Overload Detection in Semi-Autogenous Grinding: A Nonlinear Process Monitoring Approach

K.S. McClure*, R.B. Gopaluni**

* Spartan Controls Ltd, Vancouver, BC, Canada (e-mail: mcclure.ken@spartancontrols.com).
** Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC, Canada (e-mail: bhushan.gopaluni@ubc.ca)

Abstract: Detecting the onset of overloading in a semi-autogenous grinding (SAG) mill is a challenging task for operators to perform due to the complex and nonlinear nature of an overload. To detect an overload, operators must simultaneously monitor the correlations between several measurements of the SAG process. However, overloading often goes unnoticed at its early stages because the subtle changes in the correlations between measurements are difficult for an operator to observe. In addition, linear process monitoring techniques such as principal component analysis (PCA) provide inconsistent results with overload detection because of the process nonlinearity. Recently, locally linear embedding (LLE) with a linear classifier has been proposed to detect the early onset of an overload in a SAG mill. In this paper, we compare the suitability of LLE to detect the early onset of an overload against kernel PCA and support vector machines.

Keywords: process control, statistical analysis, process monitoring, nonlinearity, non-parametric

1. INTRODUCTION

In a mineral processing facility, grinding is one of the most fundamental unit operations, and a semi-autogenous grinding (SAG) mill is often used as the primary grinding stage. In a SAG mill, the ore itself along with steel grinding media is used to grind ore from 200 mm down to 0.1 mm as a means to liberate the valuable minerals (Wills & Napier-Munn, 2006). In these mills, total charge volume occupies approximately 35% of the mill interior, and the charge is lifted by lifter bars along the interior of the mill shell as the mill rotates.

The primary objective for a SAG mill is to maximize the throughput of fresh ore. Fresh feed rate, however, is constrained by the total volume occupied by the charge inside the mill (Craig & Macleod, 1995). If the charge volume exceeds a certain threshold, the mill begins to overload and grinding efficiency decreases in a positive feedback mechanism. Since it is currently infeasible to accurately measure the charge volume inside a SAG mill, operators are challenged with the task of monitoring the correlations of several variables at once to indirectly infer if the mill is beginning to overload. Moreover, the changes in measured variables and their correlations are subtle, thus a SAG mill often severely overloads by the time an operator becomes aware.

The physics of an overload are complex; Fig. 1 illustrates the three possible mechanisms of an overload. The first mechanism is the volumetric overload: a volumetric overload occurs when the charge volume is too high and is caused by overfeeding the mill for the particular steel charge volume and dynamic feed characteristics such as the particle size, hardness, and composition. Additionally, any disturbance that causes the grinding efficiency in the mill to decrease may also lead to volumetric overloading due to the accumulation of charge in the mill. As accumulation occurs, volume occupied by the charge increases such that the total distance ore can fall under gravity, known as cataracting, decreases.

![Fig. 1. Three mechanisms of overloading: volumetric overloading reduces potential energy; slurry pooling dampens impact energy; and freewheeling reduces lifting efficiency.](image)

Once a SAG mill has begun to overload, ore will rapidly accumulate until the charge level reaches the center of the mill and slurry spills out of the feed and discharge chutes.
Depending on the charge density, the bearing pressures may not climb high enough to raise an alarm. Therefore, operators find it difficult to detect volumetric overloading early since this type of overloading is a function of the unknown volume rather than the known mass of the charge. One possible solution is to observe the shift in the center of gravity of the mill. Load distribution may be measured using the pressure readings from the bearing pads supporting the mill. However, the distribution of the charge in the mill is often subtle in the early stages of an overload and also unique for each different overload mechanism.

The second overload mechanism is slurry pooling. Slurry pooling occurs when an excess of slurry accumulates inside the mill and results in the formation of a slurry pool at the toe of the charge where cataracting ore impacts. The pool acts as a strong damper to the charge’s impact energy; as a consequence, grinding efficiency is hampered. Fig. 2 illustrates the uneven distribution of slurry inside the mill during the evolution of pooling (Latchireddi & Morrell, 2003). The slurry physically fills up the interstitial spaces in the solid charge as the material is lifted up with the rotation of the mill. When all the interstitial space is occupied by slurry, a pool forms at the toe of the charge. The resulting reduction in grinding efficiency may lead to the mill overloading if corrective action to the feed is not taken. To detect this mechanism of overloading, operators rely on observing the deterioration of the positive correlation between bearing pressure and motor power. However, mill power does not have a consistently repeatable behaviour during the onset of this phenomenon due to the unique shift in the mill’s center of gravity from both slurry pool formation and subsequent charge accumulation. Consequently, the characteristic inverse correlation between power and bearing pressure of an overload only becomes obvious to operators once the mill is severely overloaded from slurry pooling.

Owing to the difficulty for operators to detect the onset of overloading, Ko & Shang (2011) proposed using principal component analysis (PCA) as a linear process monitoring technique to detect the onset of an overload. Due to its limitations as a linear method, their study does not identify which abnormalities are overload events. Likewise, the technique has found limited success in industrial applications. This limited success is likely because it is challenging to select process measurements with linear characteristics between nominal and overloaded operation to satisfy the main assumptions underlying PCA.

To overcome the drawbacks of PCA, we compare three nonlinear process monitoring techniques for overload detection in this paper: Kernel Principal Component Analysis (KPCA), Support Vector Machines (SVM), and Locally Linear Embedding (LLE). KPCA uses a nonlinear kernel to first map data to a high dimensional feature space. In this space, the correlations are more likely to be linear such that linear PCA can then be performed (Schölkopf, Smola & Müller, 1998). Conversely, SVM discovers an optimal linear classifier in the form of a hyperplane which maximizes the margin between data points on either side of the hyperplane (Tong & Koller, 2002). This classifies plant data into two categories: “faulry” and “non-fauly.” Finally, LLE maps locally linear patches of a nonlinear high-dimensional data-set into a low-dimensional feature space where classification becomes more straightforward and accurate (McClure, 2013).

2. PROCESS MONITORING TECHNIQUES

This section presents the three nonlinear process monitoring techniques compared in this study to detect a SAG overload: KPCA, SVM, and LLE.

2.1 Kernel Principal Component Analysis

KPCA is an extension of PCA that is applicable to nonlinear systems. As a fault detection technique, it is superior to previous methods, such as neural networks, because it does not require the solution of a nonlinear optimization problem. KPCA uses integral operators and nonlinear kernel functions to map an input space to a feature space. Next, it computes principal components in the feature space (Lee et al., 2004). Due to the wide variety of nonlinear kernel functions available, KPCA can handle a wide range of process nonlinearities. However, the main disadvantage of KPCA is its high computational requirement. For example, using KPCA for large systems with many samples is unreasonable because mapping new data into the feature space is too time-consuming. Moreover, the high-dimensional feature space may not be physically interpretable compared to those generated by other methods such as linear PCA.

According to Cover’s theorem, the nonlinear structure of data is likely to be linear after being nonlinearly mapped to a higher dimensional feature space (Haykin, 1999). In KPCA, a nonlinear set of data \( x_k \in \mathbb{R}^N, k=1,...,N \) where \( m \) is the number of variables, \( N \) is the number of observations, and \( x_k \) has zero mean, is mapped to the feature space using a kernel function. Afterwards, the eigenvalue problem of the
covariance matrix of the data-set is solved in the feature space to yield the principal components (PCs) (Lee et al., 2004). Correspondingly, the covariance matrix in the feature space is:

\[ C = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i) \phi(x_i)^T, \]

where \( \phi(\cdot) \) is a nonlinear kernel mapping function that projects the data to the arbitrarily high-dimensional feature space. The PC’s can be derived from the eigenvalue problem:

\[ \lambda v = Cv, \]

with eigenvalues \( \lambda \geq 0 \), where the largest \( \lambda \) is the first PC and \( v \) are the eigenvectors. The detailed solution to equation (2) is given in Lee et al. (2004) and Schölkopf, Smola & Müller (1998).

2.2 Support Vector Machines

A viable alternative to KPCA is SVMs. SVMs originated as a machine learning technique to classify data having two different pre-defined categories. Therefore, for the specific case of fault detection, SVMs can be used to classify data in a one-class problem as either normal or faulty (Mahadevan & Shah, 2009). Fig. 3 shows the application of SVMs on a two-dimensional system of arbitrary process variables \( y_1 \) and \( y_2 \).

SVMs discover an optimal linear classifier that can be geometrically interpreted as a hyperplane that maximizes the margin between data points on either side of the hyperplane. In the case of nonlinear classification, a kernel transformation is used to map the data to a higher-dimensional feature space. Here, the separating hyperplane creates a nonlinear boundary when re-mapped to a lower dimensional space. Unsupervised versions of SVMs can also be applied: the advantage to this approach is that only nominal operating data is required as opposed to requiring data from both nominal and abnormal classes. Despite its advantages, the main disadvantage for the nonlinear boundary approach is the computational requirements for use online in an industrial controller.

Fig. 3. SVM trained with two classes. Data samples located on the margins are known as support vectors.

For binary classification, SVM’s are vectors on a hyperplane to separate the training data-set into two classes by a maximum margin. Data points nearest to the hyperplane are called support vectors, and these support vectors are determined by projecting the training data into a higher-dimensional feature space where a linear classifier can be constructed. SVM generates a set of classifiers of the form:

\[ f(x) = \left( \sum_{i=1}^{N} a_i K(x_i, x) \right), \]

where \( a_i \) are the coefficients defining the maximal margin hyperplane that are determined when training the SVM model, and \( K \) is the kernel operator that is used to project the training data to the feature space. Various kernel functions are derived in detail by Tong & Koller (2002).

2.3 Locally Linear Embedding

LLE is a novel feature extraction technique that requires minimal computational effort. It was proposed by Roweis & Saul (2000) as a nonlinear data mapping technique for image processing. However, no documented version of LLE has been applied in an industrial control system.

In LLE, data is mapped from a high-dimensional space to a low dimensional space. Fig. 4 illustrates the three fundamental steps of LLE performed offline from the industrial controller: first, select the nearest neighbours for each data point of the training data; second, reconstruct each data-point from its neighbours; finally, map the data to a reduced-dimension feature space (Roweis & Saul, 2000). For the second step, the most important assumption is that each data point and its neighbours lie on a locally linear manifold (Li & Zhang, 2011).

Fig. 4. The three steps of the LLE algorithm.

Step 1. Select neighbors.

The first step is to identify the \( K \)-nearest neighbors of \( X \), where \( X \) has been standardized to zero mean and unit variance or scaled from 0% to 100%. To find the neighbors of the \( i^{th} \) observation in \( X \), \( x_i \in \mathbb{R}^{D+1} \), the Euclidean distance between \( x_i \) and every other observation in \( X \) needs to be calculated. The \( K \) observations having the shortest distance from \( x_i \) are the neighbors, \( x_i \).

The distance between \( x_i \) and any other observation in \( X \), \( x_j \), is calculated from the expression:

\[ d(x_i, x_j) = \sqrt{\sum_{k=1}^{D} (x_{ik} - x_{jk})^2}. \]
\[ ||x_i - x_j||_2, \] (4)

where \( || \cdot || \) denotes the \( l^2 \) norm.

**Step 2.** Reconstruct each data point from its neighbors.

The second step is to reconstruct each data point from its neighbors by minimizing the cost function:

\[
\begin{align*}
\text{minimize} & \quad \epsilon(W) = \sum_{i=1}^{n} ||x_i - \sum_{j=1}^{n} W_{ij}x_j||_2^2, \\
\text{subject to} & \quad W_{ij} = 0, \text{ if } x_j \notin \{x_{n1}, x_{n2}, ..., x_{nK}\}, \\
\sum_{j=1}^{n} W_{ij} = 1.
\end{align*}
\] (5)

Here, the weights \( W_{ij} \in \mathbb{R} \) gives the magnitude that the \( j^{th} \) observation in \( X \) contributes to the \( i^{th} \) reconstruction. Equation (6) constrains \( W_{ij} \) to zero if \( x_j \) do not belong to the set of neighbors for the \( i^{th} \) reconstruction. Equation (5) minimizes the sum of the squared distances between all data points and their reconstructions. Equations (5) through (7) represent a constrained least-squares problem to solve for the optimal reconstruction weights (Roweis & Saul, 2000).

**Step 3.** Map to embedded coordinates.

The final step is to map each high-dimensional observation, \( x_n \), to a low-dimensional feature vector, \( y_n \in \mathbb{R}^{d \times 1} \). Each \( y_n \) represents the \( d \)-dimensional coordinates of the low-dimensional manifold. To solve for \( y_n \), the embedding cost function is minimized:

\[
\begin{align*}
\text{minimize} & \quad \Phi(y) = \sum_{i=1}^{n} ||y_i - \sum_{j=1}^{n} W_{ij}y_j||_2^2, \\
\text{subject to} & \quad \sum_{j=1}^{n} y_i = 0, \\
\frac{1}{n} \sum_{j=1}^{n} y_{ij} y_j^T &= I,
\end{align*}
\] (8)

where \( I \) is the \( d \times d \) identity matrix. Equation (9) enforces a zero mean constraint on embedding coordinates, and equation (10) avoids degenerate solutions by enforcing a unit covariance constraint on the embedding coordinates. Roweis & Saul (2000) demonstrate that the constrained quadratic program in equations (8) through (10) is well-posed and can be reformulated to be efficiently solved as a sparse \( n \times n \) eigenvalue problem.

In practice, it is desirable to select \( d \leq 3 \) so that the embedded feature data can be graphically inspected to reveal which coordinates have the best separation between nominal and abnormal process data. Correspondingly, only the coordinates showing strong separation are required for process monitoring. Alternatively, de Ridder and Duin (2002) propose a method to automatically determine \( d \) by ensuring the amount of original information retained by the locally linear neighborhoods exceeds a desired threshold.

**LLE Projection**

Since LLE is a non-parametric method, there is no formula to map new observations to the feature space for online process monitoring. To overcome this, Li & Zhang (2011) adapted LLE to project new data to this space using the least-squares technique in a method known as LLE projection. LLE projection is performed for each new test observation online in a controller, and it has three steps similar to LLE: first, the training-set’s nearest neighbors to a new test observation are found; second, a linear mapping matrix is computed from the nearest neighbors; finally, the test observation’s embedded coordinate is calculated from the linear mapping matrix.

**Step 1.** Select neighbors.

For a new high-dimensional sample, \( x_{new} \in \mathbb{R}^{D \times 1} \), the first step is to find its \( K \) neighbors, \( X_n \) in the training data-set \( X \) by a Euclidean distance measurement, where:

\[
X_n = [x_{n1}, x_{n2}, ..., x_{nK}] \in \mathbb{R}^{D \times K}. \] (11)

**Step 2.** Find a mapping matrix.

Compute the mapping matrix, \( A \), from the training data that satisfies:

\[
Y_n = AX_n, \] (12)

in the least squares sense, where:

\[
Y_n = [y_{n1}, y_{n2}, ..., y_{nK}] \in \mathbb{R}^{d \times K}. \] (13)

and \( Y_n \) is the neighbourhood embedded coordinate matrix of the feature data corresponding to the observations of \( X_n \). The basis vectors of \( A, a_p \), are solved by performing linear least squares regression on equation (12). Furthermore, this equation can be solved online in an industrial control system using the quadratic solving feature of model predictive control (McClure et al., 2013).

**Step 3.** Compute the embedding coordinate.

The final step is to use the mapping matrix \( A \) to calculate the embedding coordinate, \( y_{new} \in \mathbb{R}^{d \times 1} \), for a new observation \( x_{new} \): 

\[
y_{new} = AX_{new}. \] (15)

**Classification**

Data mapped to the feature space by LLE need to be classified, and often a simple linear classification technique such as linear discriminant analysis (LDA) suffices (McClure et al., 2013).

To perform LDA, the embedded training data, \( Y \), is first split into two datasets \( Y_1 \) and \( Y_2 \) based on their class distinction as either nominal or abnormal operation. Then, using standard statistical techniques as outlined by Johnson & Wichern (2007), the pooled covariance, \( S_{pooled} \), and mean vectors, \( \bar{y}_1 \) and \( \bar{y}_2 \), of \( Y_1 \) and \( Y_2 \) are determined. With this information, a linear discriminator can be constructed from the following equation:

\[
\begin{align*}
\text{class} &= \{\pi_1 \quad \bar{a}^T y_i - \bar{m} \geq C \}, \\
&\{\pi_2 \quad \bar{a}^T y_i - \bar{m} < C, \}
\end{align*}
\] (16)

where:

\[
\bar{a} = (\bar{y}_1 - \bar{y}_2)^T S_{pooled}^{-1}. \] (17)
\[ \hat{m} = \frac{1}{2} (\overline{y}_1 - \overline{y}_2) \hat{S}_{pooled}^{-1} (\overline{y}_1 - \overline{y}_2). \] (18)

Here, \( \hat{a} \) is the sample coefficient vector, \( \hat{m} \) is the estimated midpoint between the two classes, and \( C \) is the cost parameter that accounts for the probability of misclassification and the cost of misclassification; \( C \) is assumed to be zero for the case of a SAG overload. Any new observation, \( y_{new} \), will fall on either side of the line given in Equation (16), thus classifying the observation as either class \( \pi_1 \) (nominal) or \( \pi_2 \) (overloaded).

3. RESULTS AND DISCUSSION

Training and test data-sets sampled at two minutes were acquired from a mining company in British Columbia, Canada, for two separate overload events in February and March, 2012. This data represents a challenging test for the process monitoring methods because the overload event for the training data-set in February was during typical operating conditions; whereas, the overload event in March was the consequence of a substantially over-estimated steel ball charge that lead to poor grinding efficiency and lower than normal bearing pressures.

Ten variables were considered for analysis by the process monitoring methods and are described in detail in McClure (2013): bearing pressure; motor power; bearing vibration; sound level, percentage feed size > 2 inches; charge centre of gravity; estimated discharge flow; recycle rate; specific energy; rate of change of bearing pressure and motor power. All variables were scaled from 0 to 1 such that the relative standard deviations were of the same magnitude.

Two test metrics are used to evaluate and compare the performance of the three process monitoring methods: sensitivity and specificity. Sensitivity and specificity are defined by the following expressions:

\[
\text{Sensitivity} = \frac{\# \text{ of true overload}}{\# \text{ of true overload + } \# \text{ of false nominal}}. \quad (19)
\]

\[
\text{Specificity} = \frac{\# \text{ of true nominal}}{\# \text{ of true nominal} + \# \text{ of false overload}}. \quad (20)
\]

where true overload is data known to be overload data and are classified correctly; false overload is data known to be nominal and were classified incorrectly as overloaded; true nominal is data known to be nominal and were classified correctly; false nominal is data known to be nominal but incorrectly classified.

3.1 Kernel Principal Component Analysis (KPCA)

Three kernel functions were explored for use in KPCA: polynomial of order 2 and 3, sigmoid, and radial basis function (RBF). Two test statistics, given in detail by Lee et al. (2004), were utilized: the Hoteling \( T^2 \) test statistic, and squared prediction error.

Results for the test set showed that only the radial basis function was capable of classifying the overload data using the \( T^2 \) test statistic. However, KPCA demonstrated to be unreliable for industrial use with a sensitivity of 5.9% and a specificity of 84.3%. The low sensitivity indicates that most of the overload data was incorrectly classified, and the high specificity indicates that most of the nominal data was correctly classified.

3.2 Support Vector Machines (SVM)

Three kernel functions were tested when training the SVM model: linear, 2nd and 3rd order polynomials, and RBF. Of these three kernels, only RBF converged within a maximum of 15,000 iterations. Using 3-fold cross-validation and a grid search of \( \alpha \) from 0.1 to 150, we found that SVM was unable to detect any of the overload data in the test set resulting in sensitivity = 0% and specificity = 100%. SVM may have performed poorly because the nonlinear structure of the data is not well-represented by the RBF kernel, and investigation into different kernel functions on data extracted from other mine sites is needed to conclude on the suitability of SVM for SAG overload detection.

3.3 Locally Linear Embedding (LLE)

From 3-fold cross validation and a grid search of \( K \), it was found that \( K = 26 \) nearest neighbours gave the best results with sensitivity = 90.8% and specificity = 93.5%. These metrics indicate that most data was correctly classified and nearly in equal proportion between the nominal and overload classes.

Fig. 5 shows the true overload data within the regions depicted by the red lines, with two of the ten variables shown for the March test data-set shown in the upper subplot. The LLE output in the lower subplot shows that the LLE output, the Euclidean distance that the data-points are located away from the linear classifier, correlates strongly with the true overload data. Here, an LLE output < 0 corresponds to data classified as being overloaded.

4. CONCLUSIONS

In this paper, we demonstrated that LLE was more successful at detecting the onset of a SAG overload than KPCA or SVM. With sensitivity and specificity greater than 90%, LLE has demonstrated its suitability as a nonlinear process.
monitoring technique for SAG mill overloading. However, only the most common kernel functions for KPCA and SVM were explored, and it is possible that these methods would have similar performance to LLE if a kernel function could be found that projects the data with more linearity in the feature space.

ACKNOWLEDGEMENTS
This work is financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Spartan Controls, Burnaby, Canada.

REFERENCES