11}dU_1 + C_{12}dU_2 + ... + C_{1(N-1)}dU_{N-1} \\ &\quad +R_1 dU_1 + D_{12}dU_2 + ... + D_{1N}dU_N \end{aligned} \tag{19}$$

The equation corresponding to $\frac{\partial J}{\partial(dU_l)} = 0$ for $l = 2, ..., N-1$ can be simplified to

$$\begin{aligned} &\textstyle\sum_{k=2}^{N} B_l^{k\,T} Q_k dY_k^* + \frac{\partial Y_l}{\partial U_l}^T Q_l dY_l^* \\ &= C_{l1}dU_1 + C_{l2}dU_2 + ... + C_{l(N-1)}dU_{N-1}+ \\ &\quad D_{l2}dU_2 + ... + D_{lN}dU_N + R_l dU_l+ \\ &\quad \textstyle\sum_{j=1}^{l-1}\frac{\partial Y_l}{\partial U_l}^T Q_l B_j^l dU_j + \frac{\partial Y_l}{\partial U_l}^T Q_l \frac{\partial Y_l}{\partial U_l} dU_l \end{aligned} \tag{20}$$

The second last term of (20) can be simplified to

$$\begin{aligned} &\textstyle\sum_{j=1}^{l-1}\frac{\partial Y_l}{\partial U_l}^T Q_l B_j^l dU_j \\ &= \frac{\partial Y_l}{\partial U_l}^T Q_l B_1^l dU_1 + \frac{\partial Y_l}{\partial U_l}^T Q_l B_2^l dU_2 + ... \\ &\quad \frac{\partial Y_l}{\partial U_l}^T Q_l B_{l-1}^l dU_{l-1} \\ &= E_{l1}dU_1 + E_{l2}dU_2 + ... + E_{l(l-1)}dU_{l-1} \end{aligned} \tag{21}$$

The equation corresponding to $\frac{\partial J}{\partial(dU_N)} = 0$ can be simplified to

$$\begin{aligned} &\frac{\partial Y_N}{\partial U_N}^T Q_N dY_N^* \\ &= \textstyle\sum_{j=1}^{N-1}\frac{\partial Y_N}{\partial U_N}^T Q_N B_j^N dU_j + \frac{\partial Y_N}{\partial U_N}^T Q_N \frac{\partial Y_N}{\partial U_N} dU_N \\ &= E_{N1}dU_1 + E_{N2}dU_2 + ... + E_{N(N-1)}dU_{N-1}+ \\ &\quad \frac{\partial Y_N}{\partial U_N}^T Q_N \frac{\partial Y_N}{\partial U_N} dU_N \end{aligned} \tag{22}$$

Matrix $C \in \Re^{N\times N}$ is defined for $e = 1, ..., N-1$ and $j = 1, ..., N-1$ as

$$C_{ej} = \textstyle\sum_{l=(j+1)}^{N}\left(B_e^l\right)^T Q_l B_j^l \tag{23}$$

otherwise, $C_{ej} = [0]_{m\times m}$. Matrix $D \in \Re^{N\times N}$ is defined for $e = 1, ..., N-1$ and $j = e+1, ..., N$ as

$$D_{ej} = B_e^{j\,T} Q_j \frac{\partial Y_j}{\partial U_j} \tag{24}$$

otherwise, $D_{ej} = [0]_{m\times m}$. Matrix $E \in \Re^{N\times N}$ is defined for $j = 1, ..., N-1$ and $e = j+1, ..., N$ as

$$E_{ej} = \left(\frac{\partial Y_e}{\partial U_e}\right)^T Q_e B_j^e \tag{25}$$

otherwise, $E_{ej} = [0]_{m\times m}$.

Compiling all the equations for all times steps, the system of equations can be written as

$$\begin{aligned} [dU_e] = {}& [C_{ej} + D_{ej} + E_{ej} + \\ & \delta_{ej}\left(R_e + \frac{\partial Y_e}{\partial U_e}^T Q_e \frac{\partial Y_e}{\partial U_e}\right)\Big]^{-1}[b_e] \end{aligned} \tag{26}$$

where δ_{ej} is the Kronecker-delta function and b_e is defined for $e = 1, ..., N$ as

$$b_e = \sum_{k=2}^{N}\left(B_e^k\right)^T Q_k dY_k^* + \frac{\partial Y_e}{\partial U_e}^T Q_e dY_e^* \tag{27}$$

Finally, the updated input at time step $k = 1, ..., N$ is

$$U_k^{i+1} = U_k^i + dU_k^i \tag{28}$$

Although this is a novel technique and some issues are still to be explored, this paper attempts to show proof of concept. Future work involves, rigorous convergence guarantees, the consolidation of input and output equality, inequality and rate constraints, as well as investigation of the robustness of the model to modelling errors and the ability to reject noise and large disturbances.

3. CIRCUIT AND MODEL DESCRIPTION

The single-stage closed grinding mill circuit and model used to illustrate the control technique developed above is described below. The three main elements in Figure 1 are the mill, sump and hydrocyclone. The mill receives four streams: mined ore (MFS), water (MIW), steel balls (MFB) and underflow from the hydrocyclone. The ground ore in the mill mixes with the water to create a slurry. The fraction of the mill filled with charge is represented by JT. The slurry from the mill is usually discharged through an end-discharge-screen where the particle size of the discharged slurry is limited by the aperture size of the screen. The slurry in the sump is diluted with water (SFW) before it is pumped to the cyclone for classification. The volume of the slurry in the sump and the flow-rate of slurry pumped to the cyclone is represented by $SVOL$ and CFF respectively. The hydrocyclone is responsible for the separation of the in- and out-of-specification ore discharged from the sump. The in-specification particles of the slurry pass to the

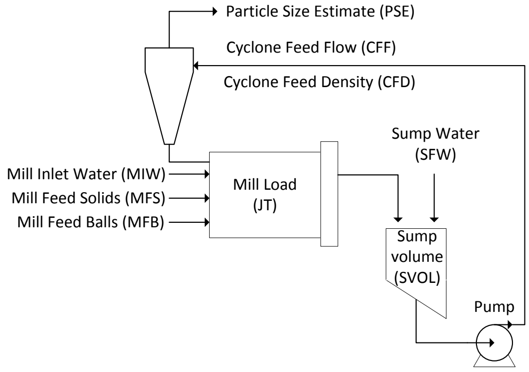

Fig. 1. A single-stage grinding mill circuit.

overflow of the hydrocyclone, while the out-of-specification particles pass to the underflow. This underflow is passed to the mill for further grinding. The overflow (PSE) contains the final product passed to downstream processes (Napier-Munn et al., 1999).

The model used to describe the circuit in Fig. 1 consists of four modules: a feeder, a semi-autogenous mill with an end-discharge screen, a sump and a hydrocyclone. All these modules can be described by the reduced complexity nonlinear model found in le Roux et al. (2013). The approach in the derivation of the model was to use as few fitted parameters as possible while still making the model produce responses that are reasonably accurate and in the right direction. The variables of the circuit are described in Table 1.

Table 1. Description of circuit variables

Manipulated Variables	
MIW	flow-rate of water to the mill [m^3/h]
MFS	feed-rate of ore to the mill [t/h]
MFB	feed-rate of steel balls to the mill [t/h]
SFW	flow-rate of water to the sump [m^3/h]
CFF	flow-rate of slurry to the classifier [m^3/h]
Controlled Variables	
JT	fraction of the mill filled [-]
$SVOL$	volume of slurry in sump [m^3]
PSE	product particle size estimate [-]

The model uses five states to represent the constituents of charge in the milling circuit. The states are rocks, solids, fines, balls and water. Rocks are ore too large to be discharged from the mill, whereas solids are ore that can be discharged from the mill. The solids consist of the sum of fine and coarse ore, where fine ore is smaller than the product specification size and coarse ore is larger than the product specification size. Balls and rocks are only found in the mill, as they are too large to pass through the apertures in the end-discharge screen.

For the equations, V denotes a flow-rate in m^3/h and X denotes the states of the model as volumes in m^3. Table 2 provides a description of the subscripts for V and X. The first subscript indicates the module considered, the second subscript specifies which of the five states are considered and in the case of flow-rates the final subscript shows if it is an inflow, overflow or underflow. The nomenclature for the model is shown in Table 3. The values for the parameters were taken from le Roux et al. (2013). The continuous time state-space description of the grinding mill circuit is shown below

$$\begin{aligned}
\dot{X}_{mw} &= MIW - \tfrac{V_V \varphi X_{mw} X_{mw}}{X_{ms}+X_{mw}} + V_{cwu} \\
\dot{X}_{ms} &= \tfrac{MFS}{D_S}(1-\alpha_r) - \tfrac{V_V \varphi X_{mw} X_{ms}}{X_{ms}+X_{mw}} + V_{csu} + \\
&\quad \tfrac{P_{mill}\varphi}{D_S \phi_r}\left(\tfrac{X_{mr}}{X_{mr}+X_{ms}}\right) \\
\dot{X}_{mf} &= \tfrac{MFS}{D_S}\alpha_f - \tfrac{V_V \varphi X_{mw} X_{mf}}{X_{ms}+X_{mw}} + V_{cfu} + \tfrac{P_{mill}}{D_S \phi_f} / \\
&\quad \left[1 + \alpha_{\phi_f}\left(\tfrac{X_{mw}+X_{mr}+X_{ms}+X_{mb}}{v_{mill}} - v_{P_{max}}\right)\right] \\
\dot{X}_{mr} &= \tfrac{MFS}{D_S}\alpha_r - \tfrac{P_{mill}\varphi}{D_S \phi_r}\left(\tfrac{X_{mr}}{X_{mr}+X_{ms}}\right) \\
\dot{X}_{mb} &= \tfrac{MFB}{D_B} - \tfrac{P_{mill}\varphi}{\phi_b}\left(\tfrac{X_{mb}}{D_S(X_{mr}+X_{ms})+D_B X_{mb}}\right) \\
\dot{X}_{sw} &= \tfrac{V_V \varphi X_{mw} X_{mw}}{X_{ms}+X_{mw}} - \tfrac{CFF X_{sw}}{X_{sw}+X_{ss}} + SFW \\
\dot{X}_{ss} &= \tfrac{V_V \varphi X_{mw} X_{ms}}{X_{ms}+X_{mw}} - \tfrac{CFF X_{ss}}{X_{sw}+X_{ss}} \\
\dot{X}_{sf} &= \tfrac{V_V \varphi X_{mw} X_{mf}}{X_{ms}+X_{mw}} - \tfrac{CFF X_{sf}}{X_{sw}+X_{ss}}
\end{aligned} \tag{29}$$

where X_{mw}, X_{ms}, X_{mf}, X_{mr} and X_{mb} are the volume of water, solids, fines, rocks and balls within the mill respectively, and X_{sw}, X_{ss} and X_{sf} are the volume of water, solids and fines within the sump respectively. The output equations are

$$\begin{aligned}
JT &= \tfrac{X_{mw}+X_{ms}+X_{mr}+X_{mb}}{v_{mill}} \\
SVOL &= X_{ss} + X_{sw} \\
PSE &= \tfrac{V_{cfo}}{V_{cso}}
\end{aligned} \tag{30}$$

where V_{cfo} and V_{cso} are the volumetric flow rate of fines and solids at the overflow of the cyclone respectively. The intermediate equations required in (29) relating the the mill are

$$\begin{aligned}
\varphi &= \left(1 - \left(\tfrac{1}{\varepsilon_{sv}} - 1\right)\tfrac{X_{ms}}{X_{mw}}\right)^{0.5} \\
P_{mill} &= P_{max}\{1 - \delta_{Pv} Z_x^2 - 2\chi_P \delta_{Pv} \delta_{Ps} Z_x Z_r - \\
&\quad \delta_{Ps} Z_r^2\} \cdot (\alpha_{speed})^{\alpha_P} \\
Z_x &= \tfrac{X_{mw}+X_{mr}+X_{ms}+X_{mb}}{v_{mill} v_{P_{max}}} - 1 \\
Z_r &= \tfrac{\varphi}{\varphi_{P_{max}}} - 1
\end{aligned} \tag{31}$$

where φ is an empirically defined rheology factor, P_{mill} is the mill power draw, Z_x is the effect of the charge within in the mill on the power draw, and Z_r is the effect of the rheology of the charge in the mill on the power draw. The intermediate equations required in (29) and (30) related to the cyclone are

$$\begin{aligned}
V_{ccu} &= \tfrac{CFF(X_{ss}-X_{sf})}{X_{sw}+X_{ss}}\left(1 - C_1 \exp\left(\tfrac{-CFF}{\varepsilon_c}\right)\right) \times \\
&\quad \left(1 - \left(\tfrac{X_{ss}}{C_2(X_{sw}+X_{ss})}\right)^{C_3}\right)\left(1 - \left(\tfrac{X_{sf}}{X_{ss}}\right)^{C_4}\right) \\
F_u &= 0.6 - \left(0.6 - \tfrac{X_{ss}}{X_{sw}+X_{ss}}\right)\exp\left(\tfrac{-V_{ccu}}{\alpha_{su}\varepsilon_c}\right) \\
V_{cwu} &= \tfrac{X_{sw}(V_{ccu}-F_u V_{ccu})}{F_u X_{sw}+F_u X_{sf}-X_{sf}} \\
V_{cfu} &= \tfrac{X_{sf}(V_{ccu}-F_u V_{ccu})}{F_u X_{sw}+F_u X_{sf}-X_{sf}} \\
V_{csu} &= V_{ccu} + \tfrac{X_{sf}(V_{ccu}-F_u V_{ccu})}{F_u X_{sw}+F_u X_{sf}-X_{sf}} \\
V_{cso} &= V_{sso} - V_{csu} \\
V_{cfo} &= V_{sfo} - V_{cfu}
\end{aligned} \tag{32}$$

where F_u represents the fraction of solids in the cyclone underflow, the flowrate of water, solids and fines at the underflow of the cyclone is V_{cwu}, V_{csu} and V_{cfu} respectively, and the flowrate of solids and fines at the sump outflow is V_{sso} and V_{cfo} respectively.

Table 2. Description of subscripts

Subscript	Description
$X_{\Delta -}$	f-feeder; m-mill; s-sump; c-cyclone
$X_{-\Delta}$	w-water; s-solids; c-coarse; f-fines; r-rocks; b-balls
$V_{--\Delta}$	i-inflow; o-outflow; u-underflow

Table 3. Feeder, mill and cyclone parameters

Parm	Value	Description
α_f	0.05	Fraction fines in the ore
α_r	0.47	Fraction rock in the ore
α_P	1.0	Fractional power reduction per fractional reduction from maximum mill speed
α_{ϕ_f}	0.01	Fractional change in kW/fines produced per change in fractional filling of mill
α_{speed}	0.71	Fraction of critical mill speed
α_{su}	0.87	Parameter related to fraction solids in underflow
C_1	0.6	Constant
C_2	0.7	Constant
C_3	4.0	Constant
C_4	4.0	Constant
δ_{P_s}	0.5	Power-change parameter for fraction solids in the mill
δ_{P_v}	0.5	Power-change parameter for volume of mill filled
D_B	7.85	Density of steel balls [t/m^3]
D_S	3.2	Density of feed ore [t/m^3]
ε_{sv}	0.6	Max fraction solids by volume of slurry at 0 slurry flow
ε_c	129	Parameter related to coarse split [m^3/h]
ϕ_b	90.0	Steel abrasion factor [kWh/t]
ϕ_f	29.5	Power needed per tonne of fines produced [kWh/t]
ϕ_r	6.00	Rock abrasion factor [kWh/t]
$\varphi_{P_{max}}$	0.57	Rheology factor for maximum mill power draw
P_{max}	1662	Maximum mill motor power draw [kW]
v_{mill}	100	Mill volume [m^3]
$v_{P_{max}}$	0.34	Fraction of mill volume filled for maximum power draw
V_V	84.0	Volumetric flow per "flowing volume" driving force [h^{-1}]
χ_P	0	Cross-term for maximum power draw

4. SIMULATION

The algorithm in this paper predicts the system output over a predefined horizon and calculates the input required to track the output setpoint over that horizon. To illustrate the effectiveness of this algorithm, this simulation shows how the controller calculates the input to maintain the desired output setpoint for the grinding mill circuit for 30 minutes using inputs MFS, SFW and CFF. Analogous to MPC, the prediction horizon and control horizon are both 30 minutes. (This paper does not show a receding horizon problem where the input trajectory is calculated at each sampling time, but rather only the first input trajectory calculation.)

The continuous time state-space description of the grinding mill circuit ((29) and (30)) is discretized with a sampling time of $T_s = 10$ s. The ball feed rate MFB is kept constant at 5.68 t^3/h and the inflow of water MIW is kept at a constant ratio of 7% of MFS. The initial states $\bar{X}_0$ are

$$\begin{aligned}&[X_{mw},\ X_{ms},\ X_{mf},\ X_{mr},\ X_{mb},\ X_{sw},\ X_{ss},\ X_{sf}]^T = \\ &[4.85,\ \ 4.90,\ \ 1.09,\ \ 1.82,\ \ 8.51,\ \ 4.11,\ \ 1.88,\ \ 0.42]^T\end{aligned} \tag{33}$$

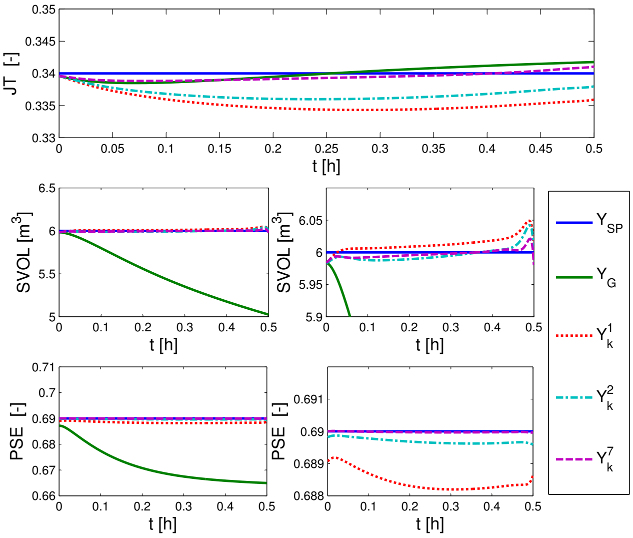

Fig. 2. Top: fraction of the mill filled JT. Middle: volume of slurry in sump $SVOL$ (normal - left; zoomed - right). Bottom: fraction of particles smaller than specification size PSE (normal - left; zoomed - right). Legend: Y_{SP} is the desired setpoint, Y_G is the output trajectory from the estimated input, Y_k^1, Y_k^2 and Y_k^7 are the outputs after 1, 2 and 7 iterations of the algorithm.

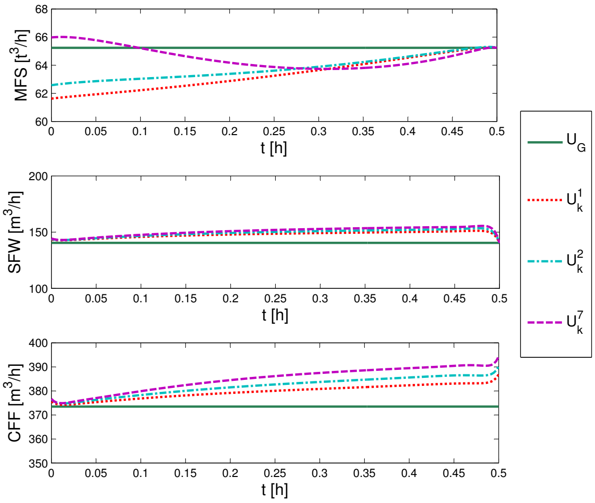

Fig. 3. Top: mill solids feed rate MFS. Middle: sump water feed rate SFW. Bottom: cyclone feed flow rate CFF. Legend: U_G is the estimated input trajectory, U_k^1, U_k^2 and U_k^7 are the inputs calculated after 1, 2 and 7 iterations of the algorithm.

The estimated input trajectory $\bar{U}_G$ is a constant input of

$$\bar{U}_G = [MFS,\ SFW,\ CFF]^T = [65.2,\ 140.5,\ 373]^T \tag{34}$$

The desired constant output setpoint $\bar{Y}_{SP}$ is

$$\bar{Y}_{SP} = [JT,\ SVOL,\ PSE]^T = [0.34,\ 6.00,\ 0.69]^T \tag{35}$$

The values above were taken from the plant survey done in le Roux et al. (2013).

The particle size estimate PSE is regarded as the most important output variable to control, since this variable determines the economic efficiency of the milling circuit.

Therefore, the Q weighting matrix for the output variables was determined such that a 5% deviation of JT from setpoint will produce an error in the cost function equal to a 1% deviation from setpoint for PSE and equal to a 10% change in $SVOL$ from setpoint, i.e.

$$Q_1\left(10\%JT_{SP}\right)^2 = Q_2\left(5\%SVOL_{SP}\right)^2 = Q_3\left(1\%PSE_{SP}\right)^2$$

Finally, the output weighting matrix is defined as

$$Q = 10^3\ \mathrm{diag}([1.25,\ 0.001,\ 7.56])$$

The R weighting matrix for the input variables was determined such that 1% changes of half the ranges of CFF, MFS and SFW will produce the same error in the cost function. The R matrix was scaled to produce 1% of the error compared to the Q matrix, i.e.

$$100R_1\left(\frac{1\%MFS_{range}}{2}\right)^2 = Q_1\left(10\%JT_{SP}\right)^2$$

and

$$R_1\left(\frac{1\%MFS_{range}}{2}\right)^2 = R_2\left(\frac{1\%SFW_{range}}{2}\right)^2 = R_3\left(\frac{1\%CFF_{range}}{2}\right)^2$$

The ranges of the input variables are given below.

$$\begin{aligned}[MFS,\ SFW,\ CFF]_{min}{}^T &= [0,\ 0,\ 100]^T\\ [MFS,\ SFW,\ CFF]_{max}{}^T &= [200,\ 400,\ 500]^T\end{aligned} \tag{36}$$

The input weighting matrix is $R = 10^{-4}\ \mathrm{diag}([36,\ 9,\ 9])$.

Termination of the algorithm occurs when either of the conditions below are met

$$\begin{aligned}\frac{\|Y_k^i - Y_k^*\|_2}{\|Y_k^*\|_2} &< [0.001, 0.001, 0.001]^T\\ \frac{\|U_k^{i+1} - U_k^*\|_\infty}{\|U_k^*\|} &< [0.01, 0.01, 0.01]^T\end{aligned} \tag{37}$$

If these conditions are not met after a maximum iteration count specified by the user, the algorithm also terminates.

The results are shown in Figs. 2 and 3. The algorithm met the conditions of (37) after $i = 7$ iterations. As seen from Fig. 2, the desired value for output PSE is maintained for the entire duration of the simulation. Even after only 1 iteration of the algorithm, PSE is already very close to the desired trajectory. Outputs JT and $SVOL$ deviate slightly from the desired trajectory, but allows PSE - the most important output - to follow the desired trajectory closely.

If there is no control action in the circuit and the estimated input trajectory U_G is maintained throughout, PSE and $SVOL$ will decrease while JT increases after 0.5 h. To maintain PSE at setpoint, CFF increases. Because this will decrease $SVOL$, there is an increase in SFW. But a higher CFF causes a higher underflow to the mill and an increase in JT. Thus, MFS is varied to maintain JT close to setpoint since a slightly lower JT assist to improve the grind efficiency (PSE).

5. DISCUSSION AND CONCLUSION

Simulation results illustrate that the suboptimal control technique presented in this paper is capable of finding a suboptimal input trajectory after 7 iterations to maintain the outputs of a grinding mill circuit at their desired setpoints. The most important output PSE is controlled at the desired setpoint with almost no variation from setpoint. The slight deviations from setpoint for both JT and $SVOL$ are regarded as minimal and have almost no effect on the grinding efficiency of the mill. Both Figs. 2 and 3 show how quickly the algorithm converges to a solution, even though the input has to be determined for a relatively large horizon. Even after only 2 iterations, both PSE and $SVOL$ closely follow the desired trajectory.

Because the minimisation of the cost function in nonlinear MPC can be time consuming for large and complex systems, the suboptimal control technique presented here has the potential to significantly reduce the computational time. This technique could enable more opportunities for on-line application of suboptimal control in the comminution industry. This can be illustrated in future studies by means of a comparison between this technique and nonlinear MPC applied to a complex system such as the grinding mill circuit.

REFERENCES

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. (2000). The Explicit Solution of Model Predictive Control via Multiparametric Quadratic Programming. In *American Control Conference*, 872–876. Chicago, USA.

Chen, X., Li, Q., and Fei, S. (2008). Constrained model predictive control in ball mill grinding process. *Powder Technol.*, 186(1), 31 – 39.

Coetzee, L.C., Craig, I.K., and Kerrigan, E.C. (2010). Robust nonlinear model predictive control of a run-of-mine ore milling circuit. *IEEE Trans. Control Syst. Technol.*, 18(1), 222–229.

Halbe, O., Raja, R.G., and Padhi, R. (2013). Robust re-entry guidance of a reusable launch vehicle using model predictive static programming. *J. Guidance Control Dynamics.* doi:10.2514/1.61615.

Hodouin, D. (2011). Methods for automatic control, observation and optimization in mineral processing plants. *J. Process Control*, 21(2), 211–225.

Kumar, P. and Padhi, R. (2014). Extension of Model Predictive Static Programming for Reference Command Tracking. In *Third Int. Conf. Advances in Control and Optimization of Dynamical Systems.* Kanpur, India. (Submitted).

le Roux, J.D., Craig, I.K., Hulbert, D.G., and Hinde, A.L. (2013). Analysis and validation of a run-of-mine ore grinding mill circuit model for process control. *Minerals Eng.*, 43-44, 121–134.

Napier-Munn, T.J., Morrell, S., Morrison, R.D., and Kojovic, T. (1999). *Mineral Communition Circuits: Their Operation and Optimisation.* JKMRC Monograph Series in Mining and Mineral Processing, 2nd edition.

Oza, H.B. and Padhi, R. (2012). Impact-angle-constrained suboptimal model predictive static programming guidance of air-to-ground missiles. *J. Guidance Control Dynamics*, 35(1), 153–164.

Padhi, R. and Kothari, M. (2009). Model predictive static programming: A computationally efficient technique for suboptimal control design. *Int. J. Innovative Computing Information Control*, 5(2), 399–411.

Wang, Y. and Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Trans. Control Syst. Technol.*, 18(2), 267–278.