

# Combined neural network and particle filter state estimation with application to a run-of-mine ore mill

M.A. Naidoo\* L.E. Olivier\* I.K. Craig\*

\* *Department of Electrical, Electronic and Computer Engineering,  
University of Pretoria, Pretoria, South Africa (e-mail:  
icraig@postino.up.ac.za).*

---

**Abstract:** A run-of-mine (ROM) ore milling circuit poses many difficulties in terms of measuring process variables and determining accurate models. Control of the ROM circuit is therefore not a trivial task to achieve. An example of a ROM circuit model with reduced complexity that works well for control purposes is discussed. The mill model is discussed in detail, as this model is used for state estimation. A neural network is trained with three disturbance parameters and used to estimate the internal states of the mill, and the results are compared with those of particle filter implementation. A novel combined neural network and particle filter state estimator is presented. The estimation performance of the neural network is promising when the disturbance magnitude used is smaller than that used to train the network.

*Keywords:* neural network, particle filter, run-of-mine ore mill, state estimation

---

## 1. INTRODUCTION

A run-of-mine (ROM) ore milling circuit is a process that is difficult to control because of significant model uncertainties, large unmeasured disturbances and process variables that are difficult to measure (Olivier et al. (2012)). This motivates the investigation of a neural network for measurement purposes. When designing a controller that requires full state feedback, the internal states of the mill must be accurately measured to achieve good control of this complex system. For example, model predictive control (MPC) requires full state feedback, which is difficult to achieve, in this case because of model inaccuracies and lack of on-line measurements. This paper investigates the use of a neural network to estimate the states of the mill in a ROM ore grinding circuit. It builds on the work of Olivier et al. (2012) and is aimed at achieving more accurate closed-loop control. The internal states of a grinding mill model (Le Roux et al. (2013)) are estimated using a neural network, and results are compared to those achieved when using a particle filter. A new combined method of state estimation using both a neural network and particle filter is also presented.

The application of robust non-linear MPC to a ROM ore milling circuit was presented by Coetzee et al. (2010). The controller described by Coetzee et al. (2010) requires full state feedback, an issue that is partially addressed in this paper and by Olivier et al. (2012). Neural networks have previously been used to aid in the control of mineral grinding circuits. In a recent publication, a radial basis function neural network was used to successfully predict the in-mill slurry density and ball load volume in a ball-milling system (Makokha and Moys (2012)). These variables are key milling process variables that have hitherto been difficult or expensive to measure. Flament et al.

(1993) uses a neural network as a direct neural controller and an inverse network controller, which is successful, provided an offset correction scheme is present. Stange (1993) builds on this work and states that neural networks can be used as adaptive predictors.

A recent literature survey on the control of grinding mill circuits is summarized by Craig (2012). The survey shows that some multivariable control methods have fairly recently been implemented in an industry where single-loop PID controllers dominate (Wei and Craig (2009)). In particular, an MPC implementation on a grinding circuit was first reported in the literature as late as 2007 (Chen et al. (2007)). Compare this to the ubiquitous nature of MPC in for example the petrochemical industry (Craig et al. (2011)); there is great potential for increasing the number of MPC implementations on mineral processing plants. The estimation technique presented in this paper could help MPC have a sufficient impact in mineral processing control.

## 2. RUN-OF-MINE ORE MILLING CIRCUIT

### 2.1 Description of the run-of-mine ore milling circuit

The goal of minerals processing is to convert raw ore to a final product which contains a higher concentration of the most valuable minerals. The ROM circuit is the focus of this study and is shown in Fig. 1. This circuit forms part of the minerals liberation process (Hodouin (2011)). The major disturbances affecting the ROM circuit are due to the variation in feed size, grindability, model-plant mismatch and variables that are difficult to measure (Olivier et al. (2012)).

A brief description of the process is provided here, similar to that given by Olivier et al. (2012).

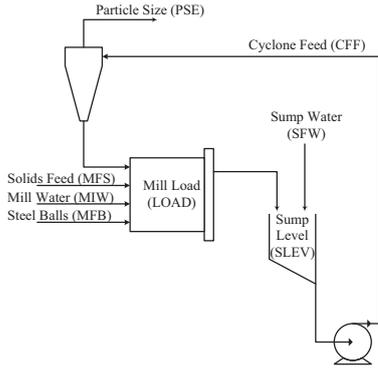


Fig. 1. Run-of-mine circuit. From Olivier et al. (2012)

Ore containing some valuable mineral (such as those containing copper, iron, platinum or gold) is fed into the mill where it is ground fine using rocks and steel balls as the grinding medium. The mill discharges into a sump where the slurry is diluted with water to achieve the correct density before it is pumped to a hydrocyclone. The cyclone separates the coarse and fine particles, with the fine particles leaving the circuit as product whilst the coarse particles are recycled back into the mill for further grinding. Ore is fed into the milling circuit at about 100 t/h, and leaves the circuit when its size is about 80% smaller than 75  $\mu\text{m}$ . A relationship between residue (valuable mineral not recovered by the downstream upgrading process) and particle size was derived by Craig et al. (1992) and the results show that a finer grind results in better recovery; however, this results in the throughput decreasing (Bauer and Craig (2008)), which could increase the overall operating cost.

Manipulated variables include the solids feed rate (MFS), the flow rate of the sump water (SFW), and the cyclone feed flow rate (CFF). Other inputs to the mill are the mill inlet water (MIW), steel balls (MFB), and the cyclone underflow. The MIW is calculated as a ratio to the solids feed and MFB is kept constant as discussed by Olivier et al. (2012). All operating points in this description can be found in Coetzee et al. (2010).

A typical mill is 9 m long and has a diameter of 5 m (Stanley (1987)). The mill is supported by pressurized-oil circumference bearings and operates at 90% of its critical speed (Coetzee et al. (2010)). Table 1 shows the constraints and operating values for the manipulated variables (MVs) and controlled variables (CVs). If the upper or lower bounds are not adhered to the controller is unsuccessful, as these bounds are due to physical constraints and cannot be violated.

### 2.2 Objectives of the run-of-mine ore milling circuit

The two major objectives for implementing control of the ROM ore milling circuit are to stabilize the process as well as optimize the economic performance of the process (Craig and MacLeod (1995)). Craig and MacLeod (1995) define a set of sub-objectives that contribute to the main objectives. The sub-objectives for the milling circuit performance are: (i) to improve the quality of the product either by increasing grind fineness or by decreasing variations in product size, (ii) to maximize throughput,

Table 1. Constraints and operating points. From Olivier et al. (2012)

Variable	Min	Max	Nom	Description
<b>MVs</b>				
MIW	0	100	33.33	Flow rate of water to the mill [ $\text{m}^3/\text{h}$ ]
MFS	0	200	100	Flow rate of solids to the mill [ $\text{t}/\text{h}$ ]
MFB	0	4	2	Flow rate of steel balls to the mill [ $\text{t}/\text{h}$ ]
CFF	400	500	442	Flow rate of slurry to the cyclone [ $\text{m}^3/\text{h}$ ]
SFW	0	400	267	Flow rate of water to the sump [ $\text{m}^3/\text{h}$ ]
<b>CVs</b>				
PSE	60	90	80	Product particle size [% < 75 $\mu\text{m}$ ]
LOAD	30	50	45	Total charge of the mill [%]
SLEV	2	37.5	30	Level of the sump $\text{m}^3$
<b>States</b>				
$X_{mw}$	0	50	8.53	Holdup of water in the mill [ $\text{m}^3$ ]
$X_{ms}$	0	50	9.47	Holdup of solid ore in the mill [ $\text{m}^3$ ]
$X_{mf}$	0	50	3.54	Holdup of fine ore in the mill [ $\text{m}^3$ ]
$X_{mr}$	0	50	20.25	Holdup of rocks in the mill [ $\text{m}^3$ ]
$X_{mb}$	0	20	6.75	Holdup of steel balls in the mill [ $\text{m}^3$ ]
<b>Parameters</b>				
$\alpha_f$	0.05	0.15	0.1	Fraction of fines in the ore
$\alpha_r$	0.05	0.15	0.1	Fraction of rocks in the ore
$\phi_f$	14	42	28	Power needed per ton of fines produced [(kW h)/t]
$\phi_r$	55	83	69	Rock abrasion factor [(kW h)/t]
$\phi_b$	89	99	94	Steel abrasion factor [(kW h)/t]

(iii) to minimize the resources (steel) used to produce final product and (iv) to minimize the power consumed for each ton of fines produced. It is impossible to satisfy all objectives, since sub-objectives (i) and (ii) have a negative effect on objectives (iii) and (iv). A trade-off between these objectives has to be found.

### 3. MILL CIRCUIT MODULES

An individual model for each module, shown in Fig. 1, has been created in the past (Coetzee et al. (2010); Le Roux et al. (2013)). A summary of the modules are presented here, similar to the models by Olivier et al. (2012). The model parameters used by Olivier et al. (2012) are employed in this paper. The mill module in this section will serve as the basis for further state estimation work. The mill has five states, namely water, rocks, solids, fines and steel balls. The variables and nomenclature used in this section are shown in Table 1 and Table 2 respectively.

The state equations for the mill are given by equations (1) to (5).

$$\dot{X}_{mw} = MIW - V_{wo} \quad (1)$$

$$\dot{X}_{ms} = \frac{MFS}{D_s}(1 - \alpha_r) - V_{so} + RC \quad (2)$$

$$\dot{X}_{mf} = \frac{MFS}{D_s}\alpha_r - V_{fo} + FP \quad (3)$$

$$\dot{X}_{mr} = \alpha_f \frac{MFS}{D_s} - RC \quad (4)$$

$$\dot{X}_{mb} = \frac{MFS}{D_b} - BC \quad (5)$$

The variables for equations (1) to (5) are given by equations (6) to (8).

$$RC \triangleq \frac{1}{D_s \phi_r} \cdot P_{mill} \cdot \varphi \cdot \left( \frac{X_{mr}}{X_{mr} + X_{ms}} \right) \quad (6)$$

$$BC \triangleq \frac{1}{D_s \phi_b} \cdot P_{mill} \cdot \varphi \cdot \left( \frac{X_{mr}}{X_{mr} + X_{ms}} \right) \quad (7)$$

$$FP \triangleq \frac{P_{mill}}{D_s \phi_f [1 + \alpha_{\phi_f} ((LOAD/v_{mill}) - v_{Pmax})]} \quad (8)$$

Table 2. Nomenclature. From Olivier et al. (2012)

Volumetric flow-rates used for internal flows: in (i), out (o)	Description
$V_{wi}, V_{wo}$	Water [m <sup>3</sup> /h]
$V_{si}, V_{so}$	Solids [m <sup>3</sup> /h]
$V_{fi}, V_{fo}$	Fines [m <sup>3</sup> /h]
$V_{ri}, V_{ro}$	Rocks [m <sup>3</sup> /h]
$V_{bi}, V_{bo}$	Steel [m <sup>3</sup> /h]
$D_s$	Density of feed ore [kg/m <sup>3</sup> ]
$D_b$	Density of steel balls [kg/m <sup>3</sup> ]

The volumetric flow-rates out of the mill, defined in Table 2, are given by equations (9) to (11).

$$V_{wo} = V_V \cdot \varphi \cdot X_{mw} \cdot \left( \frac{X_{mw}}{X_{mw} + X_{ms}} \right) \quad (9)$$

$$V_{so} = V_V \cdot \varphi \cdot X_{mw} \cdot \left( \frac{X_{ms}}{X_{mw} + X_{ms}} \right) \quad (10)$$

$$V_{fo} = V_V \cdot \varphi \cdot X_{mw} \cdot \left( \frac{X_{mf}}{X_{mw} + X_{ms}} \right) \quad (11)$$

The model includes the effect of the mill power and slurry rheology on the breakage and power functions (Shi and Napier-Munn (2002)). The rheology factor is given by equation (12).

$$\varphi \triangleq \sqrt{\frac{\max[0, (X_{mw} - ((\frac{1}{\varepsilon_{ws}} - 1)X_{ms}))]}{X_{mw}}} \quad (12)$$

The discharge grate at the end of the mill prohibits the steel balls and rocks from exiting. The LOAD is given by equation (13).

$$LOAD = X_{mw} + X_{ms} + X_{mr} + X_{mb} \quad (13)$$

The power that the mill motor supplies is given by equation (14). The mill power has a parabolic relationship to the mill load (fraction of the total mill volume) (Van Nierop and Moys (2001)). The increase in load results in an increase in power; however, when the load increases above approximately 55% (Van Nierop and Moys (2001)) full the power starts to decrease.

$$P_{mill} = P_{max} \cdot (1 - \delta_{Pv} Z_x^2 - 2\chi_P \delta_{Pv} \delta_{Ps} Z_x Z_r - \delta_{Ps} Z_r^2) \quad (14)$$

$$Z_x \triangleq \frac{X_{mw} + X_{ms} + X_{mr} + X_{mb}}{v_{pmax} \cdot v_{mill} - 1} \quad (15)$$

$$Z_r \triangleq \frac{\varphi}{\varphi P_{max}} - 1 \quad (16)$$

The hold up of water ( $X_{mw}$ ), solids ( $X_{ms}$ ), fines ( $X_{mf}$ ), rocks ( $X_{mr}$ ) and steel balls ( $X_{mb}$ ) in the mill is estimated in the sections that follow. The measured outputs are: volumetric flow-rates out of the mill, LOAD and  $P_{mill}$ .

## 4. STATE ESTIMATION

### 4.1 Simulation setup

In order to illustrate the accuracy of the estimation algorithms, a simulation run is performed while the milling circuit is kept under feedback control by PI controllers for a 20-hour period. The same PI controllers and simulation environment were used by Olivier et al. (2012) who provide further details and justification. Disturbances are introduced as follows: The value of  $\phi_f$  is decreased by 10% at time 3 h, the value of  $\alpha_r$  is decreased by 10% at time

9 h and the value of  $\alpha_f$  is increased by 10% at time 15 h. The “true” simulated states are estimated and therefore no measurement noise is taken into account. Also, this comparison aims to show the estimation accuracy and not the noise-handling capability. A 10-second sampling time was used in the simulation.

### 4.2 Particle filter

*Description* Particle filtering is a technique of implementing a recursive Bayesian filter by Monte Carlo simulations. The setup of the particle filtering simulation run is the same as that used by Olivier et al. (2012), in which more information on the particle filter is presented. A brief description is given in this section.

Particle filtering relies on the technique of representing the posterior density function (pdf), which is used for estimation, by a set of random samples and associated weights. The locations of the particles represent the locations at which the pdf is evaluated and the sizes of the particles represent the associated weights, giving an indication of the value of the pdf at this location. This representation is expandable to an arbitrary number of dimensions and is applicable to any distribution, even multi-modal and other non-Gaussian distributions. As the number of particles becomes very large, this method of representing the pdf becomes equivalent to the functional description of the posterior pdf. The pdf at time  $t$  may then be approximated as (Arulampalam et al. (2002)):

$$p(x_t|Y_t) \approx \sum_{i=1}^{N_s} w_t^i \delta(x_t - x_t^i) \quad (17)$$

where  $N_s$  is the number of particles and  $\{x_t^i, w_t^i\}_{i=1}^{N_s}$  is the set of particles and associated weights. These weights are defined to be (Ristic et al. (2004)):

$$w_t^i \propto w_{t-1}^i \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, y_t)} \quad (18)$$

where  $q(x_t^i|x_{t-1}^i, y_t)$  is a proposal distribution called an importance density. Ideally the importance density should be the true posterior distribution  $p(x_t|Y_t)$ , but as this is not known in general, a proposal distribution is used.

*State estimation* The particle filter is specified with 50 particles. A larger number of particles have been tried without too much improvement in the estimation results. In this study (as in Olivier et al. (2012)) the transitional prior is used. The initial estimates of the mill states are randomly selected from a region ( $\pm 0.01$ ) around the actual initial values in each case. The “true” states and particle filter estimates are shown in Fig. 2.

### 4.3 Neural network

*Training* The neural network was trained using 480 hours of simulation data. The inputs to the mill were used to estimate the five states of the mill. Over-training the neural network is a factor that was considered when training the network, as an over-trained network will likely

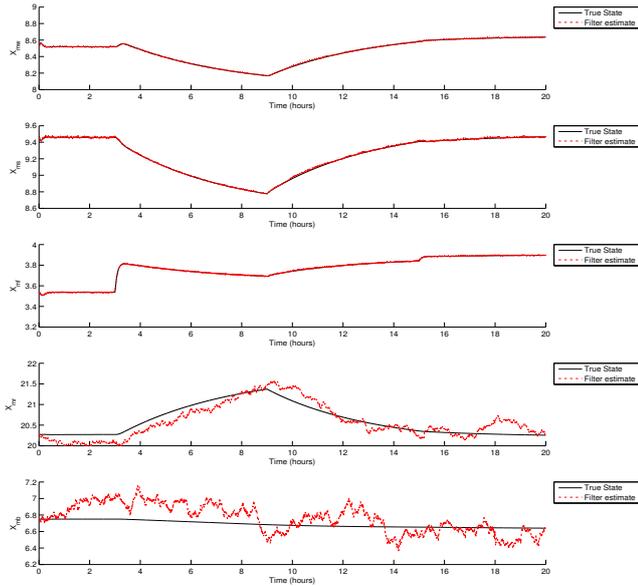


Fig. 2. Particle filter state estimates

only work for the specified case it was trained on. A positive and negative 20% step change was made to  $\phi_f$ ,  $\alpha_r$  and  $\alpha_f$ . The data consisted of 20% positive and negative step changes with various combinations of  $\phi_f$ ,  $\alpha_r$  and  $\alpha_f$  at 700 minutes, 40 minutes and 480 minutes respectively, as shown in Figures 4, 5 and 6.

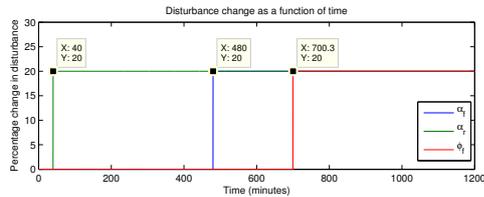


Fig. 3. Training data set 1 with positive 20% disturbance changes

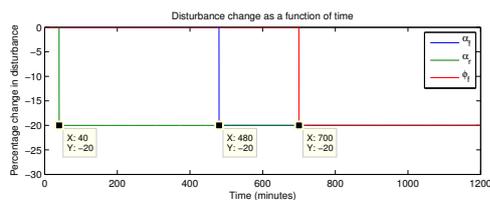


Fig. 4. Training data set 2 with negative 20% disturbance changes

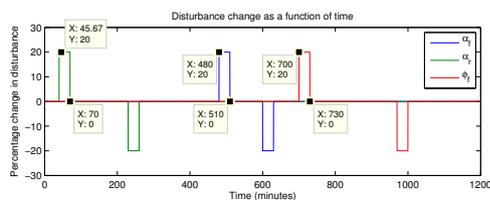


Fig. 5. Training data set 3 with positive and negative disturbance changes

A two-layer feedforward backpropagation network with sigmoid hidden neurons and linear output neurons was trained for state estimation. The data that were used for

the training was randomly arranged. The neural network was trained according to the Levenberg-Marquardt (Hagan et al. (1996)) optimization algorithm. This algorithm determines the weights and the bias values for the network. The network consisted of five hidden neurons.

**Results** The network was then tested on 10% changes using the same simulation environment as for the particle filter defined in section 4.1, to determine if the network would be able to predict smaller changes in parameters than it was trained on. The state estimation results for the neural network are shown in Fig. 7.

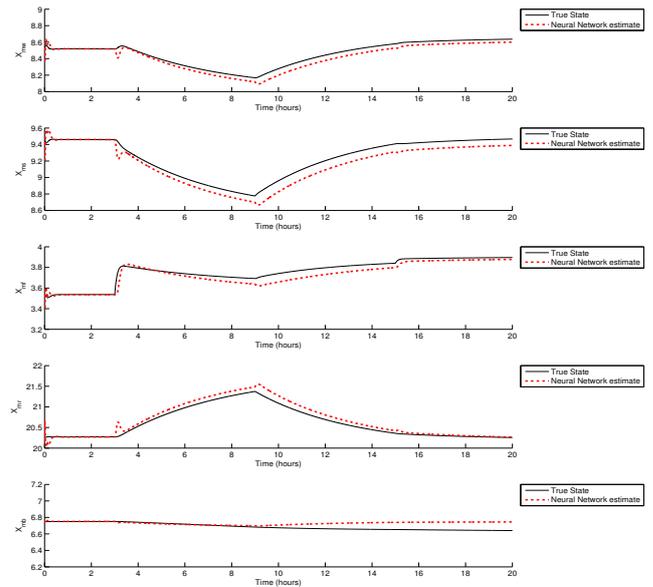


Fig. 6. State estimation from neural network

The results show that the neural network performs well for the first four states and deviates after 10 h for state 5, as shown in Fig. 6. The hold-up of steel balls (state 5) is also difficult to estimate for the particle filter, as shown in Fig. 2. A comparison between Fig. 2 and Fig. 6 indicates that the neural network performed better than the particle filter for states 4 and 5; however, the particle filter's estimates for the first three state were more accurate.

When comparing the neural network and the particle filter estimators, it should be kept in mind that the neural network estimator estimated the mill states based on only the mill input data. The particle filter, however, requires output measurements and an initial estimate of the mill parameters (presented in Olivier et al. (2012)) to function. These were provided to the particle filter estimator as described in section 4.2. The neural network estimator therefore had less information at its disposal when estimating the mill states.

#### 4.4 Combination of particle filter and neural network

An alternative method of combining the neural network and particle filter is described in this section. Fig. 2 shows that the particle filter estimate is more accurate for the first three states than the neural network estimate (Fig. 6); however, the neural network estimate was more accurate for states 4 and 5. This section investigates the use of

the particle filter to correct the offset found in the neural network estimates for the first three states, similar to offset correction described by Flament et al. (1993). For the first three states only, the average of the particle filter estimate at every hour is compared to the average of the neural network estimate. The average of the particle filter estimate is then subtracted from the average of the neural network estimates. The results of the neural network method with particle filter offset correction at every hour are shown in Fig. 7. The performance index as a function of time is shown in Fig. 8. An average performance index was determined from ten simulation runs because particle filtering is a Monte Carlo method and will therefore have a difference index for each iteration.

The algorithm, which shows how the particle filter method was used as an offset correction method for the neural network method, is presented below.  $M$  is the time at which the offset error correction will be implemented.  $N$  is the number of values used to determine the average of each method.  $N$  was chosen to be 10% of the  $M$  value. At every hour the difference between the neural network estimation and particle filter estimation was calculated based on six minutes of historical data. The offset correction was then implemented to the neural network estimation for an hour.

#### Algorithm

- Initialize average arrays to 0.
- Using three FOR loops, calculate the average of  $N$  time steps back at every  $M$  for each state. This is done for both methods.
- Find the difference between the two matrices. This is the offset between the neural network and the particle filter.
- For  $M$  time steps forward create a new matrix that subtracts the difference from the neural network estimation.
- $N$  was chosen to be 36 i.e. 6 minutes.
- $M$  was chosen to be 360 i.e. 60 minutes or 1 hour.

The combined method employs the particle filter method and therefore requires the mill parameter estimates.

#### 4.5 Comparison

The performance index shown in Fig. 8 clearly shows that an improvement has been made using the particle filter as an offset error correction method. The performance index used is shown in equation (19) where  $\bar{x}_{yz}$  represents the ideal state and  $x_{yz}$  represents the estimated state.

$$PI = \left(\frac{\bar{x}_{mw} - x_{mw}}{\bar{x}_{mw}}\right)^2 + \left(\frac{\bar{x}_{ms} - x_{ms}}{\bar{x}_{ms}}\right)^2 + \left(\frac{\bar{x}_{mf} - x_{mf}}{\bar{x}_{mf}}\right)^2 + \left(\frac{\bar{x}_{mr} - x_{mr}}{\bar{x}_{mr}}\right)^2 + \left(\frac{\bar{x}_{mb} - x_{mb}}{\bar{x}_{mb}}\right)^2 \quad (19)$$

Table 3 shows the results from 16 simulation scenarios. The first two scenarios illustrate the performance of the neural network training and therefore should have very small state errors. Tests number 3 to 7 illustrate a scenario when all three disturbance parameters are positive. This could occur when the parameters are underestimated. It should be noted that in Test number 7 the neural network could not estimate the states accurately, as a 30% disturbance change was made and the network was only trained on a

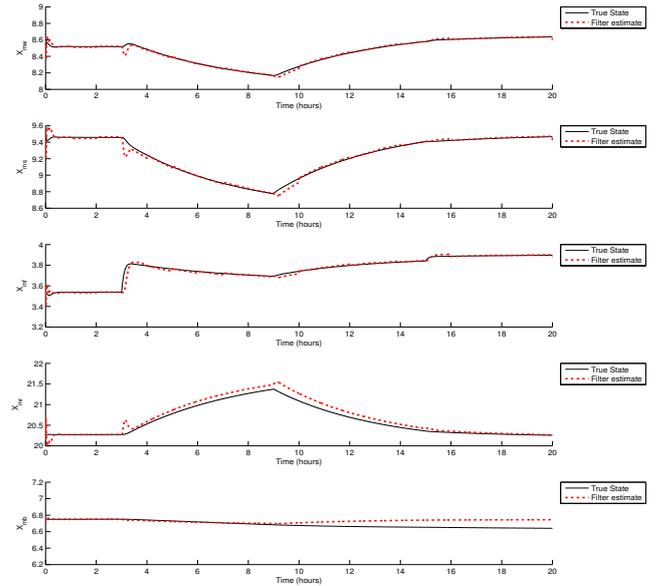


Fig. 7. State estimation from neural network with particle filter correction

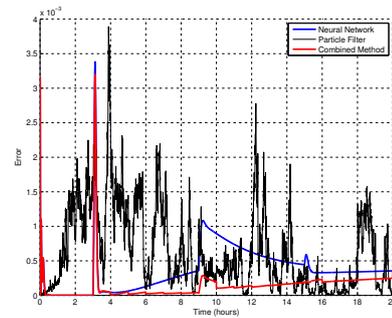


Fig. 8. Squared error comparison of all three methods

20% change. This shows that the neural network needs to be trained on a worse-case scenario of disturbance changes else the state estimation results are not reliable. Similarly, Test numbers 8 to 12 illustrate negative disturbance changes. Test numbers 13 to 16 illustrate a scenario when positive and negative disturbance changes occur. The results show that as the magnitude of the disturbances increases, the estimation accuracy for both particle filter and neural network decreases. The particle filter results are more consistent compared to the neural network results. The combined method results are always better than the individual neural network and particle filter results except in Test number 7 and 12.

## 5. CONCLUSION

The work presented shows that it is possible to do internal state estimation for a milling circuit using a neural network trained on input data. The network was trained using disturbance changes in model parameters of 20%, and then used to predict the effect of 10% disturbances. A new method that combines a neural network and particle filter estimator for offset correction was presented. Initial results indicate that such a method can work well.

Table 3. State estimation validation tests and corresponding performance index results

Test No.	$\alpha_r$		$\alpha_f$		$\phi_f$		State error <sup>a</sup>		
	Change (%)	Time (mins)	Change (%)	Time (mins)	Change (%)	Time (mins)	NN	PF	Combined
1	20	40	20	480	20	700	0.6893	16.849	0.4394
2	-20	40	-20	480	-20	700	0.8389	11.547	0.5505
3	5	540	5	900	5	180	0.3905	10.782	0.2399
4	10	540	10	900	10	180	1.0691	7.6039	0.7326
5	15	540	15	900	15	180	2.2655	15.126	1.4364
6	20	540	20	900	20	180	4.6991	13.889	2.2946
7	30	540	30	900	30	180	23.586	11.844	13.682
8	-5	540	-5	900	-5	180	1.2861	14.868	0.3885
9	-10	540	-10	900	-10	180	2.7745	13.913	1.0001
10	-15	540	-15	900	-15	180	3.4191	13.158	2.0793
11	-20	540	-20	900	-20	180	6.2087	17.003	3.5542
12	-30	540	-30	900	-30	180	75.278	13.967	14.469
13	-5	540	5	900	-5	180	1.4202	11.286	0.3881
14	-10	540	10	900	-10	180	2.6133	11.155	0.9994
15	-15	540	15	900	-15	180	3.3618	10.977	2.0905
16	-20	540	20	900	-20	180	7.7633	14.980	3.5137

<sup>a</sup> Based on the summation of equation 19 throughout a 20-hour simulation at the sampling interval of 10 seconds.

Training the neural network in practice will be difficult, as the method is highly dependent on the quality of the training data. It may also not be possible to train the neural network on one disturbance at a time, as was done here. Plants outside the minerals processing industry are often accompanied by accurate simulators (Garatti and Bittanti (2008)) and according to Garatti and Bittanti (2008) a set of experiments using neural networks can be performed “virtually” by simulation trials. Further research should be done on how much training data are required and to what extent the neural network method is accurate in mineral processing applications. A promising start has, however, been made.

## REFERENCES

- Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Process.*, 50, 1741-1788.
- Bauer, M. and Craig, I.K. (2008). Economic assessment of advanced process control - A survey and framework. *J. Process Control*, 18(1), 2-18.
- Chen, X., Zhai, J., Li, S., and Li, Q. (2007). Application of model predictive control in ball mill grinding circuit. *Minerals Eng.*, 20(11), 1099 - 1108.
- Coetzee, L., Craig, I., and Kerrigan, E. (2010). Robust nonlinear model predictive control of a run-of-mine ore milling circuit. *IEEE Transactions on Control Systems Technology*, 18(1), 222 - 229.
- Craig, I., Aldrich, C., Braatz, R., et al. (2011). Control in the process industries, in the impact of control technology. Technical report, IEEE Control Systems Society. URL <http://www.ieeecss.org/main/IoCT-report>.
- Craig, I., Hulbert, D.G., Metzner, G., and Moulton, S.P. (1992). Optimized multivariable control of an industrial run-of-mine circuit. *Journal of the South African Institute of Mining and Metallurgy*, 2, 169 - 176.
- Craig, I. (2012). Grinding mill modelling and control: past, present and future. Plenary Lecture, 31st Chinese Control Conference, Hefei, China.
- Craig, I. and MacLeod, I. (1995). Specification framework for robust control of a run-of-mine ore milling circuit. *Control Engineering Practice*, 3(5), 621 - 630.
- Flament, F., Thibault, J., and Hodouin, D. (1993). Neural network based control of mineral grinding plants. *Minerals Engineering*, 6(3), 235 - 249.
- Garatti, S. and Bittanti, S. (2008). Parameter estimation via artificial data generation with the “two-stage” approach. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 5600-5604.
- Hagan, M.T., Demuth, H.B., and Beale, M. (1996). *Neural network design*. PWS Publishing Co., Boston, MA, USA.
- Hodouin, D. (2011). Methods for automatic control, observation, and optimization in mineral processing plants. *Journal of Process Control*, 21(2), 211 - 225.
- Le Roux, J., Craig, I., Hulbert, D., and Hinde, A. (2013). Analysis and validation of a run-of-mine ore grinding mill circuit model for process control. *Minerals Engineering*, 43-44, 121-134.
- Makokha, A. and Moys, M. (2012). Multivariate approach to on-line prediction of in-mill slurry density and ball load volume based on direct ball and slurry sensor data. *Minerals Engineering*, 26(1), 13-23.
- Olivier, L., Huang, B., and Craig, I. (2012). Dual particle filters for state and parameter estimation with application to a run-of-mine ore mill. *Journal of Process Control*, 22(4), 710 - 717.
- Ristic, B., Arulampalam, S., and Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston.
- Shi, F. and Napier-Munn, T. (2002). Effects of slurry rheology on industrial grinding performance. *International Journal of Mineral Processing*, 65(3-4), 125-140.
- Stange, W. (1993). Using artificial neural networks for the control of grinding circuits. *Minerals Engineering*, 6(5), 479 - 489.
- Stanley, G.G. (1987). *The extractive metallurgy of gold in South Africa*. South African Institute of Mining and Metallurgy: Johannesburg.
- Van Nierop, M. and Moys, M. (2001). Exploration of mill power modelled as function of load behaviour. *Minerals Engineering*, 14(10), 1267-1276.
- Wei, D. and Craig, I.K. (2009). Grinding mill circuits - a survey of control and economic concerns. *International Journal of Mineral Processing*, 90(14), 56 - 66.