

Dual particle filters for state and parameter estimation with application to a run-of-mine ore mill

L.E. Olivier^a, B. Huang^b, I.K. Craig^{a,*}

^a*Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, Pretoria, South Africa.*

^b*Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Canada.*

Abstract

Measurements are not readily available for grinding mills owing to the nature of the milling operation. State and parameter estimation for a grinding mill which forms part of a run-of-mine ore milling circuit has been implemented. These estimates may then be used in an advanced control algorithm. The estimation was done with dual particle filters as well as with a simultaneous estimation scheme, on simulated data, to compare the performances. The sensitivity analyses for the different schemes show the class of systems in which dual estimation may produce superior results.

Keywords: Dual estimation, milling, particle filter, state/parameter estimation, run-of-mine ore

*Corresponding author. Address: Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, Pretoria, South Africa.

Tel.: +27 12 420 2172; fax: +27 12 362 5000.

Email address: icraig@postino.up.ac.za (I.K. Craig)

1. Introduction

Mineral processing operations are generally difficult to control owing to the presence of strong external disturbances, poor process models and process variables that are difficult to measure [1]. Run-of-mine (ROM) ore milling is an example of such a process, as grinding ore down to a fine product is usually the first step in any metallurgical extraction process [2]. Compensating for the effects of strong external disturbances on the milling circuit through the use of disturbance observers has previously been addressed [3], [4]. The use of model-plant mismatch detection [5] to assess the quality of the available model has also been addressed [6]. In this paper the focus is on simultaneous state and parameter estimation concerning the mill, for which measurements are not readily available.

State estimation is important for advanced control techniques such as model predictive control ([7]) in which the state values are used to determine the optimal control signal. This is especially true for nonlinear predictive control where nonlinear observers are necessary to produce estimates of the unmeasured state variables [8].

The parameters with large variances that are contained in the milling circuit have a big effect on the grinding performance [9] and accurate estimation of these parameter values would be valuable if they were to be incorporated into the control strategy.

Particle filters are sophisticated model estimation techniques that make use of sequential Monte Carlo (SMC) estimation based on point mass (particle) representations of probability densities [10]. Because the method approximates the posterior distribution by a set of weighted samples (the particles)

without making any explicit assumption about the form of the distribution, it can be used in general nonlinear, non-Gaussian systems [11]. Because of the nonlinearities present in the milling equations, the method of particle filtering is preferred for state and parameter estimation.

This paper presents the application of particle filtering to a simulated ROM ore milling circuit for state and parameter estimation. The paper also investigates the performance of dual estimation over simultaneous estimation through a sensitivity analysis and shows a comparison of the two estimation approaches.

2. State estimation

Consider the general state-space representation of a dynamic system

$$\dot{x}(t) = f(x(t), u(t), \theta(t), v(t)) \quad (1)$$

$$y(t) = g(x(t), \theta(t), e(t)) \quad (2)$$

where $x \in \mathbb{R}^n$ is the state vector and $y \in \mathbb{R}^m$ is the output vector, $f(\cdot)$ and $g(\cdot)$ are possibly nonlinear functions describing the state transitions and the outputs respectively, $u(t)$ contains the exogenous inputs, $\theta(t)$ represents the parameters, $v(t)$ is the state noise and $e(t)$ is the measurement noise.

2.1. State augmentation

In order to do simultaneous state and parameter estimation, the parameters are first augmented as states to the system, and then the procedure reduces to a pure state estimation problem. Consider the original system

in (1) and (2); the original state vector ($x(t)$) is now augmented with the parameters ($\theta(t)$) to be

$$x^*(t) = \begin{bmatrix} x(t) \\ \theta(t) \end{bmatrix}. \quad (3)$$

Each parameter is now defined a state transition equation in the form

$$\dot{\theta}_k(t) = \dot{x}_{n_x+k}(t) = 0 + w(t) \quad (4)$$

where $w(t)$ is the parameter noise, n_x is the original number of states and $k \in 1, \dots, n_p$ with n_p the number of parameters to be estimated. The model used to describe the parameter transitions in (4) is called a random walk model, in which no dynamic transition term is included. Any change in the parameter is attributed to the noise term only. This noise term will later be a tuning knob in the parameter estimation procedure as a large variance in $w(t)$ makes for faster tracking but with a larger estimation variance. A small variance in $w(t)$ will make the tracking much smoother but also much slower. The system equations now become

$$\dot{x}^*(t) = f(x^*(t), u(t), v^*(t)) \quad (5)$$

$$y(t) = g(x^*(t), e(t)) \quad (6)$$

with $v^*(t)$ the augmented noise vector $v^*(t) = \begin{bmatrix} v(t) \\ w(t) \end{bmatrix}$ comprising the original state noise and the parameter noise.

2.2. State estimation procedure

Let $Y_t = \{y_0, \dots, y_t\}$ represent the sequence of all measurements up to the current time, then the general state estimation problem is formulated

as the solution of the conditional distribution function $p(x_t|Y_t)$, which is the distribution of the state given all the observations up to time t . The general solution to the latter is then given by [12]:

$$p(x_t|Y_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|Y_{t-1})dx_{t-1} \quad (7)$$

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \quad (8)$$

where (7) is also known as the Chapman-Kolmogorov equation. Equations (7) and (8) give a recursive procedure for state estimation that consists of two steps:

1. Prediction step: Predict the pdf, $p(x_t|Y_{t-1})$ from the state transition equation $f(\cdot)$ via (7).
2. Update step: Determine $p(x_t|Y_t)$ from $p(x_t|Y_{t-1})$ via (8).

This recursive procedure is only a conceptual solution though, as the equations are not computable in general [13]. Solutions do exist in special cases, but when the analytic solution is intractable, particle filters may be employed that approximate the optimal Bayesian solution.

3. Particle filtering

Particle filtering is a technique of implementing a recursive Bayesian filter by Monte Carlo simulations. The required posterior density function (pdf) is represented by a set of random samples and associated weights. The idea of representing a pdf in this fashion is illustrated in Fig. 1. 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

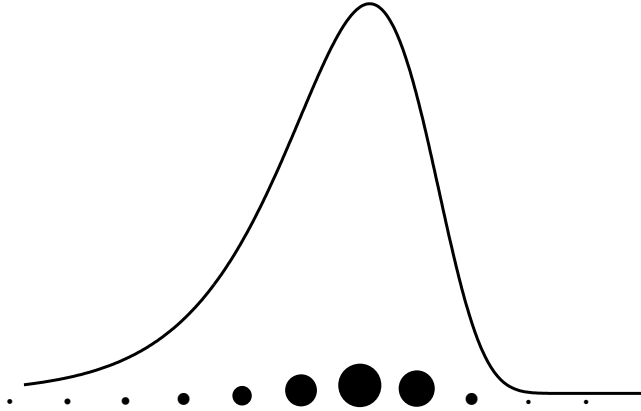


Figure 1: PDF represented by particles and associated weights

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 posterior density function at time t may then be approximated as [13]:

$$p(x_t|Y_t) \approx \sum_{i=1}^{N_s} w_t^i \delta(x_t - x_t^i) \quad (9)$$

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 [10]

$$w_t^i \propto w_{t-1}^i \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, y_t)} \quad (10)$$

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. The importance density plays an important role in the design and is discussed in Section 3.2.

3.1. Degeneracy

It has been shown [14] that the variance of the importance weights of (10) can only increase over time. This has a harmful effect on accuracy and is known as degeneracy because after only a couple of iteration steps all but one particle will have negligible normalized weights. Practically this means that a lot of computational time is spent to update particles that produce negligible contributions to the approximation of the posterior distribution. An indication of the degree of degeneracy is the effective number of particles defined by

$$\hat{N}_{eff} = \frac{1}{\frac{1}{N_s} \sum_{i=1}^{N_s} (w_t^i)^2}. \quad (11)$$

Degeneracy may be eliminated through the use of resampling that eliminates particles with low importance weights and multiplies particles with high importance weights. This means that the particles located at x_t^i with weights w_t^i are replaced by particles located at x_t^{i*} with uniform weights. One possible resampling scheme is systematic resampling [15], for which the pseudo-code is listed in Algorithm 1. The algorithm uses the cumulative sum of weights (CSW) defined as

$$CSW_i = \sum_{j=1}^i w_t^j \quad (12)$$

and draws samples from $u \in [0, 1]$ to map new samples from the CSW. This process is illustrated in Fig. 2 where the original particles that give form

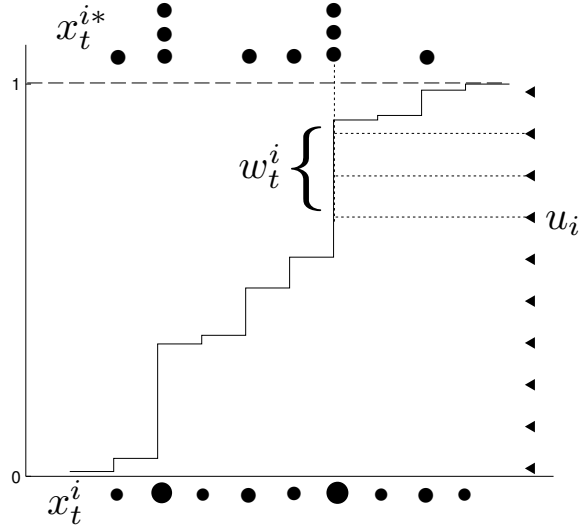


Figure 2: Resampling procedure

to the CSW are shown at the bottom of the figure. Samples are uniformly drawn at the points indicated by the left-pointing triangles and based on the value of the CSW function at that point, the resampled particles (x_t^{i*}) are determined from the locations of the original particles (x_t^i). From Fig. 2 it is visible that particles with larger weights have a bigger chance of being drawn (and consequently duplicated) than particles with smaller weights.

3.2. Importance density

The optimal importance density function that minimizes the variance of importance weights conditioned upon x_{t-1}^i and y_t has been shown [14] to be

$$q(x_t|x_{t-1}^i, y_t)_{opt} = p(x_t|x_{t-1}^i, y_t) \quad (13)$$

$$= \frac{p(y_t|x_t, x_{t-1}^i)p(x_t|x_{t-1}^i)}{p(y_t|x_{t-1}^i)} \quad (14)$$

This optimal importance density is however only useable in a specific class

Algorithm 1 Systematic resampling

 $[\{x_t^{j*}, w_t^j\}_{j=1}^N] = \text{RESAMPLE}[\{x_t^i, w_t^i\}_{i=1}^N]$

- Initialize the CSW: $c_1 = w_t^1$
 - FOR $i = 2 : N$
 - Construct CSW: $c_i = c_{i-1} + w_t^i$
 - END FOR
 - Begin at bottom of the CSW: $i = 1$
 - Draw a starting point: $u_1 \sim \mathcal{U}[0, \frac{1}{N}]$
 - FOR $j = 1 : N$
 - Move along the CSW: $u_j = u_1 + \frac{1}{N}(j - 1)$
 - Find the smallest value of i such that $u_j \leq c_i$
 - Assign sample: $x_t^{j*} = x_t^i$
 - Assign weight: $w_t^j = \frac{1}{N}$
 - END FOR
-

of problems where it is possible to sample from $p(x_t|x_{t-1}^i, y_t)$ and where

$$p(y_t|x_{t-1}^i) = \int p(y_t|x_t)p(x_t|x_{t-1}^i)dx_t \quad (15)$$

can be calculated up to a normalizing constant, which is not the case in general. One popular suboptimal choice is the transitional prior

$$q(x_t|x_{t-1}^i, y_t) = p(x_t|x_{t-1}^i) \quad (16)$$

which, if it is furthermore assumed that the process noise is additive zero-mean Gaussian noise, simply becomes

$$p(x_t|x_{t-1}^i) = \mathcal{N}(x_t; f_{t-1}(x_{t-1}^i), \mathbf{Q}_{t-1}) \quad (17)$$

which can easily be calculated. This assumption also means that particles can be drawn from a Gaussian distribution with a mean equal to the previous particle location propagated through the system equations and standard deviation equal to the noise standard deviation as

$$x_t^i \sim \mathcal{N}(f_{t-1}(x_{t-1}^i), \mathbf{Q}_{t-1}). \quad (18)$$

3.3. SIR particle filter

Now that all the elements of the particle filtering algorithm have been described, the complete algorithm can be given. The sampling importance resampling (SIR) particle filter [16] is a version of the general particle filtering algorithm [13] in which the importance density is chosen as the transitional prior and resampling is done at each step. The assumptions required to use the SIR particle filter are very weak [10], making it suitable for a wide variety of implementations. The SIR type particle filter is used throughout in this study.

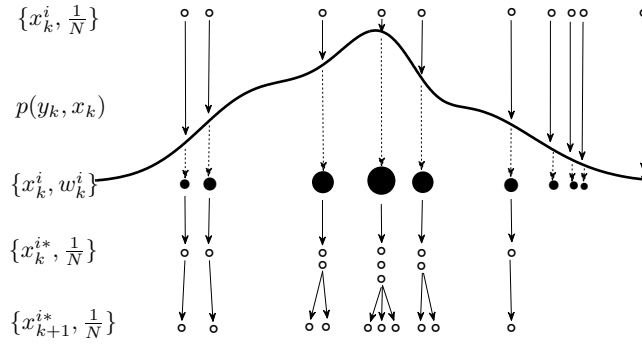


Figure 3: One iteration of the particle filter algorithm (adapted from [10])

One iteration of the SIR particle filter algorithm is presented in Fig.3; the pseudo-code describing the algorithm is listed as Algorithm 2.

Because resampling is done at every time step it is unnecessary to pass the weights on between successive iterations, as all the weights are uniform $w_t^i = \frac{1}{N}$.

4. Dual particle filtering

Under certain circumstances, which will be explored in Section 6, simultaneous state and parameter estimation through augmenting the parameters as states might not produce good results. In these situations dual estimation may be implemented, which makes use of two particle filters running in parallel, as shown in Fig. 4. Here one particle filter is used solely for state estimation and the other solely for parameter estimation.

In the dual estimation framework the two particle filters operate in an iterative fashion. At each time step the current state estimate is given as a

Algorithm 2 SIR particle filter

 $[\{x_t^i\}_{i=1}^N] = \text{SIR}[\{x_{t-1}^i\}_{i=1}^N, y_t]$

- FOR $i = 1 : N$
 - Draw $x_t^i \sim p(x_t|x_{t-1})$
 - Calculate $\tilde{w}_t^i = p(y_t|x_t)$
- END FOR
- Calculate the sum of the weights: $\Sigma_w = \text{SUM}[\{\tilde{w}_t^i\}]$
- FOR $i = 1 : N$
 - Normalize: $w_t^i = \frac{\tilde{w}_t^i}{\Sigma_w}$
- END FOR
- Resample using Algorithm 1

 $[\{x_t^i\}_{i=1}^N] = \text{RESAMPLE}[\{x_t^i, w_t^i\}_{i=1}^N]$

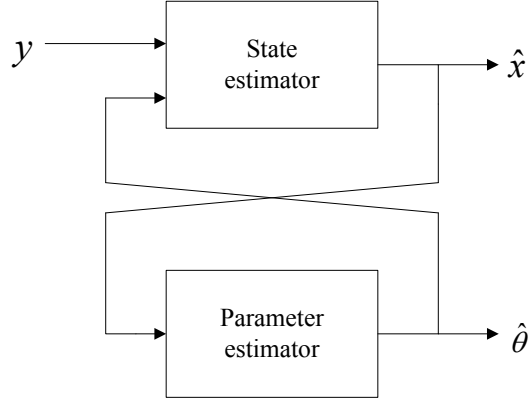


Figure 4: Dual state and parameter estimation

known input to the particle filter doing parameter estimation. Similarly the current parameter estimate is given as a known input to the particle filter doing the state estimation. The state estimator draws particles as

$$x_t^i \sim p(x_t | x_{t-1}^i, \hat{\theta}_{t-1}) \quad (19)$$

and then the weights are calculated as

$$\tilde{w}_{t,x}^i = p(y_t | x_t^i). \quad (20)$$

The parameter estimator draws particles as

$$\theta_t^i \sim p(\theta_t | \theta_{t-1}^i, \hat{x}_{t-1}) \quad (21)$$

and then the weights are calculated as

$$\tilde{w}_{t,\theta}^i = p(\hat{x}_t | \theta_t^i, \hat{x}_{t-1}). \quad (22)$$

The SIR particle filter given in Algorithm 2 is still directly applicable in this case although the samples are drawn from different distributions and

different transitional priors are used. This method has previously been implemented with two Kalman filters as discussed in [11] but as far as the authors know it has not been done with two particle filters.

A possible drawback of the dual estimation algorithm is the relative time delay between when the state and parameter estimates become available. The iterative nature of the dual estimation algorithm makes it computationally expensive. However, given that processes in the mineral processing industry usually have large time constants and sampling times, they are suited for dual estimation applications. For this system the sampling time is 10 seconds which is more than enough time for the estimation algorithms to be completed.

5. Description of the ROM Ore Milling Circuit

This section gives a description of a ROM ore milling circuit, operated in a closed circuit topology, similar to the description of [4]. This is the process for which state and parameter estimation is done.

Precious metal-bearing ore (such as gold or platinum) is fed to the milling circuit at about 100 t/h, see Fig. 5. The ore is ground down to a product with a particle size of 80% smaller than 75 μm ($P_{80} = 75\mu\text{m}$). A hydrocyclone is used in closed circuit with the mill to separate the product from the out-of-specification material. The metal, which leaves the milling circuit as part of the overflow of the cyclone, is then extracted through a leaching or flotation process downstream. The underflow of the cyclone that contains the out-of-specification material is fed back into the mill for further grinding.

The feed to the mill consists of the underflow of the cyclone, feed ore,

water and steel balls. Steel balls are added to help with the grinding down of ore inside the mill. The addition of steel balls is usually done in discrete quantities by the operator but in this study it will be treated as a continuous variable. The mill discharges sufficiently ground slurry into a sump through an end-discharge grate. The slurry is diluted with water in the sump and pumped to the hydrocyclone for classification. The product of the milling circuit is the overflow of the hydrocyclone.

The controlled variables in the milling circuit are the product particle size (PSE), the fraction of the mill volume filled with material (LOAD), and the volume of slurry in the sump (SLEV). The manipulated variables are the feed-rate of solids into the mill (MFS), the feed-rate of water into the mill (MIW), the feed-rate of steel balls into the mill (MFB), the flow-rate of water into the sump (SFW), and the flow-rate of slurry into the cyclone (CFF). The operating point of the milling circuit variables and constraints on these variables are based on [9] and given in the nomenclature table.

The milling circuit model is based on phenomenological equations and consists of separate modules for the feeder, mill, sump and hydrocyclone such that arbitrary circuit topologies may be constructed. The model uses five states, namely water, rocks, solids, fines, and steel balls to describe the flow of material through the milling circuit. All the equations that constitute the non-linear model are based on these material classifications. A full description of these equations can be found in [9].

5.1. Mill module

As this study is concerned with the combined state and parameter estimation for the mill, the phenomenological equations describing the milling

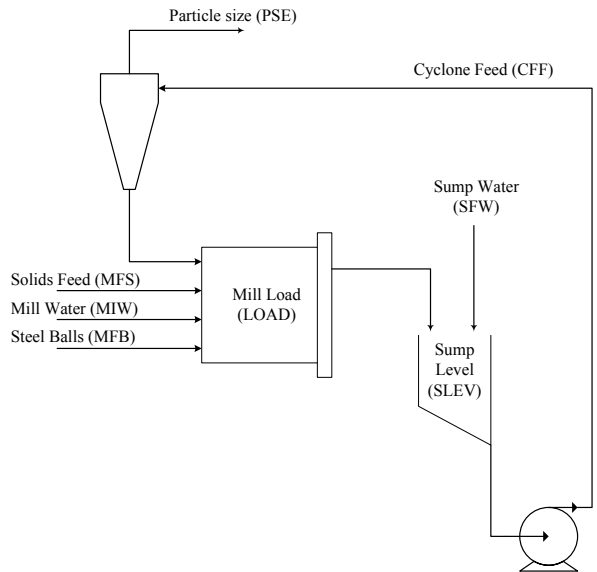


Figure 5: ROM ore milling circuit

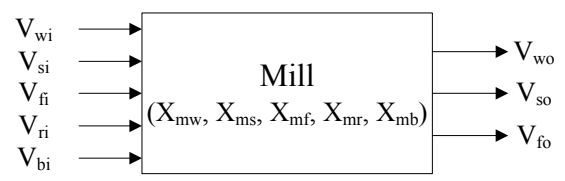


Figure 6: Mill showing component flows and states

module are given here. The mill has five states, which are the holdups of the five classifications of material in the mill, namely water, rocks, solids, fines, and steel balls (see Fig. 6). Fig. 6 also shows the streams into and out of the mill that have been broken up into these separate components.

Nomenclature					
	Minimum	Maximum	Nominal	% Δ	Description
Variables					
MIW	0	100	33.33		Flow-rate of water to the mill [$\frac{m^3}{h}$]
MFS	0	200	100		Flow-rate of solids to the mill [$\frac{t}{h}$]
MFB	0	4	2		Flow-rate of steel balls to the mill [$\frac{t}{h}$]
CFE	400	500	442		Flow-rate of slurry to the cyclone [$\frac{m^3}{h}$]
SFW	0	400	267		Flow-rate of water to the sump [$\frac{m^3}{h}$]
PSE	60	90	80		Product particle size [% < 75 μ m]
LOAD	30	50	45		Total charge of the mill [%]
SLEV	2	37.5	30		Level of the sump [m^3]
Internal flows					
V_{wi}, V_{wo}					Flow of water into/out of the mill [$\frac{m^3}{h}$]
V_{si}, V_{so}					Flow of solids into/out of the mill [$\frac{m^3}{h}$]
V_{fi}, V_{fo}					Flow of fines into/out of the mill [$\frac{m^3}{h}$]
V_{ri}, V_{ro}					Flow of rocks into the mill [$\frac{m^3}{h}$]
V_{bi}, V_{bo}					Flow of steel balls into the mill [$\frac{m^3}{h}$]
States					
X_{mw}	0	50	8.53		Holdup of water in the mill [m^3]
X_{ms}	0	50	9.47		Holdup of solid ore in the mill [m^3]
X_{mf}	0	50	3.54		Holdup of fine ore in the mill [m^3]
X_{mr}	0	50	20.25		Holdup of rocks in the mill [m^3]
X_{mb}	0	20	6.75		Holdup of steel balls in the mill [m^3]
Parameters					
α_f	0.05	0.15	0.1	50	Fraction of fines in the ore [dimensionless]
α_r	0.05	0.15	0.1	50	Fraction of rocks in the ore [dimensionless]
ϕ_f	14	42	28	50	Power needed per ton of fines produced [$\frac{kW \cdot h}{t}$]
ϕ_r	55	83	69	20	Rock abrasion factor [$\frac{kW \cdot h}{t}$]
ϕ_b	89	99	94	5	Steel abrasion factor [$\frac{kW \cdot h}{t}$]
Constants					
ε_{ws}			0.6		Maximum water-to-solids volumetric flow at zero pulp flow [dimensionless]
V_V			40		Volumetric flow per "flowing volume" driving force [h^{-1}]
P_{max}			2000		Maximum mill motor power [kW]
δ_{P_v}			1		Power change parameter for volume [dimensionless]
δ_{P_s}			1		Power change parameter for fraction solids [dimensionless]
$v_{P_{max}}$			0.45		Fraction of mill volume filled for maximum power [dimensionless]
$\varphi_{P_{max}}$			0.51		Rheology factor for maximum mill power [dimensionless]
α_P			0.82		Fractional power reduction per fractional reduction from maximum mill speed [dimensionless]
v_{mill}			100		Mill volume [m^3]
α_{ϕ_f}			0.01		Fractional change in kW/fines produced per change in fractional filling of mill [dimensionless]
χ_P			0		Cross term for maximum power [dimensionless]

The state equations are given by

$$\dot{X}_{mw} = MIW - V_{wo} \quad (23)$$

$$\dot{X}_{ms} = \frac{MFS}{D_s} (1 - \alpha_r) - V_{so} + RC \quad (24)$$

$$\dot{X}_{mf} = \alpha_f \frac{MFS}{D_s} - V_{fo} + FP \quad (25)$$

$$\dot{X}_{mr} = \alpha_r \frac{MFS}{D_s} - RC \quad (26)$$

$$\dot{X}_{mb} = \frac{MFB}{D_b} - BC \quad (27)$$

where each of the feed streams has been replaced by its respective expression; RC is the amount of rocks consumed,

$$RC \triangleq \frac{1}{D_s \phi_r} \cdot P_{\text{mill}} \cdot \varphi \cdot \left(\frac{X_{mr}}{X_{mr} + X_{ms}} \right), \quad (28)$$

BC is the amount of balls consumed,

$$BC \triangleq \frac{1}{D_b \phi_b} \cdot P_{\text{mill}} \cdot \varphi \cdot \left(\frac{X_{mr}}{X_{mr} + X_{ms}} \right), \quad (29)$$

FP is the amount of fines produced,

$$FP \triangleq \frac{P_{\text{mill}}}{D_s \phi_f \left[1 + \alpha_{\phi_f} \left(\frac{LOAD}{v_{\text{mill}}} - v_{P_{\text{max}}} \right) \right]}, \quad (30)$$

and the flows out of the mill are given by

$$V_{wo} = V_V \cdot \varphi \cdot X_{mw} \left(\frac{X_{mw}}{X_{mw} + X_{ms}} \right) \quad (31)$$

$$V_{so} = V_V \cdot \varphi \cdot X_{mw} \left(\frac{X_{ms}}{X_{mw} + X_{ms}} \right) \quad (32)$$

$$V_{fo} = V_V \cdot \varphi \cdot X_{mw} \left(\frac{X_{mf}}{X_{mr} + X_{ms}} \right) \quad (33)$$

where φ is the rheology factor

$$\varphi \triangleq \sqrt{\frac{\max \left[0, \left(X_{mw} - \left(\frac{1}{\varepsilon_{ws}} - 1 \right) X_{ms} \right) \right]}{X_{mw}}}. \quad (34)$$

No rocks or steel balls can exit the mill as they are restricted by the discharge grate. Two other important expressions contained in the milling equations are the total charge in the mill (*LOAD*) and the power drawn from the mill motor (P_{mill}) given by

$$LOAD = X_{mw} + X_{ms} + X_{mr} + X_{mb} \quad (35)$$

$$P_{\text{mill}} = P_{\text{max}} \cdot \{1 - \delta_{P_v} Z_x^2 - 2\chi_p \delta_{P_v} \delta_{P_s} Z_x Z_r - \delta_{P_s} Z_r^2\}, \quad (36)$$

where Z_x is the effect of the load on the power consumption defined as $Z_x \triangleq (X_{mw} + X_{ms} + X_{mr} + X_{mb}) / (v_{P_{\text{max}}} \cdot v_{\text{mill}} - 1)$ and Z_r is the effect of the rheology on power consumption defined as $Z_r \triangleq (\varphi / \varphi_{P_{\text{max}}}) - 1$. All the other parameters and constants in the milling equations are listed in the nomenclature table.

The outputs are

$$g(x) = \begin{bmatrix} V_{wo} \\ V_{so} \\ V_{fo} \\ LOAD \\ P_{\text{mill}} \end{bmatrix}. \quad (37)$$

All five mill states as well as both the parameters α_r and ϕ_f are to be estimated from these equations.

5.2. Controller

In order to run the simulations the milling circuit needs to be operated under feedback control as the load inside the mill and the sump level are both open-loop unstable. These basic stabilization requirements coupled with the metallurgical needs from the circuit operation necessitates the incorporation of a feedback controller. Once the state and parameter estimates are available an advanced control strategy that makes use of full state feedback as well as incorporating the parameter values into the control signal calculation would provide good controller performance. Such a control strategy has however not yet been formulated and an alternative control strategy needs to be implemented to illustrate the state and parameter estimation.

PID control is the most commonly used methodology for controlling ROM ore milling circuits [17]. Variables are usually coupled in SISO loops and most of the time the derivative action is not implemented [1]. The decentralized PI controller used by [4] to control the ROM ore milling circuit is therefore used here. The manipulated variables are paired with the controlled variables as $CFF \rightarrow PSE$, $MFS \rightarrow LOAD$, and $SFW \rightarrow SLEV$. The feed of steel balls into the mill is kept constant at the nominal value and the feed of water into the mill is derived from MFS through a constant water-to-solids ratio for feed into the mill as discussed by [18].

6. Sensitivity analysis

The effect a parameter or a state has on the output is a good indication of how easily that parameter or state can be estimated from output data. The investigation of these effects is done through a sensitivity analysis [19].

The output sensitivity for the simultaneous estimation algorithm with respect to the augmented state vector (x^*) from Section 2.1 gives the output sensitivity with respect to the states and parameters and is given by

$$S_{y,s} = \frac{\partial y}{\partial x^*} \quad (38)$$

$$= \frac{\partial g}{\partial x} \frac{\partial x}{\partial x^*} + \frac{\partial g}{\partial \theta} \frac{\partial \theta}{\partial x^*} \quad (39)$$

$$= \frac{\partial g}{\partial x} \begin{bmatrix} \frac{\partial x}{\partial x} \\ \frac{\partial x}{\partial \theta} \end{bmatrix} + \frac{\partial g}{\partial \theta} \begin{bmatrix} \frac{\partial \theta}{\partial x} \\ \frac{\partial \theta}{\partial \theta} \end{bmatrix} \quad (40)$$

$$= \frac{\partial g}{\partial x} \begin{bmatrix} \text{I} \\ \frac{\partial x}{\partial \theta} \end{bmatrix} \quad (41)$$

$$= \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial x} \frac{\partial x}{\partial \theta} \end{bmatrix} \quad (42)$$

where we used the fact that in this case $\frac{\partial g}{\partial \theta} = 0$ to go from (40) to (41). This is because the parameters are not contained in the output equations, but only in the state transition equations. The first row of (42) applies to state estimation and the second row to parameter estimation. The partial derivatives $\frac{\partial g}{\partial x}$ can easily be determined from $y = g(x^*)$ but the derivatives $\frac{\partial x}{\partial \theta}$ are not directly calculable for the lack of a direct expression for the state values. The time derivative of the states are however known through (5) from which $\frac{\partial x}{\partial \theta}$ may be derived by writing

$$\frac{dx}{dt} = f(x, u), \quad (43)$$

then taking the partial derivative to θ on both sides to give

$$\frac{\partial}{\partial \theta} \left(\frac{dx}{dt} \right) = \frac{\partial f}{\partial \theta} + \frac{\partial f}{\partial x} \frac{\partial x}{\partial \theta} \quad (44)$$

and then changing the order of differentiation to give

$$\frac{d}{dt} \left(\frac{\partial x}{\partial \theta} \right) = \dot{S}_x = \frac{\partial f}{\partial \theta} + \frac{\partial f}{\partial x} S_x. \quad (45)$$

S_x is then found by integrating the function (45) from the initial time (t_o) to the final time (t_f).

For the dual estimation algorithm, a change in a parameter causes a change in the states and then the state transition equations are used to calculate the particle weights for parameter estimation (see (22)). This means that the sensitivity function to consider in this case is $\frac{\partial x}{\partial \theta} = S_x$ as defined previously. This change in the state will then cause the output to change, from which the particle weights are calculated for state estimation (see (20)). The sensitivity function describing this effect is $\frac{\partial y}{\partial x}$. The output sensitivity function for the dual estimation algorithm is therefore given by

$$S_{y,d} = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial x}{\partial \theta} \end{bmatrix}. \quad (46)$$

The only difference between the sensitivity functions $S_{y,s}$ and $S_{y,d}$ is therefore the extra $\frac{\partial g}{\partial x}$ term which is present in the second row of $S_{y,s}$ pertaining to parameter estimation. This is because in the simultaneous estimation algorithm a change in a parameter value must firstly cause a change in the state values before having an effect on the outputs, from which both the state and parameter values are then inferred. In the dual estimation scheme a change in a parameter value has a direct effect on the state transition equations from which the parameter values are inferred.

This implies that for systems where $\frac{\partial g}{\partial x}$ is ill-conditioned, deteriorated accuracy for parameter estimation would result using the simultaneous formulation.

In order to determine the conditioning of the matrix, the condition number is calculated. This is the ratio (C) of the largest to the smallest singular values of a matrix [20]. The condition number of a matrix gives an estimate of the worst-case loss of precision when solving a linear system with that matrix. The condition number for a $n \times n$ square matrix with elements randomly chosen from a normal distribution is of the order \sqrt{n} [21]. This result suggests that this problem would be more common in systems with high dimensionality.

7. Estimation results

In order to illustrate the accuracy of the simultaneous estimation as well as the dual estimation algorithms, a simulation run for the milling circuit is performed while kept in feedback control by a PI controller over a period of 20 hours. The value of ϕ_f is decreased by 20% at time 2 hours, the value of α_r is decreased by 20% at time 8 hours and the value of α_f is increased by 20% at time 14 hours. Time plots for the manipulated variables are shown in Fig. 7.

The nominal data generated by the simulation are processed by both estimation schemes to determine estimates of all five of the states and both the parameters ϕ_f and α_r . The particle filters used by the simultaneous as well as the dual estimation schemes are specified with 50 particles each. Even though the dual estimation scheme makes use of two particle filters, the sum of the dimensions equals the dimensionality of the simultaneous estimation scheme such that an equal number of function evaluations are required. The initial estimates of the states and parameters are randomly selected from a

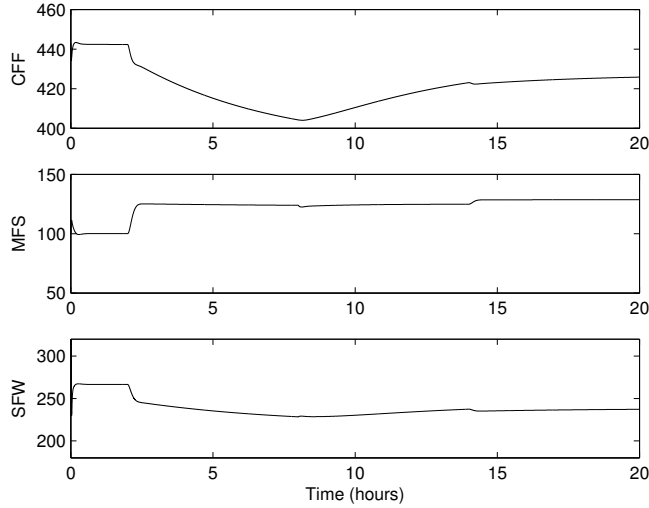


Figure 7: Manipulated variables

region (± 0.01) around the actual initial values in each case.

The state estimation results from the simultaneous estimation algorithm are shown in Fig. 8 and the parameter estimates are shown in Fig. 9.

The state estimation results from the dual estimation algorithm are shown in Fig. 10 and the parameter estimates are shown in Fig. 11.

The condition number of the matrix $\frac{dq}{dx}$ for the entire simulation run is shown in Fig. 12. Here it can be seen that the condition number of this matrix is very large for the entire simulation run, resulting in ill-conditioning of the system.

It is clear from the results that the dual state and parameter estimation scheme produces superior results to the simultaneous estimation scheme for this system. It is especially noticeable that when α_r changes value, the simultaneous estimation scheme cannot correctly attribute the amount of change in the output caused by a change in the parameter and the amount

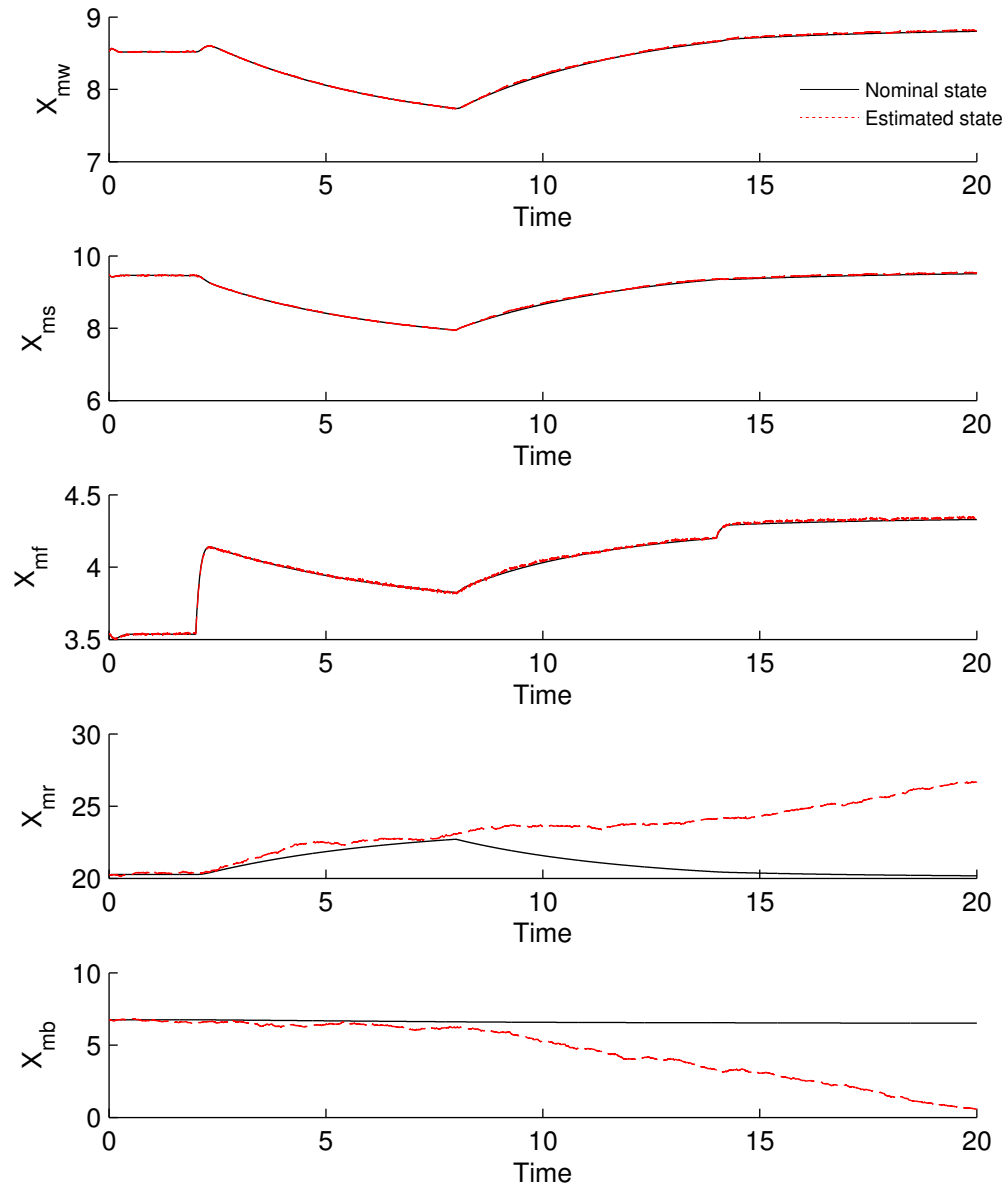


Figure 8: State estimates from simultaneous estimation algorithm

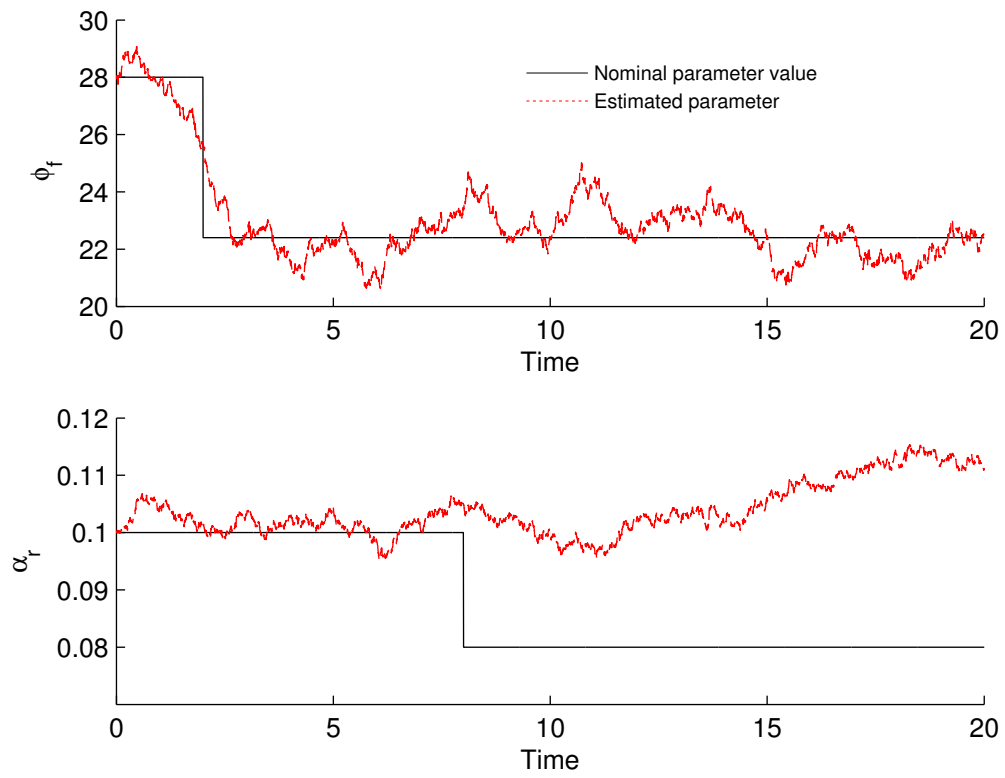


Figure 9: Parameter estimates from simultaneous estimation algorithm

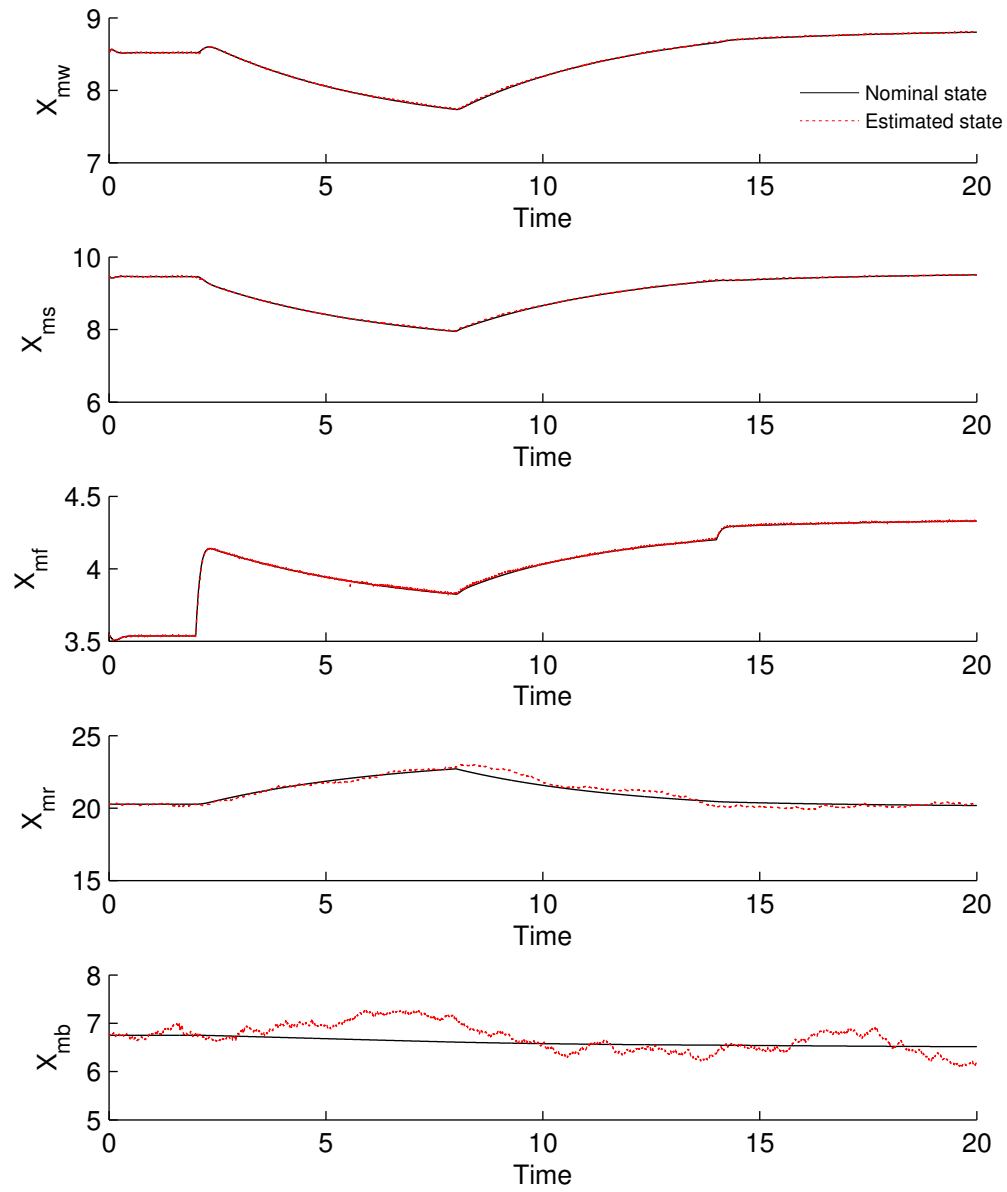


Figure 10: State estimates from dual estimation algorithm

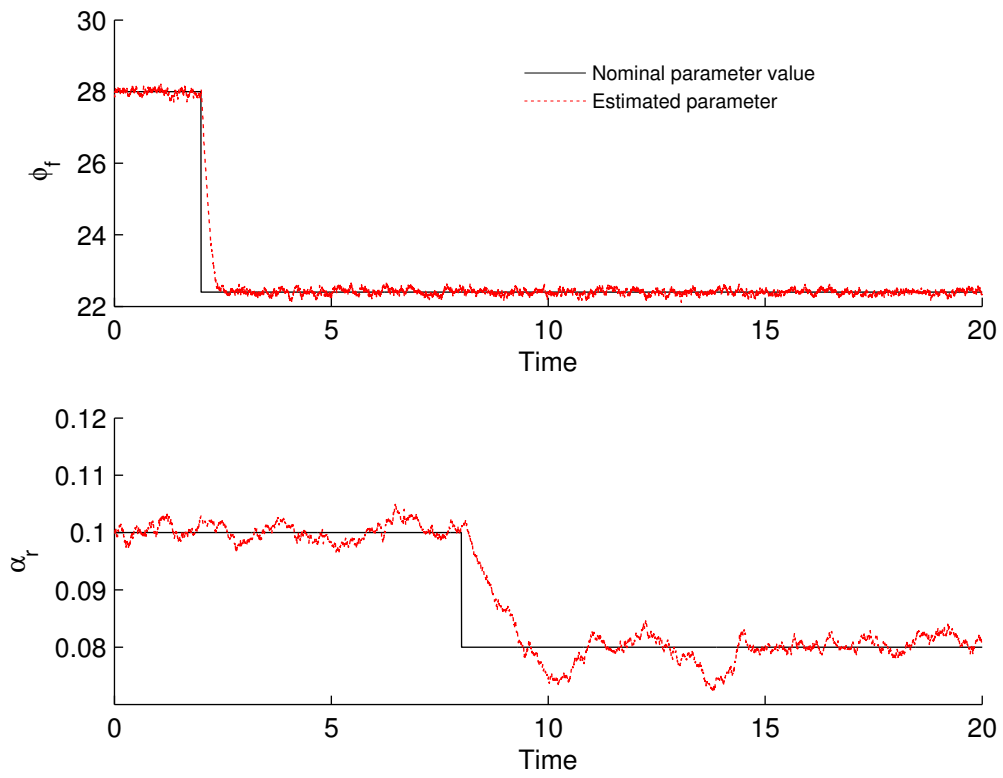


Figure 11: Parameter estimates from dual estimation algorithm

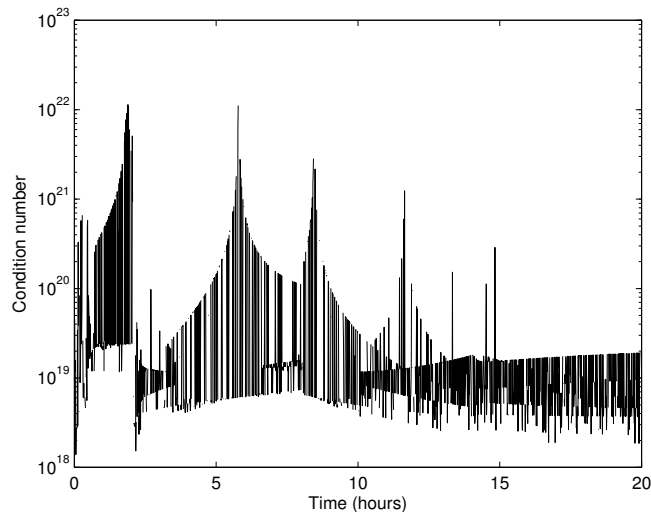


Figure 12: Condition number of $\frac{dg}{dx}$

of change caused by the subsequent change in the state value. For this reason the accuracy of state and parameter estimation deteriorates.

8. Conclusion

A change in a parameter value causes the state transition equation to change value. At this point the dual estimation algorithm will use the state transition equation information to infer the values of the parameters. The states will then change to cause a change in the outputs. At this point the dual estimation algorithm uses the outputs to determine the state values. It is only at this point where the simultaneous estimation algorithm infers the values of the states and parameters. The disregard of the information supplied by the state transition equations in the simultaneous estimation scheme may lead to deteriorated estimation accuracy depending on the conditioning of the matrix $\frac{\partial g}{\partial x}$. This result was obtained from the sensitivity analysis done

for both estimation schemes in Section 6.

This simulation study shows how the dual estimation scheme outperforms simultaneous estimation in such a situation.

Run-of-mine ore milling circuits are generally difficult to control, partly because of unavailable process variable measurements. This study shows that accurate state and parameter estimation can be done for a grinding mill with the application of dual particle filters. These estimates may then be used in an advanced process control scheme to improve control performance.

Acknowledgement

Laurentz Olivier would like to thank Prof. Biao Huang and his research group for hosting his visit to the University of Alberta at which time this work was completed. Yuri Shardt is acknowledged for his discussions pertaining to this work. The support of the University of Pretoria postgraduate study abroad bursary program is also greatly acknowledged.

References

- [1] D. Hodouin. Methods for automatic control, observation, and optimization in mineral processing plants. *J. Process Control*, 21, pp. 211 – 225, 2011.
- [2] I.K. Craig and I.M. MacLeod. Specification Framework for Robust Control of a Run-of-Mine ore Milling Circuit. *Control Eng. Practice*, 3, pp. 621 – 630, 1995.

- [3] X.S. Chen, J. Yang, S.H. Li and Q. Li. Disturbance observer based multivariable control of ball mill grinding circuits. *J. Process Control*, 19, pp. 1205 – 1213, 2009.
- [4] L.E. Olivier and I.K. Craig and Y.Q. Chen. Fractional order and BICO disturbance observers for a run-of-mine ore milling circuit. *J. Process Control*, 22, pp. 3 – 10, 2012.
- [5] A.S. Badwe, R.D. Gudi, R.S. Patwardhan, S.L. Shah and S.C. Patwardhan. Detection of Model-Plant Mismatch in MPC applications. *J. Process Control*, 19, pp. 1305 – 1313, 2009.
- [6] L.E. Olivier and Ian K. Craig. Parameter mismatch detection in a run-of-mine ore milling circuit under model predictive control. In: *Proc. 18th IFAC World Congress*, Milan, Italy, pp. 9929 – 9934, 2011.
- [7] S.J. Qin and T.A. Bagwell. A survey of industrial model predictive control technology. *Control Eng. Practice*, 11, pp. 733 – 764, 2003.
- [8] M.A. Henson. Nonlinear model predictive control: current status and future directions. *Computers and Chemical Engineering*, 23, pp. 187 – 202, 1998.
- [9] L.C. Coetzee, I.K. Craig and E.C. Kerrigan. Robust Nonlinear Model Predictive Control of a Run-of-Mine Ore milling Circuit. *IEEE Trans. Control Syst. Technol.*, 18, pp. 222 – 229, 2010.
- [10] B. Ristic, S. Arulampalam and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, Boston, 2004.

- [11] R. Van der Merwe. Sigma-point Kalman filters for probabilistic inference in dynamic state-space models. Ph.D. thesis, Oregon Health & Science University, 2004.
- [12] N. Bergman. Recursive Bayesian estimation: Navigation and tracking applications. Ph.D. thesis, Linkoping University, 1999.
- [13] M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing*, 50, pp. 174 – 188, 2002.
- [14] A. Doucet, S. Godsill and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10, pp. 197 – 208, 2000.
- [15] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5, pp. 97 – 112, 1996.
- [16] N.J. Gordon, D.J. Salmond and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F Radar and Signal processing*, 140, pp. 107 – 113, 1993.
- [17] D. Wei and I.K. Craig. Grinding mill circuits – A survey of control and economic concerns. *Int. J. Miner. Process.*, 90, pp. 56 – 66, 2009.
- [18] L.C. Coetzee. Robust nonlinear model predictive control of a closed run-of-mine ore milling circuit. Ph.D. thesis, University of Pretoria, 2009.

- [19] B.R. Jayasankar, A. Ben-Zvi and B. Huang. Identifiability and estimability study for a dynamic solid oxide fuel cell model. *Computers and Chemical Engineering*, 33, pp. 484 – 492, 2009.
- [20] B.N. Datta. *Numerical Linear Algebra and Applications*. Pacific Grove: Brooks/Cole, 1994.
- [21] A.M. Turing. Rounding-off errors in matrix processes. *Quart. J. Mech. Appl. Math.*, 1, pp. 287 – 308, 1948.