DYNAMIC OBSERVER MODEL BASED ON MODULAR PLANT FLOWSHEET

M. Mulholland[*], J-R. Vosloo[*], B.K. Loveday[*] and D.G. Hulbert[#]

[*] *School of Chemical Engineering, University of KwaZulu-Natal, Durban 4001, South Africa*
[#] *MINTEK, Randburg 2125, South Africa*

Abstract: The flow schemes of processing plants generally have well-defined nodes, such as reactors or separators, interconnected with pipes. Bearing in mind that the unit representing each node has a certain hold-up inventory, the dynamic description of the combined network can be expected to comprise differential and algebraic equations, the latter representing stream additions or characteristic relations within the units. In this paper, a method is demonstrated for a node-based description of networks of arbitrarily interconnected units, and the combined system is solved in real-time using a Kalman filter, allowing any combination of variables to be switched between "observed" and "unobserved" at any time, with appropriate reconciliation. The target application is the flotation network of a Platinum-concentrating plant. *Copyright 2005 IFAC*

Keywords: dynamic, observer, DAE, extended Kalman filter, reconciliation, nonlinear, mineral flotation

1. INTRODUCTION

The observer model was developed to gain insight into the phenomena occurring in mineral-processing circuits, which usually have too few measurements to illucidate conditions in the complex flow networks used. The model was tested using data from a South African platinum flotation plant, processing the UG2 chromite seam of the Bushveld Complex (Fig.1). This plant uses 10 flotation stages to concentrate platinum-containing copper/nickel sulphide minerals from about 0.2%$^{m}/_{m}$ to 18%$^{m}/_{m}$. The concentrations of chromite, gangue and water must also be tracked through this system.

As will be seen, a method has been developed which continuously estimates not only the system states, but associated variables such as flows and key parameters determining the operation. Amongst the latter are the first-order flotation rate constants for the main classes of mineral, namely platinum-containing sulphides, chromite and gangue, for each flotation stage. An extended Kalman filter is able to reconcile dynamically such measurements as are presented to it at any time, even allowing the measurement selection to vary in real time. The model has been integrated into the *PlantStar* web-based monitoring and control system developed by MINTEK in Randburg, South Africa.

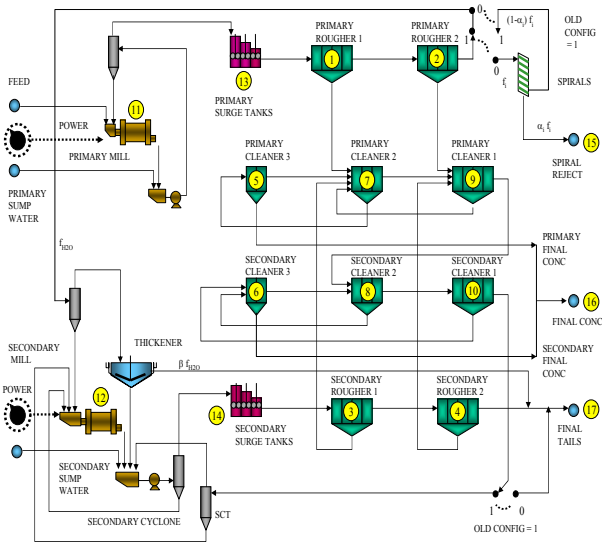

Fig. 1. Platinum flotation plant

The definition of variables, equations (differential and algebraic), processing units and interconnections defining the processing plant, is done in a modular fashion, allowing easy reconfiguration for other plants, or even other systems like multi-stage distillation.

Yang and Lee (1997) apply an extended Kalman filter to the identification of a distillation column. In order to reduce the dimensionality of the problem, the column was divided into four sections. Bagui *et al.* 2004) were able to use a Kalman filter to accurately estimate interior temperatures in a heat exchanger, by first invoking theoretical profiles based on a steady-sate solution.

Chiari *et al.* (1997) outline a typical refinery steady-state data reconciliation and optimisation problem. Before an economic optimiser can be applied, a consistent plant data set is required. McBrayer *et al.* (1998) describe the application of a nonlinear *dynamic* data reconciliation procedure, capable also of detecting gross error (bias).

Albuquerque and Biegler (1997) propose a method for resolution of the estimation problem for systems described by a *large* set of DAEs. In their method they discretise the DAE model with an implicit Runge-Kutta scheme, to avoid problems arising from stiffness, then use SQP to solve the resulting NLP.

Although the Kalman filter is used in the present study under the assumption that errors are randomly distributed with zero mean, a number of workers have recognised the importance of first detecting gross error (eg. offset). Thus Bagajewicz and Jiang (1998), and McBrayer *et al.* (1998), propose methods for identification of gross error (eg. offset) in dynamic data reconciliation for plants.

In the present paper, the extendend Kalman filter will be used to both reconcile measurements, *and* integrate the set of DAE. Thus it is interesting to note an important problem in DAE solutions – consistent initialisation. If the integration does not start at a feasible point, the likelihood of a useful result is much diminished. The search for a consistent initial condition for a set of DAE is an optimisation problem in which one attempts to get as close as possible to consistency (Biscaia and Vieira, 2000, and Vieira and Biscaia, 2001) Fortunately, the method of initialisation and solution in the present work adequately avoided initialisation problems.

Mjaavatten and Foss (1997) develop a *modular* dynamic flowsheeting technique for diagnosis and estimation of a fertilizer plant. Their approach is similar to that used in the present paper, except that they decomposed the *estimation* problem into the modules (ie. distinct plant units like reactors and strippers), or subsystems of modules. As will be seen later, the large dimensionality of the estimation problem, if this is not done, demands considerable computation. On the other hand, there is a problem of devising sensible decomposition rules, and dealing with interactions, in their method.

The simulation of processing plants has reached an advanced state with current commercial flowsheet modelling products such as those of *Aspen, Hysys,* and *SimSci.* One such package developed specifically for the minerals environment is *MODSIM.* The vendors usually also provide modelling formats that may be used for plant data reconciliation, eg. *SimSci*'s *DATACON*. The flowsheeting packages often include a dynamic simulation option, which allows the modeller to simulate some of the dynamic phenomena in an unsteady process. At this point there does not appear to be a commercial dynamic *data reconciliation* version available - that is the job of a Kalman filter!

A useful feature of the vendor-supplied packages is obviously that a process flowsheet can be built up by inter-linking standard modelling elements. A modular modelling format was likewise aimed for in the present work. The development of the extended Kalman filter (EKF) framework, the dynamic model representing Lonmin's Eastern Platinum B-Stream, and the implementation within MINTEK's *PlantStar* software, is more fully described in a thesis by Vosloo (2004). It is noteworthy that the present observer was developed as part of a larger on –line system of MINTEK which illucidates plant operation, including verbal messages. In the same organisation, Singh *et al.* (2003) provide the *FloatStar* algorithm for stabilisation of the level and flow oscillations which can occur in flotation networks, whilst Smith *et al.* (2004) provide the *MillStar* algorithm for stabilisation of the associated mill.

2. THEORY

2.1 Definition of modelling elements and circuit topology

The dynamics of the system arise from the points of hold-up - ie. the volumes of pulp in the interconnected vessels. The most complicated such vessel is the flotation cell shown in Fig. 2, which has one feed stream and two product streams. There are C such flotation cells in the circuit. We consider M different species moving through the system, and we always take the Mth component as the water which carries the solids. Then $M-1$ mineral components are allowed for (eg. 1:Platinum-containing sulphides, 2:Chromite, 3:Gangue, 4:water). In Fig.2, 'j' refers to any one of these species. Lower-case characters refer to the froth phase leaving the top of the vessel, and capitals refer to the pulp phase. The double character names SS and ss refer to inventories, whilst the single characters S^*, S, s and s_p refer to stream flow rates. A is the air flow rate to the cell.

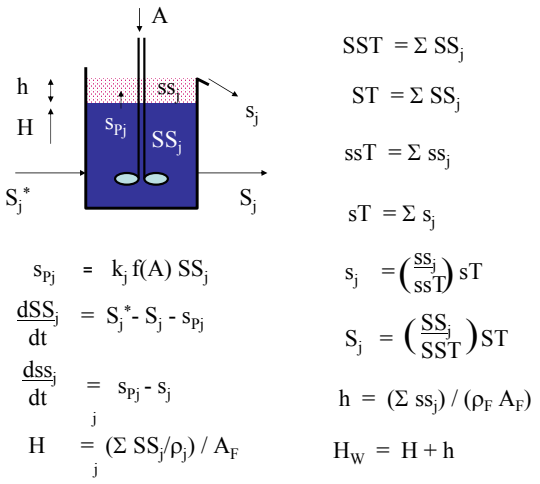

Fig. 2. Flotation cell representation

It is already apparent that the definition of this flotation cell requires 2*M* differential equations, and quite a number of algebraic equations. The network topology, that is the interconnection of the vessels, will be described by writing the feed streams S^* as the sum of streams *S* and *s* from other vessels, so this will of course add even more algebraic equations. These equations can possibly be combined to produce a set of differential equations only, but in the process a lot of algebra will be required, and the set of equations will lose readability. So the trend nowadays is to solve the system as it stands - a set of Differential and Algebraic Equations (DAE). In the next section a method will be introduced to deal with this in the context of the Kalman filter, which deals only with a set of differential equations.

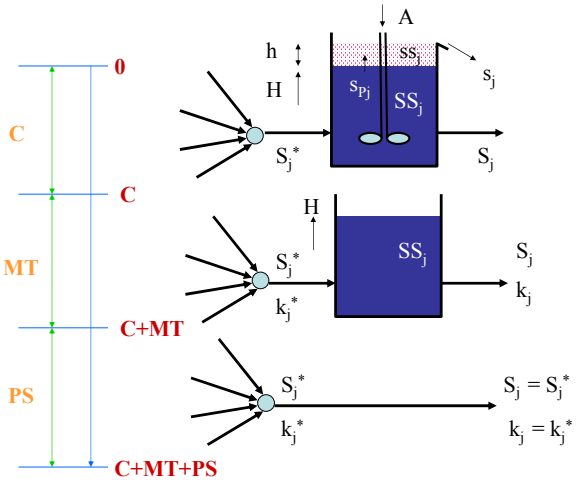

Fig. 3. The three different types of elements

The other vessels which introduce dynamics in our mineral-processing circuits are surge tanks and mills, on account of their hold-up inventories. From a dynamic viewpoint, both of these may be represented as a stirred vessel with one feed and one product stream. The equations are just a sub-set of those for the flotation cell, so the same "module" may be used. There are *MT* such elements in our circuit.

So far, our only means of combining streams is by our definition of the "collection node" S^* which feeds to a flotation cell, mill or tank. To provide this service without, eg., having to use a zero-volume tank, a third conceptual member of this family is defined as a 'Product Stream'. There are *PS* such elements in our circuit. Again, this is a subset of the same module above.

Figure 3 illustrates how the complete plant can be constructed from *C+MT+PS* nodal elements, each of which can sum as many streams as desired on its input.

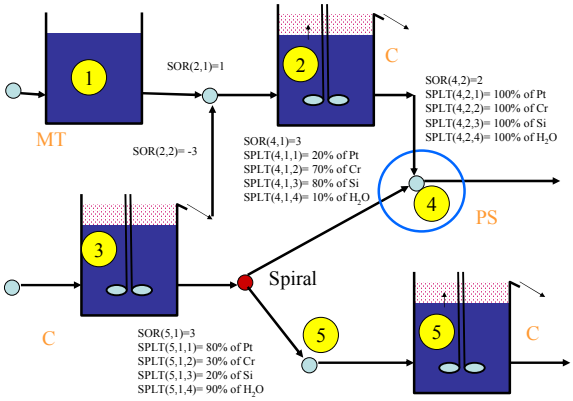

Fig. 4. Use of receiving node selection matrix to represent stream splitters

So far we have not considered how to deal with a stream split, either a total division, or selectively by component as in a thickener or spiral separator. This is quite simply dealt with by means of a selection matrix operating at the summation node of each element. Any number of downstream elements can reference fractions of the components of a single upstream exit flow. It turns out, in the circuits considered, a maximum of 4 streams arrive at any one receiving node. So our selection matrix needs 4 columns, and *C* rows, each with a fraction to be selected from that component flow in the relevant stream. Another vector keeps the name (number) of the four possible elements of which the exit stream is being selected. If this number is shown as negative, it is referring to the froth exit stream instead of the tailings exit stream (Fig. 4). Thus the final set of algebraic equations required to define the system topology has the form

$$S^*_{i,j} = \sum_{k=1}^{4} SPLT(i,k,j) \bullet S_{SOR(i,k)|_{IF\ POSITIVE},j} + \sum_{k=1}^{4} SPLT(i,k,j) \bullet S_{-SOR(i,k)|_{IF\ NEGATIVE},j} \quad (1)$$

It has been mentioned that the entire definition of the circuit is node-based. The node indicates which of

the three *types* of processing units it is, with the associated *equipment parameters*, and it gives its receiving node *selection matrix*. However, for a "cold" or "raw" start of the model, it is necessary to give further information (initial values of the states) to provide a realistic starting-point for the solution. In Fig. 5, an extract from the original initialisation program shows the definition of two elements. Initial values for the compositions and flows of the exit streams of the element are based on plant assays where available, or are otherwise roughly estimated.

```
%-------------------------------------------------------------------------------
% (13) Primary Surge Tanks      % (1)Primary Rougher No.1      %
%                               %   4xWemco 144D              %
; j=13                         ;  q=1                        ;
; Sorc(j,:) =[ 11  0  0  0   ] ;  Sorc(q,:) =[ 13  0  0  0   ] ;
; Splt(j,:,1)=[  1  1  1  1   ] ;  Splt(q,:,1)=[  1  1  1  1   ] ;
; Splt(j,:,2)=[  1  1  1  1   ] ;  Splt(q,:,2)=[  1  1  1  1   ] ;
; Splt(j,:,3)=[  1  1  1  1   ] ;  Splt(q,:,3)=[  1  1  1  1   ] ;
; Splt(j,:,4)=[  1  1  1  1   ] ;  Splt(q,:,4)=[  1  1  1  1   ] ;
; WtSiBal(j) =1                 ;  WtSiBal(q) =1              ;
; FMio (j) =60                  ;  FMio (q) =140              ;
; SMFio(j) =0.4                 ;  SMFio(q) =.4               ;
; Xio(j,:) =[4.6e-6 .243 .757]  ;  Xio(q,:) =[4.6e-6 .243 .757] ;
; AF(j)   =4                    ;  AF(q)   = 4*2.74*3.66      ;
; Hw(j)   =4                    ;  Hw(q)   = 1.41             ;
; Ho(j)   =0.8*Hw(j)            ;  Ho(q)   = 0.92*Hw(q)       ;
; FMo (j) =60                   ;  FMo (q) =138               ;
; SMFo(j) =0.4                  ;  SMFo(q) =.441              ;
; XoXo(j,:) =[4.6e-6 .243 .757] ;  Xo(q,:) =[1.8e-6 .25 .75   ] ;
;                               ;                             ;
;                               ;  Ao(q)   =4*9.7             ;
;                               ;  fmo(q)  =5                 ;
;                               ;  smfo(q) =.182              ;
;                               ;  xo(q,:) =[82.5e-6 .044 .9559 ] ;
;                               ;                             ;
%-------------------------------------------------------------------------------
```

Fig. 5. Node-based definition of topology, parameters & initial state

Reviewing the above, we note that there are two types of variables: Those that appear in the derivative terms (states) and those that do not, but are otherwise necessary in the algebraic equations which interconnect the system ("incidental"). Actually, there is a third class of variable - actual physical parameters describing the equipment. It will be seen that it is going to suit us to allow some of these (eg. flotation rate constants) to apparently vary in time, like a flow, level or composition - we can lose this distinction within the EKF. So we have the two broad types of time-varying parameters, and two types of equation: differential and algebraic.

2.2 Solution of the system of Differential and Algebraic Equations (DAE)

Consider the system of first order differential and algebraic equations

$$\frac{d\,\boldsymbol{y}}{d\,t} = \boldsymbol{f}(\boldsymbol{y},\boldsymbol{z}) \tag{2}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{y},\boldsymbol{z}) \tag{3}$$

Here the values in the $\boldsymbol{y}$ vector are clearly the states, and those in the $\boldsymbol{z}$ vector are the "incidental variables" such as flows. The set of algebraic equations has been rearranged to give zero on the left. Defining the Jacobians

$$\boldsymbol{A} = \boldsymbol{J}_{fy} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \cdots & \frac{\partial f_1}{\partial y_N} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \cdots & \frac{\partial f_2}{\partial y_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial y_1} & \frac{\partial f_N}{\partial y_2} & \cdots & \frac{\partial f_N}{\partial y_N} \end{bmatrix} \tag{4}$$

$$\boldsymbol{B} = \boldsymbol{J}_{fz} = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \cdots & \frac{\partial f_1}{\partial z_M} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \cdots & \frac{\partial f_2}{\partial z_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial z_1} & \frac{\partial f_N}{\partial z_2} & \cdots & \frac{\partial f_N}{\partial z_M} \end{bmatrix} \tag{5}$$

$$\boldsymbol{C} = \boldsymbol{J}_{gy} = \begin{bmatrix} \frac{\partial g_1}{\partial y_1} & \frac{\partial g_1}{\partial y_2} & \cdots & \frac{\partial g_1}{\partial y_N} \\ \frac{\partial g_2}{\partial y_1} & \frac{\partial g_2}{\partial y_2} & \cdots & \frac{\partial g_2}{\partial y_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_N}{\partial y_1} & \frac{\partial g_N}{\partial y_2} & \cdots & \frac{\partial g_N}{\partial y_N} \end{bmatrix} \tag{6}$$

$$\boldsymbol{D} = \boldsymbol{J}_{gz} = \begin{bmatrix} \frac{\partial g_1}{\partial z_1} & \frac{\partial g_1}{\partial z_2} & \cdots & \frac{\partial g_1}{\partial z_M} \\ \frac{\partial g_2}{\partial z_1} & \frac{\partial g_2}{\partial z_2} & \cdots & \frac{\partial g_2}{\partial z_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_N}{\partial z_1} & \frac{\partial g_N}{\partial z_2} & \cdots & \frac{\partial g_N}{\partial z_M} \end{bmatrix} \tag{7}$$

linearise the right hand sides of (2) and (3) about ($\boldsymbol{y}_t$,$\boldsymbol{z}_t$) to obtain the augmented system

$$\begin{pmatrix} \dot{\boldsymbol{y}} \\ \hline \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}(\boldsymbol{y}_t,\boldsymbol{z}_t) \\ \hline \boldsymbol{g}(\boldsymbol{y}_t,\boldsymbol{z}_t) \end{pmatrix} + \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right] \begin{pmatrix} (\boldsymbol{y}-\boldsymbol{y}_t) \\ \hline (\boldsymbol{z}-\boldsymbol{z}_t) \end{pmatrix}$$

$$= \begin{pmatrix} \boldsymbol{F}_t \\ \hline \boldsymbol{G}_t \end{pmatrix} + \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right] \begin{pmatrix} \boldsymbol{y} \\ \hline \boldsymbol{z} \end{pmatrix} \tag{8}$$

where $\begin{pmatrix} \boldsymbol{F}_t \\ \hline \boldsymbol{G}_t \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}(\boldsymbol{y}_t,\boldsymbol{z}_t) - \boldsymbol{A}\boldsymbol{y}_t - \boldsymbol{B}\boldsymbol{z}_t \\ \hline \boldsymbol{g}(\boldsymbol{y}_t,\boldsymbol{z}_t) - \boldsymbol{C}\boldsymbol{y}_t - \boldsymbol{D}\boldsymbol{z}_t \end{pmatrix}$

Some of the elements in the $\boldsymbol{z}$ vector may be independent (eg. plant feed rate), whilst others may be determined by the solution of (8). In either case, we gain access to them by conferring additional dynamic behaviour in which $\boldsymbol{z}$ will move towards some given value $\boldsymbol{z}_t$.

$$\dot{\boldsymbol{z}} = \tfrac{1}{\tau}(\boldsymbol{z}_t - \boldsymbol{z})$$

Here we have chosen a single time-constant τ that applies to all such variables. It is chosen to be short – of similar size to the integration interval used in the solution, implying that the incidental variables will steadily track the given values. Then

$$\begin{pmatrix} \dot{\boldsymbol{y}} \\ \hline \dot{\boldsymbol{z}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_t \\ \hline \boldsymbol{H}_t \end{pmatrix} + \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{0} & \boldsymbol{E} \end{array}\right] \begin{pmatrix} \boldsymbol{y} \\ \hline \boldsymbol{z} \end{pmatrix} \tag{9}$$

with

$\boldsymbol{H}_t = \frac{1}{\tau}\boldsymbol{z}_t$

$\boldsymbol{E} = -\frac{1}{\tau}\boldsymbol{I}$

and we have the additional requirement that

$$\left[\boldsymbol{C} \mid \boldsymbol{D}\right]\left(\frac{\boldsymbol{y}}{\boldsymbol{z}}\right) = -\boldsymbol{G}_t \qquad (10)$$

Letting $\boldsymbol{x} = \left(\frac{\boldsymbol{y}}{\boldsymbol{z}}\right)$, $\boldsymbol{\Phi} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{0} & \boldsymbol{E} \end{array}\right]$ and $\boldsymbol{u}_t = \left(\frac{\boldsymbol{F}_t}{\boldsymbol{H}_t}\right)$,

integrate (9) from t to $t+\Delta t$ keeping $\boldsymbol{u}_t$ fixed:

$$\boldsymbol{x}_{t+\Delta t} = e^{\boldsymbol{\Phi}\Delta t}\boldsymbol{x}_t + \left[e^{\boldsymbol{\Phi}\Delta t} - \boldsymbol{I}\right]\boldsymbol{\Phi}^{-1}\boldsymbol{u}_t \qquad (11)$$

ie.

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{A}_t\boldsymbol{x}_t + \boldsymbol{B}_t\boldsymbol{u}_t \qquad (12)$$

with $\boldsymbol{A}_t = e^{\boldsymbol{\Phi}\Delta t}$ (13)

$$\boldsymbol{B}_t = \left[e^{\boldsymbol{\Phi}\Delta t} - \boldsymbol{I}\right]\boldsymbol{\Phi}^{-1} \qquad (14)$$

Note that the possible singularity of $\boldsymbol{\Phi}$ is circumvented by evaluating (14) as

$$\left[e^{\boldsymbol{\Phi}\Delta t} - \boldsymbol{I}\right]\boldsymbol{\Phi}^{-1} = \Delta t\left\{\frac{\boldsymbol{I}}{1!} + \frac{(\boldsymbol{\Phi}\Delta t)}{2!} + \frac{(\boldsymbol{\Phi}\Delta t)^2}{3!} + \frac{(\boldsymbol{\Phi}\Delta t)^3}{4!} + \ldots\ldots\ldots\right\} \qquad (15)$$

Now augment equation (10) as

$\boldsymbol{C}_t = \left[\begin{array}{c|c} \boldsymbol{L} & \boldsymbol{0} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right]$, $\hat{\boldsymbol{w}}_t = \left(\frac{\hat{\boldsymbol{y}}_t}{-\boldsymbol{G}_t}\right)$, with $\boldsymbol{L}$ selecting (from $\boldsymbol{y}$) values corresponding to an observation subset $\hat{\boldsymbol{y}}_t$, ie. we seek to satisfy as closely as possible

$$\boldsymbol{C}_t\boldsymbol{x}_t = \hat{\boldsymbol{w}}_t \qquad (16)$$

where $\hat{\boldsymbol{w}}_t$ is clearly a current set of measurements.

Thus configure a Kalman filter as follows:

$$\boldsymbol{K}_t = \boldsymbol{M}_t\,\boldsymbol{C}_t^T\left[\boldsymbol{C}_t\,\boldsymbol{M}_t\,\boldsymbol{C}_t^T + \boldsymbol{R}\right]^{-1} \qquad (17)$$

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{A}_t\,\boldsymbol{x}_t + \boldsymbol{B}_t\,\boldsymbol{u}_t + \boldsymbol{K}_t\left[\hat{\boldsymbol{w}}_t - \boldsymbol{C}_t\,\boldsymbol{x}_t\right] \qquad (18)$$

$$\boldsymbol{M}_{t+\Delta t} = \boldsymbol{A}_t\left[\boldsymbol{I} - \boldsymbol{K}_t\,\boldsymbol{C}_t\right]\boldsymbol{M}_t\,\boldsymbol{A}_t^T + \boldsymbol{Q} \qquad (19)$$

where the covariance matrix is initialised with $\boldsymbol{M}_0$ small, diagonal, and $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the expected error covariance matrices for the model and the measurements respectively.

What has not been mentioned so far is the meaning and use of the 'given' values $\boldsymbol{z}_t$ of the incidental variables. Only a selection of these will be available as observations. For that selection, the tuning of the Kalman filter will be used to determine how closely the model follows the first-order approach to the given values. For the remaining 'free' incidental variables, they have to find values which will allow the best overall fit of model and observations. The mechanism used to achieve this is simply to reset the 'observed value' z_t for the next time-step to the final value of z at the end of the preceding time-step. Thus the tuning of the filter can be used to determine how *fast* these variables move, but there is no absolute resistance to their overall range of movement.

Using as the initial condition a measured data set of the plant (assays, and some stream flow rates), no problem was encountered due to inconsistency, a common problem noted by others in section 1. The equations above show that the covariance matrix $\boldsymbol{M}_t$ is the only entity carrying the filter information from time-step to time-step. It is square with a side length equal to the total number of variables ($\boldsymbol{y}$ & $\boldsymbol{z}$) in the system. It does not change size as variables enter or leave the measurement set, though there is some impact on it as the observation selection matrix $\boldsymbol{C}_t$ varies. The effect of a complete switch from normal 'forward modelling' (feed → products), to inverse modelling where flotation rate constants are being determined from given internal stream properties, is only a little disruptive in the evolution of $\boldsymbol{M}_t$, and quite tolerable, though the coding does provide for the automatic swopping of $\boldsymbol{M}_t$ with previously-converged copies appropriate for different observation set configurations.

An automatic technique sets the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices, based on a nominal range for each variable. The user specifies only a percentage error of the range to be expected in the tracking of each variable. The nominal ranges are generally updated (as twice the present value) on each time-step using a slow filter.

3. CASE STUDY RESULTS

Although the model has been installed on the plant under MINTEK's *PlantStar* open SCADA system, and undergone preliminary testing, with acceptable results, the identification study reported is based on an off-line simulation. It is a typical identification scenario, where an observation of the total mass-flow of water and mineral in the froth of selected cells is available. From this information, the "Cell Factor", which multiplies all individual flotation kinetic constants for that cell, can be inferred. In the test sequence depicted in figures 5 and 6, the froth mass flow rates of cells 1,2 and 4 were set to "observed", and the corresponding cell factors were freed as "unobserved" (The cell numbering is according to Fig. 1). Firstly the froth flow for cell 1 was "observed" to drop to 50%, then that for cell 2 was observed to drop to 50%, then rise to 150%. Finally, that for cell 4 dropped to 50%. Actually, by the time cell 4 is dropped, the impact of changes upstream has asserted on the secondary roughers, and the froth flow on cell 4 cannot be held up to 50% of its original. Indeed, with the simultaneous changes in cell inventories that result, there is no guarantee that a froth flow change will proportionally change the cell factor, though this is indeed seen to be the case in cells 1 and 2 in Fig. 6.

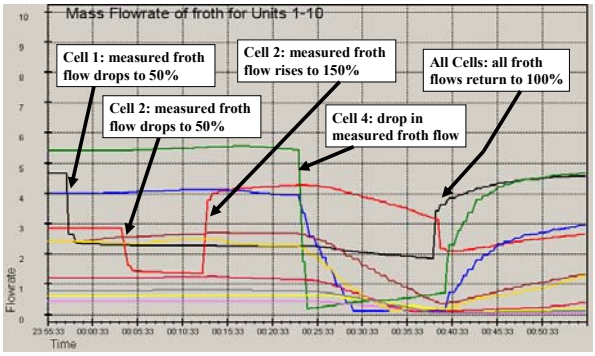

Fig. 5. Manually set changes in measured froth flows on cells 1,2 and 4 with resultant secondary changes in other froth flows downstream

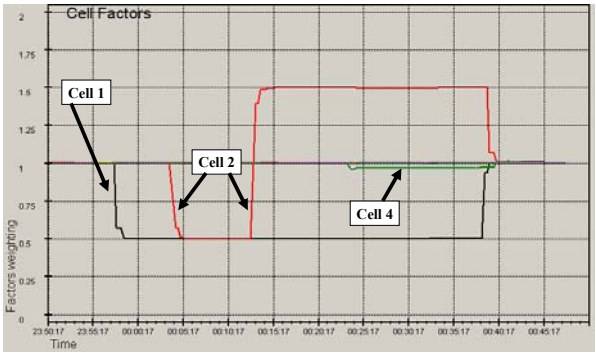

Fig. 6. Cell-factor identification following manually set changes in froth flows for cells 1, 2 and 4 in Fig. 5.

4. CONCLUSION

Problems arising from inconsistent initialisation of the system of DAEs appeared insignificant – possibly because a balanced initial condition was set using a measured plant assay and a number of measured internal flow-rates. No attempt was made to decompose the system, so the most problematic and time-consuming aspect of the solution was equation (17), which was rearranged as the linear system

$$\boldsymbol{x} = \boldsymbol{A} \backslash \boldsymbol{y}$$

Some problems of accuracy were experienced using a sparse matrix (conjugate gradient) solution. A faster, more accurate solution was obtained using the "MATLAB Engine". In the case of the Lonmin B Stream model, the side of $\boldsymbol{A}$ varies from about 70 up to 150 in full identification mode (full assay available). The other dimension of $\boldsymbol{x}$ and $\boldsymbol{y}$ corresponds to the 300 or so fitted variables.

Variable values produced by the Kalman filter are clipped back to their constraints on each time-step, Lower constraints are usually zero, but upper constraints (also used for normalisation) slowly expand to eventually free high values again. The clipping can be a source of instability until variables are again freed.

This work has demonstrated that a fairly straightforward solution of a large set of DAEs using a Kalman filter is feasible, with the advantage that observations may be arbitrarily set. Though the computation proved somewhat faster than real-time, systems larger than that considered will need dedicated computers using present-day technology.

REFERENCES

Albuquerque, J.S. and L.T. Biegler (1997) Decomposition algorithms for on-line estimation with nonlinear DAE models. *Computers chem. Engng,* **21**, 283-299.

Bagajewicz, M.J. and Q. Jiang (1998). Gross error modeling and detection in plant linear dynamic reconciliation. *Computers chem. Engng,* **22**, 1789-1809.

Bagui, F., M.A. Abdelghani-Idrissi and H. Chafouk (2004). Heat exchanger Kalman filtering with process dynamic acknowledgement. *Computers chem. Engng,* **28**, 1465-1473.

Biscaia Jr., E.C. and R.C. Vieira (2000). Heuristic optimization for consistent initialization of DAEs. *Computers chem. Engng,* **24**, 2183-2191.

Chiari, M., G. Bussani, M.G. Grottoli and S. Pierucci (1997). On-line data reconciliation and optimisation: refinery applications. *Computers chem. Engng,* **21**(Suppl.), S1185-S1190.

McBrayer, K.F., T.A. Soderstrom, T.F. Edgar and R.E. Young (1998). The application of nonlinear dynamic data reconciliation to plant data. *Computers chem. Engng,* **22**, 1907-1911.

Mjaavatten, A. and B.A. Foss (1997). A modular system for estimation and diagnosis. *Computers chem. Engng,* **21**, 1203-1218.

Singh, A., J.J. Louw and D.G. Hulbert (2003). Flotation stabilization and optimization. *J. S.A.Inst. Mining & Metallurgy,* November 2003, 581-588.

Smith, G.C., L. Jordaan, A. Singh, V. Vandayar, V.C. Smith, B. Muller and D.G. Hulbert (2004). Innovative process control technology for milling and flotation circuit operations. *J. S.A.Inst. Mining & Metallurgy,* May 2004, 353-365.

Vieira, R.C. and E.C. Biscaia Jr. (2001). Direct methods for consistent initialization of DAE systems. *Computers chem. Engng,* **25**, 1299-1311.

Vosloo, J-R. (2004). Development and implementation of a real-time observer model for mineral-processing circuits. *MScEng Thesis, University of KwaZulu-Natal,* January, 2004.

Yang, D.R. and K.S. Lee (1997) Monitoring of a distillation column using modified extended Kalman filter and a reduced order model. *Computers chem. Engng,* **21**(Suppl.), S565-S570.